PCI-Lanner

Linux Software Manual

Warranty

All products manufactured by ICP DAS are warranted against defective materials for a period of one year from the date of delivery to the original purchaser.

Warning

ICP DAS assume no liability for damages consequent to the use of this product. ICP DAS reserves the right to change this manual at any time without notice. The information furnished by ICP DAS is believed to be accurate and reliable. However, no responsibility is assumed by ICP DAS for its use, nor for any infringements of patents or other rights of third parties resulting from its use.

Copyright

Trademark

The names used for identification only may be registered trademarks of their respective companies.

Tables of Content

1. Linux UART Driver Installation

The PCI Lanner UART driver can be used in linux kernel 2.6.X or later kernel version. For Linux O.S, the recommended installation and uninstall steps are given in Sec 1.1 ~ 1.2

1.1 Linux UART Driver Installing Procedure

Step 1: Download the linux driver “ixcom.tar.gz” (version 0.8.7 or the later ixcom package version) from ICP DAS webpage http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/multiport/linux/ixcom.tar.gz to the linux host.

Step 2: You must use the ‘**root**’ identity to compile and install linux UART driver.

Step 3: Decompress the software package “ixcom.tar.gz”.

Step 4: Type ‘**cd**' to the directory containing the package's source code and type ‘**./configure**' to configure the package for your linux system.

Step 5: Type ‘**make**' to compile the package.

Step 6: You can type ‘**./ixcom.inst**' to install the PCI Lanner UART driver Module and build the device interface “**ttySV***”.

Step 7: You can type ‘**dmesg**' to check the UART interface. Please refer to the Figure 1-1 (the figure show the information of UART interface).

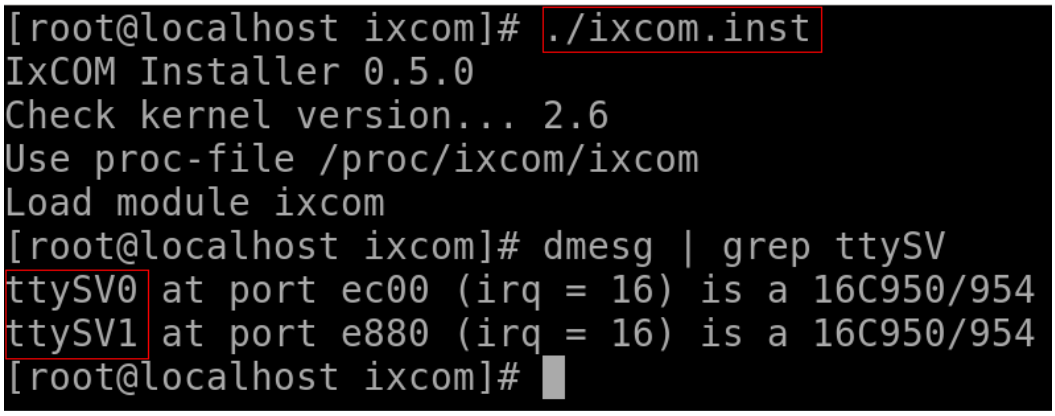

Figure 1-1

1.2 Linux Driver Uninstalling Procedure

Step 1: Type '**cd**' to the directory containing the package's source code.

Step 2: Type '**./ixcom.remove**' to remove the UART driver module.

2. Linux PCI I/O Driver Installation

The PCI Lanner I/O driver can be used in linux kernel 2.6.X or later kernel version. For Linux O.S, the recommended installation and uninstall steps are given in Sec 2.1 ~ 2.2

2.1 Linux PCI I/O Driver Installing Procedure

Step 1: Download the linux driver “ixpci.tar.gz” (version 0.7.10 or the later ixpci package version) from ICP DAS webpage http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/linux/ixpci.tar.gz to the linux host.

Step 2: You must use the ‘**root**’ identity to compile and install linux I/O driver.

Step 3: Decompress the software package “**ixpci.tar.gz**”.

Step 4: Type ‘**cd**' to the directory containing the package's source code and type ‘**./configure**' to configure the package for your linux system.

Step 5: Type ‘**make**' to compile the package.

Step 6: Type `.**/ixpci.inst**' to install the PCI driver module and build the device file “**ixpci***” in the device directory “**/dev**” automatically.

2.2 Linux Driver Uninstalling Procedure

Step 1: Type ‘**cd**’ to the directory containing the package's source code.

Step 2: Type ‘**./ixpci.remove**' to remove the PCI I/O driver module.

3. Linux PCI I/O Library Function Description

The static library is the collection of function calls of the PCI-Lanner I/O cards for linux kernel 2.6.X(or later kernel version) system. The application structure is presented as following figure 3-1. The user application program developed by C(C++) language can call library “libpci.a” in user mode. And then static library will call the PCI Lanner I/O modules to access the hardware system.

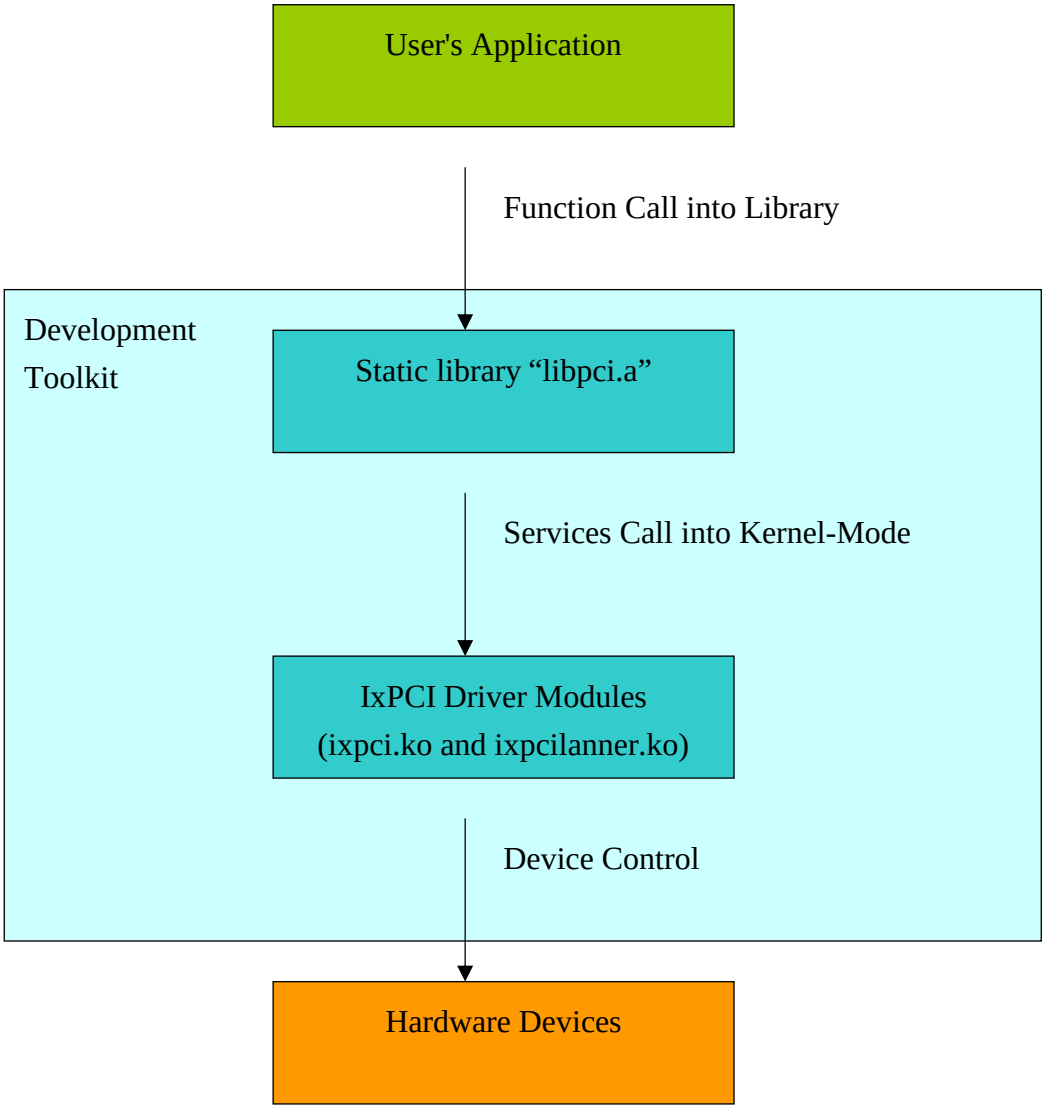

Figure 3-1

3.1 Table of Error Code and Error ID

Error Code	Error ID	Error String
0	PCIDA_NOERROR	OK (No error !)
1	PCIDA_MODULE_NAME_GET_ERROR	Get IxPCI device name failure
2	PCIDA_LANNER_DIGITAL_OUTPUT_ERROR	Digital output failure
3	PCIDA_LANNER_DIGITAL_INPUT_ERROR	Digital input failure
4	PCIDA_LANNER_COUNTER_NUMBER_ERROR	Counter channel over range
5	PCIDA_LANNER_READ_COUNTER_ERROR	Read count failure
6	PCIDA_LANNER_CLEAR_COUNTER_ERROR	Clear counter failure
7	PCIDA_LANNER_SET_CHANNEL_GAIN_ERROR	Set MUX failure
8	PCIDA_LANNER_SET_POLLING_ERROR	Set Polling mode failure
9	PCIDA_LANNER_ANALOG_INPUT_ERROR	Read AI failure
10	PCIDA_LANNER_ANALOG_CHANNEL_ERROR	Channel number over range
11	PCIDA_LANNER_ANALOG_GAIN_ERROR	Gain number over range
12	PCIDA_LANNER_EEPROM_ADDR_ERROR	EEPROM address over range
13	PCIDA_LANNER_EEPROM_WRITE_ERROR	Set EEPROM failure

Table 3.1

3.2 Function Descriptions

Function Definition
int PCIDA_Open(char *dev_file);

Function Definition
WORD PCIDA_Close(WORD fd);
WORD PCIDA_DriverInit(WORD fd);
char* PCIDA_GetDriverVersion(void);
char* PCIDA_GetLibraryVersion(void);
WORD PCI_LANNER_Digital_Output(WORD fd, BYTE data);
WORD PCI_LANNER_Digital_Input(WORD fd, BYTE *di_data);
WORD PCI_LANNER_Read_Count(WORD fd, WORD count, DWORD *count_value);
WORD PCI_LANNER_Clear_Count(WORD fd, WORD count);
WORD PCI_LANNER_Set_Voltate_Gain_MUX(WORD fd, WORD channel, WORD gain);
WORD PCI_LANNER_Set_Current_Gain_MUX(WORD fd, WORD channel);
WORD PCI_LANNER_ReadAI_Hex(WORD fd, WORD *hex_value);
WORD PCI_LANNER_Read_Voltage(WORD fd, WORD channel, WORD gain, float *fvalue);
WORD PCI_LANNER_Read_CalVoltage(WORD fd, WORD channel, WORD gain, float *fvalue);
WORD PCI_LANNER_Read_Current(WORD fd, WORD channel, float *fvalue);
WORD PCI_LANNER_Read_CalCurrent(WORD fd, WORD channel, float *fvalue);
WORD PCI_LANNER_EEPROM_WriteEnable(WORD fd);
WORD PCI_LANNER_EEPROM_WriteDisable(WORD fd);
WORD PCI_LANNER_EEPROM_WriteWord(WORD fd, WORD addr, WORD value);
WORD PCI_LANNER_EEPROM_ReadWord(WORD fd, WORD addr, WORD *value);

Function Definition
WORD PCI_LANNER_Read_Voltage_Polling(WORD fd, WORD channel, WORD gain, DWORD datacount, float *voltage)
WORD PCI_LANNER_Read_Current_Polling(WORD fd, WORD channel, WORD gain, DWORD datacount, float *current)

Table 3.2

3.3 Linux PCI I/O Library FUNCTIONS

3.3.1 PCIDA_Open

- **Description:**
  To open IxPCI I/O device file.
- **Syntax:**
  int PCIDA_Open(char *dev_file)
- **Parameter:**
  dev_file : The path of IxPCI I/O device file
- **Return:**
  The file descriptor of device file. If the file descriptor < 0, it means that open device file failure.

3.3.2 PCIDA_Close

- **Description:**
  To close device file..
- **Syntax:**
  WORD PCIDA_Close(WORD fd)
- **Parameter:**
  fd : The file descriptor of device file that get from function PCIDA_Open.
- **Return:**
  “PCIDA_NOERROR”
  The code “PCIDA_NOERROR”(Please refer to "Section 3.1 Error Code").

3.3.3 PCIDA_DriverInit

- **Description:**
  To allocates the computer resource for the device. This function must be called once before applying other PCIDA functions.
- **Syntax:**
  WORD PCIDA_DriverInit(WORD fd)
- **Parameter:**
  fd : The file descriptor of device file that get from function PCIDA_Open.
- **Return:**
  “PCIDA_MODULE_NAME_GET_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.4 PCIDA_GetDriverVersion

- **Description :**
  To get the version of PCI linux driver.
- **Syntax :**
  char* PCIDA_GetDriverVersion(void)
- **Parameter :**
  none.
- **Return:**
  The version of IxPCI Driver.

3.3.5 PCIDA_GetLibaryVersion

- **Description :**
  To get the version of IxPCI library.
- **Syntax :**
  char* PCIDA_GetLibraryVersion(void);
- **Parameter :**
  none.
- **Return:**
  The version of IxPCI library.

3.3.6 PCI_LANNER_Digital_Output

- **Description :**

To output DO data.

- **Syntax :**

  WORD PCI_LANNER_Digital_Output(WORD fd, BYTE data)

- **Parameter :**

  fd : The file descriptor of device file that get from function PCIDA_Open.

  data : The DO data.

- **Return:**

  “PCIDA_LANNER_DIGITAL_OUTPUT_ERROR”

  “PCIDA_NOERROR”

  Please refer to "Section 3.1 Error Code"

3.3.7 PCI_LANNER_Digital_Input

- **Description :**

  To get DI data.

- **Syntax :**

  WORD PCI_LANNER_Digital_Input(WORD fd, BYTE *di_data)

- **Parameter :**

  fd : The file descriptor of device file that get from function PCIDA_Open.

  di_data : A variable address used to storage the digital input data.

- **Return:**

  “PCIDA_LANNER_DIGITAL_INPUT_ERROR”

  “PCIDA_NOERROR”

  Please refer to "Section 3.1 Error Code"

3.3.8 PCI_LANNER_Read_Count

- **Description :**

  To read counter data.

- **Syntax :**

  WORD PCI_LANNER_Read_Count(WORD fd, WORD count, DWORD *count_value)

- **Parameter :**

  fd : The file descriptor of device file that get from function PCIDA_Open.

  count : The count channel(COUNT0 or COUNT1).

count_value : A variable address used to storage the counter data

- **Return:**

  “PCIDA_LANNER_COUNTER_NUMBER_ERROR”
  “PCIDA_LANNER_READ_COUNTER_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.9 PCI_LANNER_Clear_Count

- **Description :**

  To clear Counter data.

- **Syntax :**

  WORD PCI_LANNER_Clear_Count(WORD fd, WORD count);

- **Parameter :**

  fd : The file descriptor of device file that get from function
  PCIDA_Open.
  count : The count channel(COUNT0 or COUNT1).

- **Return:**

  “PCIDA_LANNER_CLEAR_COUNTER_ERROR”
  “PCIDA_LANNER_COUNTER_NUMBER_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.10 PCI_LANNER_Set_Voltate_Gain_MUX

- **Description :**

  To set AI(Voltage) channel and gain mode.

- **Syntax :**

  WORD PCI_LANNER_Set_Voltate_Gain_MUX(WORD fd, WORD channel, WORD gain)

- **Parameter :**

  fd : The file descriptor of device file that get from function
  PCIDA_Open.
  channel : The AI channel number(0, 1, 2, 3).
  gain : 0 → +/- 10V
  1 → +/- 5V
  2 → +/- 2.5V
  3 → +/- 1.25V

- **Return:**
  “PCIDA_LANNER_ANALOG_CHANNEL_ERROR”
  “PCIDA_LANNER_ANALOG_GAIN_ERROR”
  “PCIDA_LANNER_SET_CHANNEL_GAIN_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.11 PCI_LANNER_Set_Current_Gain_MUX

- **Description :**
  To set AI(Current) channel and gain mode..
- **Syntax :**
  WORD PCI_LANNER_Set_Current_Gain_MUX(WORD fd, WORD channel)
- **Parameter :**
  fd : The file descriptor of device file that get from function PCIDA_Open.
  channel : The AI channel number(0, 1, 2, 3).
- **Return:**
  “PCIDA_LANNER_ANALOG_CHANNEL_ERROR”
  “PCIDA_LANNER_SET_CHANNEL_GAIN_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.12 PCI_LANNER_ReadAI_Hex

- **Description :**
  To get AI hex data.
- **Syntax :**
  WORD PCI_LANNER_ReadAI _Hex(WORD fd, WORD *hex_value);
- **Parameter :**
  fd : The file descriptor of device file that get from function PCIDA_Open.
  hex_value : A variable address used to storage the AI hex data.
- **Return:**
  “PCIDA_LANNER_SET_POLLING_ERROR”
  “PCIDA_LANNER_ANALOG_INPUT_ERROR“
  “PCIDA_NOERROR”

Please refer to "Section 3.1 Error Code"

3.3.13 PCI_LANNER_Read_Voltage

- **Description :**
  To get voltage data without calibrating.
- **Syntax :**
  WORD PCI_LANNER_Read _Voltage(WORD fd, WORD channel, WORD gain, float *fvalue);
- **Parameter :**
  fd : The file descriptor of device file that get from function PCIDA_Open.
  channel : The AI channel number(0, 1, 2, 3).
  gain : 0 → +/- 10V
  1 → +/- 5V
  2 → +/- 2.5V
  3 → +/- 1.25V
  fvalue : A variable address used to storage the voltage data.
- **Return:**
  “PCIDA_LANNER_SET_POLLING_ERROR”
  “PCIDA_LANNER_ANALOG_INPUT_ERROR“
  “PCIDA_LANNER_ANALOG_CHANNEL_ERROR”
  “PCIDA_LANNER_ANALOG_GAIN_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.14 PCI_LANNER_Read_CalVoltage

- **Description :**
  To get calibrated voltage data.
- **Syntax :**
  WORD PCI_LANNER_Read _CalVoltage(WORD fd, WORD channel, WORD gain, float *fvalue);
- **Parameter :**
  fd : The file descriptor of device file that get from function PCIDA_Open.
  channel : The AI channel number(0, 1, 2, 3).
  gain : 0 → +/- 10V
  1 → +/- 5V

2 → +/- 2.5V

3 → +/- 1.25V

fvalue : A variable address used to storage the voltage data.

- **Return:**

"PCIDA_LANNER_SET_POLLING_ERROR"

"PCIDA_LANNER_ANALOG_INPUT_ERROR"

"PCIDA_LANNER_ANALOG_CHANNEL_ERROR"

"PCIDA_LANNER_ANALOG_GAIN_ERROR"

"PCIDA_NOERROR"

Please refer to "Section 3.1 Error Code"

3.3.15 PCI_LANNER_Read_Current

- **Description :**

To get current data without calibrating.

- **Syntax :**

WORD PCI_LANNER_Read_Current(WORD fd, WORD channel, float *fvalue)

- **Parameter :**

fd : The file descriptor of device file that get from function PCIDA_Open.

channel : The AI channel number(0, 1, 2, 3).

fvalue : A variable address used to storage the current data.

- **Return:**

"PCIDA_LANNER_SET_POLLING_ERROR"

"PCIDA_LANNER_ANALOG_INPUT_ERROR"

"PCIDA_LANNER_ANALOG_CHANNEL_ERROR"

"PCIDA_NOERROR"

Please refer to "Section 3.1 Error Code"

3.3.16 PCI_LANNER_Read_CalCurrent

- **Description :**

To get calibrated current data.

- **Syntax :**

WORD PCI_LANNER_Read_Current(WORD fd, WORD channel, float *fvalue)

- **Parameter :**

fd : The file descriptor of device file that get from function

PCIDA_Open.

channel : The AI channel number(0, 1, 2, 3).

fvalue : A variable address used to storage the current data.

- **Return:**

"PCIDA_LANNER_SET_POLLING_ERROR"

"PCIDA_LANNER_ANALOG_INPUT_ERROR"

"PCIDA_LANNER_ANALOG_CHANNEL_ERROR"

"PCIDA_NOERROR"

Please refer to "Section 3.1 Error Code"

3.3.17 PCI_LANNER_EEPROM_WriteEnable

- **Description :**

To enable EEPROM.

- **Syntax :**

WORD PCI_LANNER_EEPROM_WriteEnable(WORD fd)

- **Parameter :**

fd : The file descriptor of device file that get from function PCIDA_Open.

- **Return:**

"PCIDA_NOERROR"

Please refer to "Section 3.1 Error Code"

3.3.18 PCI_LANNER_EEPROM_WriteDisable

- **Description :**

To disable EEPROM.

- **Syntax :**

WORD PCI_LANNER_EEPROM_WriteDisable(WORD fd)

- **Parameter :**

fd : The file descriptor of device file that get from function PCIDA_Open.

- **Return:**

"PCIDA_NOERROR"

Please refer to "Section 3.1 Error Code"

3.3.19 PCI_LANNER_EEPROM_WriteWord

- **Description :**

To write data to EEPROM.

- **Syntax :**
  WORD PCI_LANNER_EEPROM_WriteWord(WORD fd, WORD addr, WORD value)
- **Parameter :**
  fd : The file descriptor of device file that get from function PCIDA_Open.
  addr : The EEPROM block address(0~63).
  value : The data that write to EEPROM.
- **Return:**
  “PCIDA_LANNER_EEPROM_ADDR_ERROR”
  “PCIDA_LANNER_EEPROM_WRITE_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.20 PCI_LANNER_EEPROM_ReadWord

- **Description :**
  To read EEPROM data.
- **Syntax :**
  WORD PCI_LANNER_EEPROM_ReadWord(WORD fd, WORD addr, WORD *value)
- **Parameter :**
  fd : The file descriptor of device file that get from function PCIDA_Open.
  addr : The EEPROM block address(0~63).
  value : The data that write to EEPROM.
- **Return:**
  “PCIDA_LANNER_EEPROM_ADDR_ERROR”
  “PCIDA_NOERROR”
  Please refer to "Section 3.1 Error Code"

3.3.21 PCI_LANNER_Read_Voltage_Polling

- **Description :**
  To use polled operation to read AI(voltage) data.
- **Syntax :**
  WORD PCI_LANNER_Read_Voltage_Polling(WORD fd, WORD

channel, WORD gain, DWORD datacount, float *voltage)

- **Parameter :**

  fd : The file descriptor of device file that get from function PCIDA_Open.

  channel : The AI channel number(0, 1, 2, 3).

  gain : 0 → +/- 10V

  1 → +/- 5V

  2 → +/- 2.5V

  3 → +/- 1.25V

  datacount : the polling count(max count : 32786).

  voltage : A float array used to storage the voltage data.

- **Return:**

  “PCIDA_LANNER_ANALOG_CHANNEL_ERROR”

  “PCIDA_LANNER_ANALOG_GAIN_ERROR”

  “PCIDA_LANNER_SET_CHANNEL_GAIN_ERROR”

  “PCIDA_LANNER_SET_POLLING_ERROR”

  “PCIDA_NOERROR”

  Please refer to "Section 3.1 Error Code"

3.3.22 PCI_LANNER_Read_Current_Polling

- **Description :**

  To use polled operation to read AI(Current) data.

- **Syntax :**

  WORD PCI_LANNER_Read_Current_Polling(WORD fd, WORD channel, DWORD datacount, float *current)

- **Parameter :**

  fd : The file descriptor of device file that get from function PCIDA_Open

  channel : The AI channel number(0, 1, 2, 3).

  datacount : the polling count(max count : 32786).

  voltage : A float array used to storage the current data.

- **Return:**

  “PCIDA_LANNER_ANALOG_CHANNEL_ERROR”

  “PCIDA_LANNER_SET_CHANNEL_GAIN_ERROR”

  “PCIDA_LANNER_SET_POLLING_ERROR”

  “PCIDA_NOERROR”

  Please refer to "Section 3.1 Error Code"

4. Linux PCI I/O Demo

All of demo programs will not work normally if IxPCI I/O driver would not be installed correctly. After driver (version 0.7.10 or the later driver version) compiled and installation, the related IxPCI I/O library, demo and header files for different development environments are presented as follows.

Table 4.1

Package Name	Directory Path	File Name	Description
ixpci-0.7.10	include	pcidio.h	The IxPCI I/O library header
	lib	libpci.a	The static library of IxPCI I/O.
	examples/pcilanner	Digital I/O Demo (dio.c, do_readback.c, dio_a.c)	Digital I/O demo for PCI Lanner card
		Count Demo (count.c count_a.c)	Counter demo for PCI Lanner card
		Analog Input Demo (ai_voltage.c, ai_voltage_a.c, ai_current.c, ai_current_a.c, precision_test.c, ad_polling_a.c, ad_scan_a.c)	The Analog input demo.for PCI Lanner card
		EEPROM Demo (eeprom_ead_a.c)	EEPROM demo for PCI Lanner card