

# **NDS-DLL6**

Dynamic Linking Library  
for NuDAM 6000 Modules

## **User Guide**



@Copyright 1997~1999 ADLink Technology Inc.  
All Rights Reserved.

Manual first edition: October 15, 1997  
Manual Rev. 3.30: April 1, 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.

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**

NuDAM is a registered trademark of ADLink Technology Inc., IBM PC is a registered trademark of International Business Machines Corporation. Intel is a registered trademark of Intel Corporation. Other product names mentioned herein are used for identification purposes only and may be trademarks and/or registered trademarks of their respective companies.



# CONTENTS

What is NDS-DLL6? .....	1
Getting Start .....	3
<b>2.1 Installing NDS-DLL6</b> .....	<b>3</b>
2.1.1 System Requirements .....	3
2.1.2 Backup Your Disk .....	4
2.1.3 Installation .....	4
<b>2.2 System Architecture</b> .....	<b>5</b>
<b>2.3 Using NDS-DLL6</b> .....	<b>6</b>
2.3.1 Creating An Application Using Visual Basic and NDS-DLL6 .....	6
2.3.2 Creating An Application Using Visual C/C++ and NDS-DLL6 .....	9
Sample Programs .....	11
<b>3.1 Sample Programs Included</b> .....	<b>11</b>
<b>3.2 Sample Programs Development Environment</b> .....	<b>12</b>
3.2.1 Visual Basic Sample Programs .....	12
3.2.2 Visual C/C++ Sample Programs .....	12
<b>3.3 Execute Sample Programs</b> .....	<b>13</b>
<b>3.4 The Detail Descriptions of these Sample Programs</b> .....	<b>14</b>
3.4.1 VB\COMM .....	14

3.4.2	VB\6011	.....15
3.4.3	VB\6050	.....15
3.4.4	SDK\6060	.....16
<b>Appendix</b>	<b>Function Support</b>	<b>..... 17</b>

# How to Use This Guide

This manual is to help you install the NDS-DLL6 software driver for ADlink's NuDAM 6000 series data acquisition modules. The manual describes how to install and use the library to meet your requirements and help you program your own software applications. For the detailed description of each function, please see the on-line help or the PDF file of function reference manual provided by NDS-DLL6. This manual is divided into four chapters:

- Chapter 1, "What is NDS-DLL6" gives an overview of the contents included in the software package.
- Chapter 2, "Getting Start" describes how to install the software and how to program your applications by using NDS-DLL6.
- Chapter 3, "Sample Programs" describes some sample programs in the software.
- Appendix, "Function Support" shows which modules each NDS-DLL6 function supports.





## What is NDS-DLL6?

NDS-DLL6 includes Microsoft Windows 3.1, Win-95, Win-98, and Win-NT4.0 drivers for ADLink's NuDAM 6000 series data acquisition modules. It is a high performance data acquisition driver for developing custom applications under Windows 3.1, 95, 98, or NT. This driver is DLL (Dynamic Linking Library) version for use under Windows environment. They can work with any Windows programming language that allows calls to a DLL, such as Microsoft Visual C/C++, Microsoft Visual Basic, Borland Delphi, etc.

Using this DLL 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 NDS-DLL6's DLL make it easy to create custom user interfaces and graphics.

In addition to the DLL drivers, several sample programs are provided. You can refer to them and save a lot of programming time and get some other benefits as well.



# 2

## Getting Start

---

### 2.1 Installing NDS-DLL6

#### 2.1.1 System Requirements

NDS-DLL6 requires the following minimum configuration:

- An IBM PC/AT or 100% compatibles with Windows 3.1, 95, 98, or NT 4.0 as its operating system
- Eight megabytes of available memory
- A hard disk with enough disk space to install NDS-DLL6
- A 1.44-MB 3.5-inch disk drive or CD-ROM drive depending on the media stored NDS-DLL6 software
- Application development system: Microsoft Visual C/C++, Microsoft Visual Basic, or any Windows programming language that allows calls to a DLL.
- ADLink's NuDAM 6000 series data acquisition modules

## 2.1.2 Backup Your Disk

The diskette of NDS-DLL6 is in DOS format which is compatible with DOS 3.0 or higher versions. It is advisable to make a back up copy before using the software.

## 2.1.3 Installation

### Windows 3.1

**step 1.** Place the diskette or CD-ROM in the appropriate drive.

**step 2. With NDS-DLL6 diskette**

Execute **x:\setup.exe**. (**x** identifies the floppy drive that contains the NDS-DLL6 diskette)

**With ADLink's "Manual & Software Utility" CD-ROM**

Execute **x:\Software\NDS-DLL6\W31\setup.exe**. (**x** identifies the CD-ROM drive that contains the ADLink's "Manual & Software Utility" CD-ROM)

### Windows 95/98/NT

**step 1.** Place the diskette or CD-ROM in the appropriate drive.

**step 2. With NDS-DLL6 diskette**

Execute **x:\setup.exe**. (**x** identifies the floppy drive that contains the NDS-DLL6 diskette)

**With ADLink's "Manual & Software Utility" CD-ROM**

If the autorun setup program is not invoked, please execute **x:\setup.exe**. (**x** identifies the CD-ROM drive that contains the ADLink's "Manual & Software Utility" CD-ROM).

Select Software Package → NDS-DLL6 → Win95/98/NT to setup NDS-DLL6 software.

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

<b>File/Subdirectory</b>	<b>Description</b>
MANUAL <DIR>	The PDF format manual file, including User Guide and Function Reference.
HELP <DIR> (95/NT Version)	On-line help file
LIB <DIR>	NDS-DLL6 import library and DLL. Dll6.lib is the import library for Visual C/C++. Dll6_bc.lib is the import library for Borland C++.
INCLUDE <DIR>	Include files for application programming. DLL6.BAS for Visual Basic Programming. DLL6.H for Visual C/C++ and Borland C++ programming. DLL6.PAS for Delphi programming.
SAMPLES\SDK\6060 <DIR>	ND-6060 Visual C/C++ sample program
SAMPLES\VB\COMM <DIR>	COM port communication Visual Basic sample program
SAMPLES\VB\SDK6011 <DIR>	ND-6011 Visual Basic sample program
SAMPLES\VB\SDK6050 <DIR>	ND-6050 Visual Basic sample program

---

## 2.2 System Architecture

The following diagram shows the interface between your applications and NDS-DLL6 drivers:

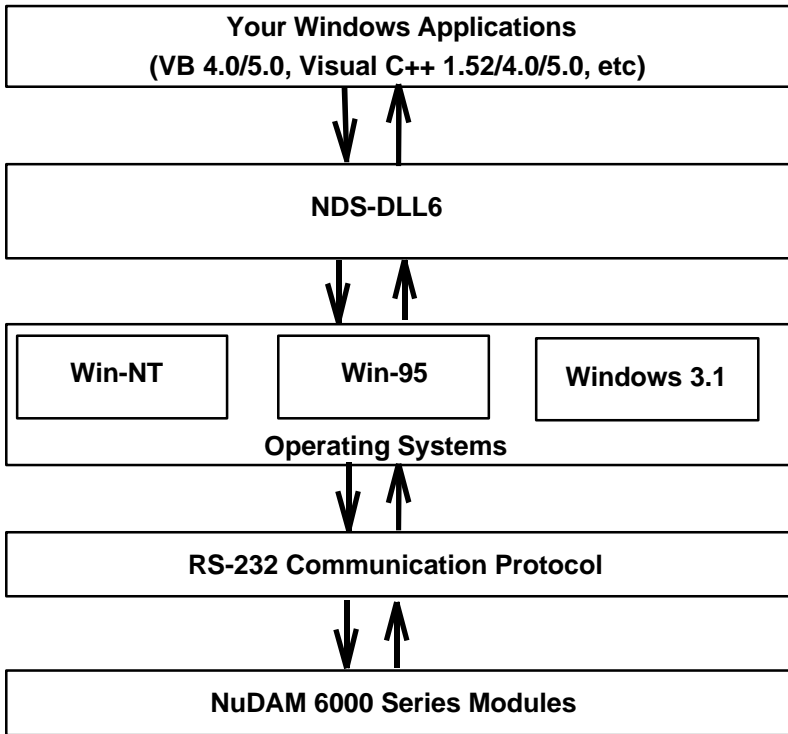


Figure 2.1

---

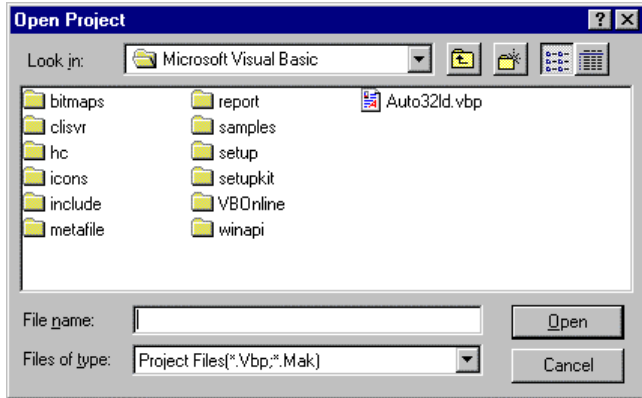
## 2.3 Using NDS-DLL6

### 2.3.1 Creating An Application Using Visual Basic and NDS-DLL6

To create a data acquisition application using NDS-DLL6 and Visual Basic, follow these steps after entering Visual Basic:

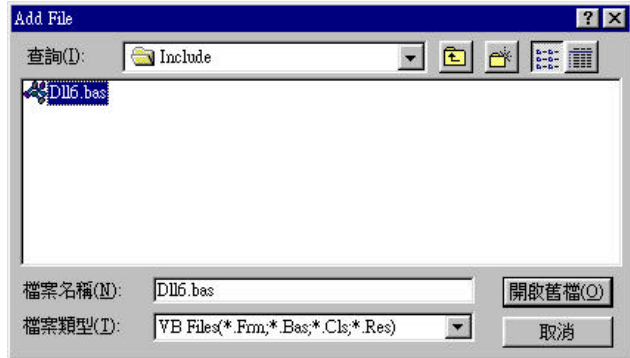
- step 1.** Open the project in which you want to use NDS-DLL6. This can be a new or existing project.

Open a C 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 DLL6.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.




Select DLL6.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. DLL6.bas is located in <INSTALLDIR>\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 DLL6.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 NDS-DLL6 software driver DLL6.DLL. This DLL file should be copied to the Windows System directory (by default, C:\WINDOWS\SYSTEM for Windows 3.1, 95, or 98, C:\WINNT\SYSTEM32 for Win-NT).

### 2.3.2 Creating An Application Using Visual C/C++ and NDS-DLL6

To create a data acquisition application using NDS-DLL6 and Microsoft Visual C/C++, follow these steps:

- step 1** Open the project in which you want to use NDS-DLL6. This can be a new or existing project

**step 2** Include header files DLL6.H in the C/C++ source files that call NDS-DLL6 functions. DLL6.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 "DLL6.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 NDS-DLL6's Visual C/C++ import library file DLL6.LIB.

# 3

## Sample Programs

---

### 3.1 Sample Programs Included

There are several sample programs provided in this software. They could help you to program your own applications by using NDS-DLL6 easily. The brief descriptions of these programs are specified as follows:

<b>SDK6060</b>	D/O of ND-6060 Visual C/C++ Program
<b>COMM</b>	General purpose communication Visual Basic Program
<b>VB6011</b>	Functions of ND-6011 Visual Basic Program
<b>VB6050</b>	D/I and D/O ND-6050 Visual Basic Program

---

## 3.2 Sample Programs Development Environment

### 3.2.1 Visual Basic Sample Programs

There are three Visual Basic sample programs provided in this software package. They are located in directory <INSTALLDIR>\SAMPLES\VB. The following files are included in each sample program (Using VB6011 as an example):

- VB project file --- VB6011.VBP
- VB form files --- VB6011.FRM,VB6011.FRX
- Executable file --- VB6011.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.

### 3.2.2 Visual C/C++ Sample Programs

We provide one Visual C/C++ sample program in this package. It is located in directory <INSTALLDIR>\SAMPLES\SDK. The following files are included in the sample program:

- C source file --- SDK6060.C
- Workspace file --- SDK6060.MDP
- Resource script file --- SDK6060.RC, RESOURCE.H
- Make file --- SDK6060.MAK
- Executable file --- SDK6060.EXE

You can use any editor or Microsoft Visual C++ 4.0 or above to view or modify these source files. However, to build the executable SDK6060.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++.

---

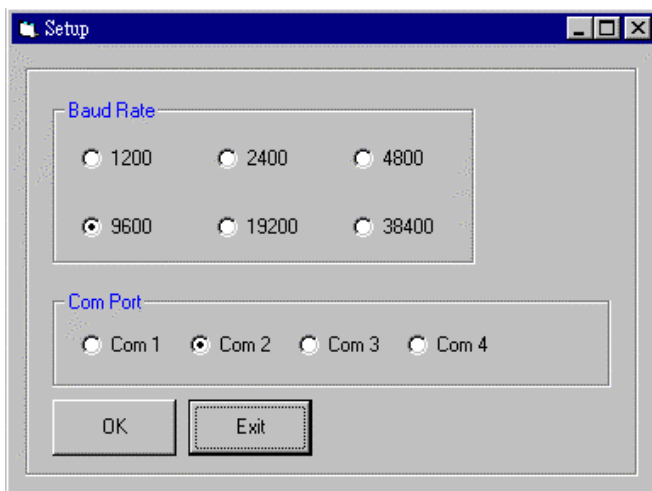
### 3.3 Execute Sample Programs

To run the sample programs, please follow these steps (Using VB6011 as example):

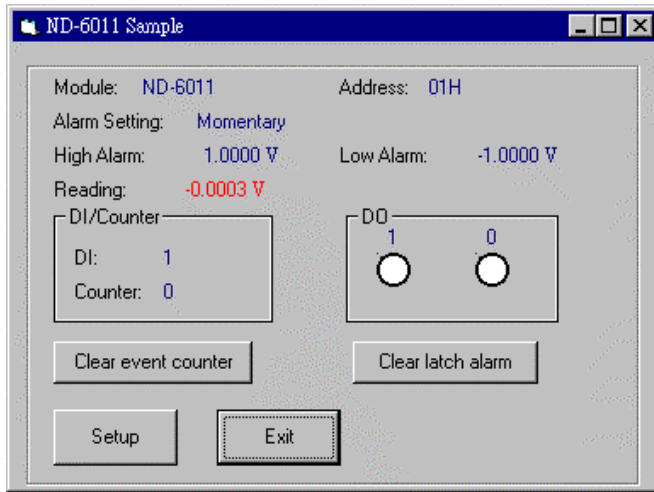
**step 1** Open the sample program

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** The sample program will first prompt the “setup” dialog box (Figure 3.1) for you to configure the COM port. Please select the COM port that connect the NuDAM module and the baud rate to operate.



**Figure 3.1**



**Figure 3.2**

### **step 3** Module Operation

After configuring the COM port, the diagnostic dialog box of the module will appear (Figure 3.2). It displays the information and the functions of the module. You can diagnostic or operate the module by this dialog box.

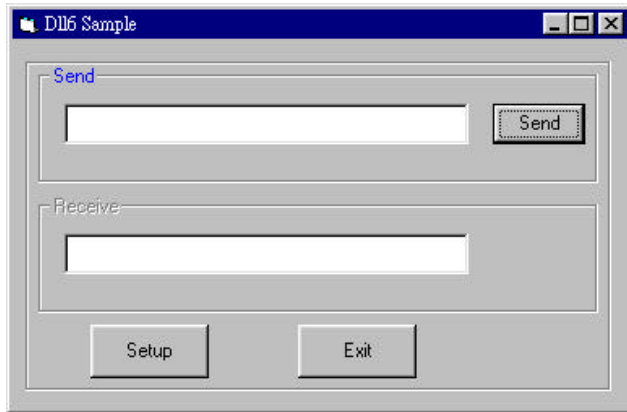
---

## 3.4 The Detail Descriptions of these Sample Programs

### 3.4.1 VB\COMM

This is a general purpose software for RS-232 communication testing. The user interface is shown as figure 3.3 below. You can type any command (a string of character) in the Send dialog box, and press the "send" button to send out the command through the COM port you specified in the "setup" dialog box.

The Receive box will show the message which is got from the COM port. This is a very simple example to show you how to configure the COM port and send/receive data through it.



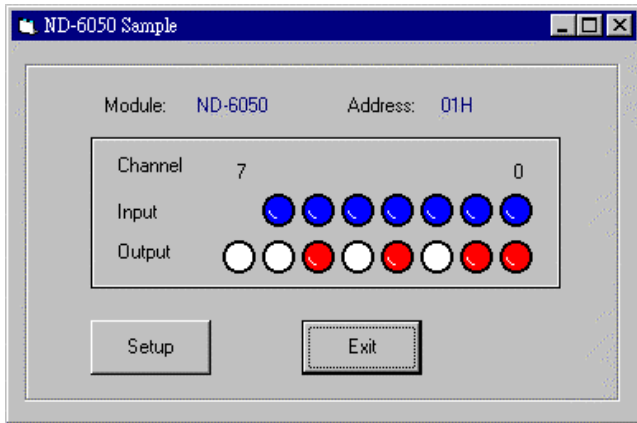
**Figure 3.3**

#### **3.4.2 VB\6011**

This sample program is used to show how to control the ND-6011 multi-function analog input module (Figure 3.2). You have to set the ND-6011's ID address as 01H by configuration utility before running this program. It displays the information and the setting status of the module. Also it periodically read analog input, digital input, and counter values.

#### **3.4.3 VB\6050**

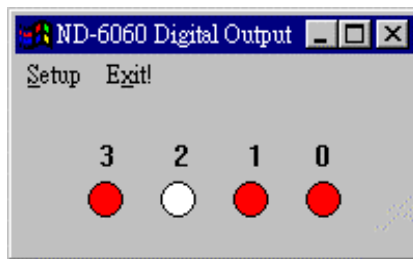
This sample is used to show how to control the ND-6050 8 digital outputs and 7 digital inputs (Figure 3.4). You have to set the ND-6050's ID address as 01H by configuration utility before running this program. You can press the LED image to turn on or turn off the output status. Also, the input LED will indicate the input status.



**Figure 3.4**

#### **3.4.4 SDK\6060**

This sample is used to show how to control the ND-6060 4-channel relay digital outputs (Figure 3.5). You have to set the ND-6060's ID address as 01H by configuration utility before running this program. You can press the LED image to turn on or turn off the output status.



**Figure 3.5**



# Appendix Function Support

This appendix shows which modules each NDS-DLL6 function supports. The functions with an asterisk (\*) are the functions for host or COM port.

Function	Module																
	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	
	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	6	
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
	1	1	1	1	1	1	2	2	5	5	5	5	5	5	6	6	
	1	2	3	4	7	8	1	4	0	2	3	4	6	8	0	3	
	/	/		D													
	D	D															
ND_ClearCtr																	●
ND_ClearLatchAlarm	●	●															
ND_ConfigLeadingCode	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ND_DisableAlarm	●	●		●													●
ND_DisableChecksum*																	
ND_DisableCtrAlarm																	●
ND_EnableAlarm	●	●		●													●
ND_EnableChecksum*																	
ND_EnableCtrAlarm																	●
ND_EnableDigitalFilter																	●
ND_EnableLinearMapping				●													
ND_GetAlarmStatus	●	●		●													
ND_GetConfig	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ND_GetCtrAlarmStatus																	●
ND_GetHighAlarmLimit	●	●		●													
ND_GetLowAlarmLimit	●	●		●													
ND_GetModuleName	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ND_GetVersion	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●

Function	Module															
	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D	N D
ND_GetWDT	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
ND_HostOK*																
ND_InitialComm*																
ND_ReadAI	•	•	•	•												
ND_ReadAllAI			•													
ND_ReadBackAO								•								
ND_ReadBackDO	•	•		•						•					•	•
ND_ReadChanAI					•	•										
ND_ReadCJC	•						•									
ND_ReadCtrAlarmLimit																•
ND_ReadCtrGateMode																•
ND_ReadCtrInitialValue																•
ND_ReadCtrInputMode																•
ND_ReadCtrMax Value																•
ND_ReadCtrOverflowFlag																•
ND_ReadCtrStatus																•
ND_ReadCtrValue																•
ND_ReadDI	•	•		•						•	•	•	•	•		•
ND_ReadEventCount	•	•		•												
ND_ReadFilterStatus																•
ND_ReadMappingSource				•												
ND_ReadMappingTarget				•												
ND_ReadMinWidthAtHigh																•
ND_ReadMinWidthAtLow																•
ND_ReadSyncData	•	•	•	•						•	•	•	•	•		•
ND_ReadTTLInputHigh																•
ND_ReadTTLInputLow																•
ND_ReleaseComm*																

Function	Module																		
	ND1111111111111111	ND2222222222222222	ND3333333333333333	ND4444444444444444	ND7777777777777777	ND8888888888888888	ND1111111111111111	ND2222222222222222	ND5555555555555555	ND0000000000000000	ND2222222222222222	ND3333333333333333	ND4444444444444444	ND6666666666666666	ND8888888888888888	ND0000000000000000	ND3333333333333333	ND8888888888888888	
ND_ResetEventCount	●	●		●															
ND_ResetStatusAO								●	●										
ND_SendCommand*																			
ND_SetConfig	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ND_SetCtrAlarmLimit																			●
ND_SetCtrGateMode																			●
ND_SetCtrInitialValue																			●
ND_SetCtrInputMode																			●
ND_SetCtrMax Value																			●
ND_SetHighAlarmLimit	●	●		●															
ND_SetLowAlarmLimit	●	●		●															
ND_SetMinWidthAtHigh																			●
ND_SetMinWidthAtLow																			●
ND_SetMUXChans			●		●	●													
ND_SetTTLInputHigh																			●
ND_SetTTLInputLow																			●
ND_SetWDT	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●	●
ND_StartCtr																			●
ND_Sync*																			
ND_TimeOut*																			
ND_WriteAO								●											
ND_WriteChanAO									●										
ND_WriteDOLine										●			●	●	●	●			
ND_WriteDOPort	●	●		●						●			●	●	●	●	●		
ND_WriteMappingSource				●															
ND_WriteMappingTarget				●															