PISO-P16R16
PEX-P8R8/P16R16

Digital I/O Card

Linux Software Manual

Tables of Content

1. Linux Software Installation

The PIO-D48 can be used in Linux kernel 2.4.X to 4.15.X. For Linux O.S, the recommended installation and uninstall steps are given in Sec 1.1 ~ 1.2

1.1 Linux Driver Installing Procedure

Step 1: Copy the Linux driver “ixpio.tar.gz” in the directory “NAPDOS\Linux” of the companion CD or download the latest driver from our website to the Linux host.

Step 2: You must use the ‘**root**’ identity to compile and install PIO/PISO Linux driver.

Step 3: Decompress the tarball “ixpio.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 ‘**./ixpio.inst**' to install the PIO/PISO driver module and build the device file “ixpioX” in the device directory “/dev” automatically.

1.2 Linux Driver Uninstalling Procedure

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

Step 2: Type `./ixpio.remove' to remove the PIO/PISO driver module.

2. Static Library Function Description

The static library is the collection of function calls of the PIO-DIO cards for linux kernel 2.4.x and 2.6.x to 4.15.x system. The application structure is presented as following figure. The user application program developed by C (C++) language can call library “libpio.a” in user mode. And then static library will call the module ixpio to access the hardware system.

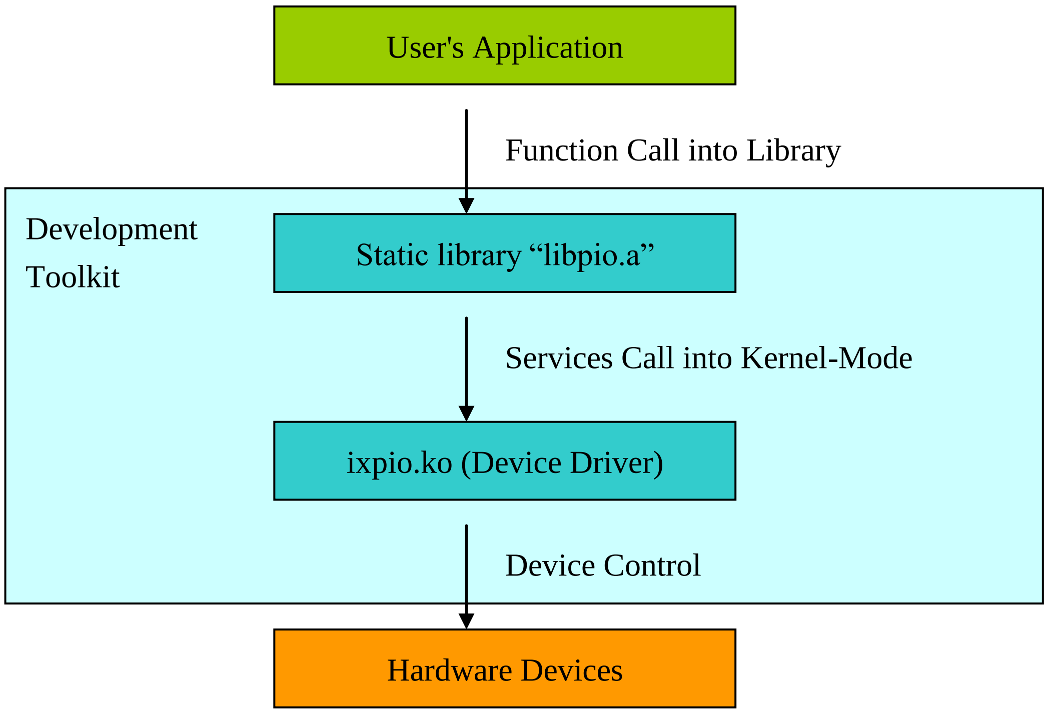


Figure 2.1

2.1 Table of ErrorCode and ErrorString

Table 2.1

Error Code	Error ID	Error String
0	PIODA_NOERROR	OK (No error!)
1	PIODA_MODULE_NAME_GET_ERROR	Module name can't get from file /proc/ixpio/ixpio
4	PIODA_PORT_DEFINED_ERROR	Port number out of range
5	PIODA_DIGITAL_OUTPUT_ERROR	Digital output error
6	PIODA_DIGITAL_INPUT_ERROR	Digital input error
23	PIODA_CARD_ID_ERROR	Read Card ID error

2.2 Function Descriptions

Table 2.2

Function Definition
char * PIODA_GetDriverVersion(void);
char * PIODA_GetLibraryVersion(void);
int PIODA_Open(char *dev_file);
WORD PIODA_Close(WORD fd);
WORD PIODA_DriverInit(WORD);
WORD PIODA_Digital_Output(WORD, WORD, byte);
WORD PIODA_Digital_Input(WORD, WORD, WORD *);
WORD PIODA_CARD_ID(WORD fd, byte *id);

2.3 PISO-P16R16/PEX-P8R8/P16R16 FUNCTIONS

2.3.1 PIODA_GetDriverVersion

- **Description:**
  To show the version number of PIO/PISO linux driver.
- **Syntax:**
  char * PIODIO_GetDriverVersion(Void)
- **Parameter:**
  None
- **Return:**
  The version of PIO/PISO linux driver version.

2.3.2 PIODA_GetLibraryVersion

- **Description:**
  To show the version number of PIO/PISO linux static library.
- **Syntax:**
  char * PIODIO_GetLibraryVersion(void)
- **Parameter:**
  None
- **Return:**
  PIO/PISO linux static library version.

2.3.3 PIODA_Open

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

2.3.4 PIODA_Close

- **Description :**

  To close device file.

- **Syntax :**

  Word PIODIO_Close(WORD fd)

- **Parameter :**

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

- **Return:**

  “PIODA_NOERROR”

  (Please refer to "Section 2.1 Error Code")

2.3.5 PIODA_DriverInit

- **Description :**

  To allocates the computer resource for the device. This function must be called once before applying other PIODA functions.

- **Syntax :**

  WORD PIODA_DriverInit(WORD fd)

- **Parameter :**

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

- **Return:**

  “PIODA_MODULE_NAME_GET_ERROR”

  “PIODA_NOERROR”

  (Please refer to "Section 2.1 Error Code")

2.3.6 PIODA_Digital_Output

- **Description :**

  This subroutine sends the 8 bits data to the specified I/O port.

- **Syntax :**

  WORD PIODA_Digital_Output(WORD fd, WORD port, byte data);

- **Parameter :**

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

  port : The output port number.

  PISOP16R16_DIOA: DO0~DO7

  PISOP16R16_DIOA: DO8~DO15

  PISOP16R16_DIO_ALL: DO0~DO15

data : 8 bits data.

- **Return:**
  “PIODA_PORT_DEFINED_ERROR”
  “PIODA_DIGITAL_OUTPUT_ERROR”
  “PIODA_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.7 PIODA_Digital_Input

- **Description :**
  This subroutine reads the 8 bits data from the specified I/O port.
- **Syntax :**
  WORD PIODA_Digital_Input(WORD fd, WORD port, WORD *di_data);
- **Parameter :**
  fd : The file descriptor of device file that get from function PIODIO_Open.
  port : The input port number.
  PISOP16R16_DIOA: DI0~DI7
  PISOP16R16_DIOA: DI8~DI15
  PISOP16R16_DIO_ALL: DI0~DI15
  di_data : A variable address used to storage the 8 bits input data.
- **Return:**
  “PIODA_PORT_DEFINED_ERROR“
  “PIODA_DIGITAL_INPUT_ERROR“
  “PIODA_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.8 PIODA_CARD_ID

- **Description :**
  This subroutine can get Card ID.
  The Card ID can be set using the SW1 dip switch, so it is easy to set the correct connections between cards and devices. So, by reading the Card ID users can check whether their program is accessing the correct card.
- **Syntax :**
  WORD PISOP16R16_Card_ID(WORD fd, byte *id)
- **Parameter :**
  fd : The file descriptor of device file that get from function PIODIO_Open.
  id : The number of Card ID.

- **Return:**
  “PIODA_CARD_ID_ERROR”,
  “PIODA_NOERROR”.
  (Please refer to "Section 2.1 Error Code")

3. PISO-P16R16/PEX-P8R8/P16R16 Linux Demo

All of demo programs will not work normally if PIO/PISO linux driver would not be installed correctly. During the installation process of PIO/PISO linux driver, the install-scripts “ixpio.inst” will setup the correct kernel driver. After driver (version 0.23.1 or the later driver version) compiled and installation, the related demo programs, development library and declaration header files for different development environments are presented as follows.

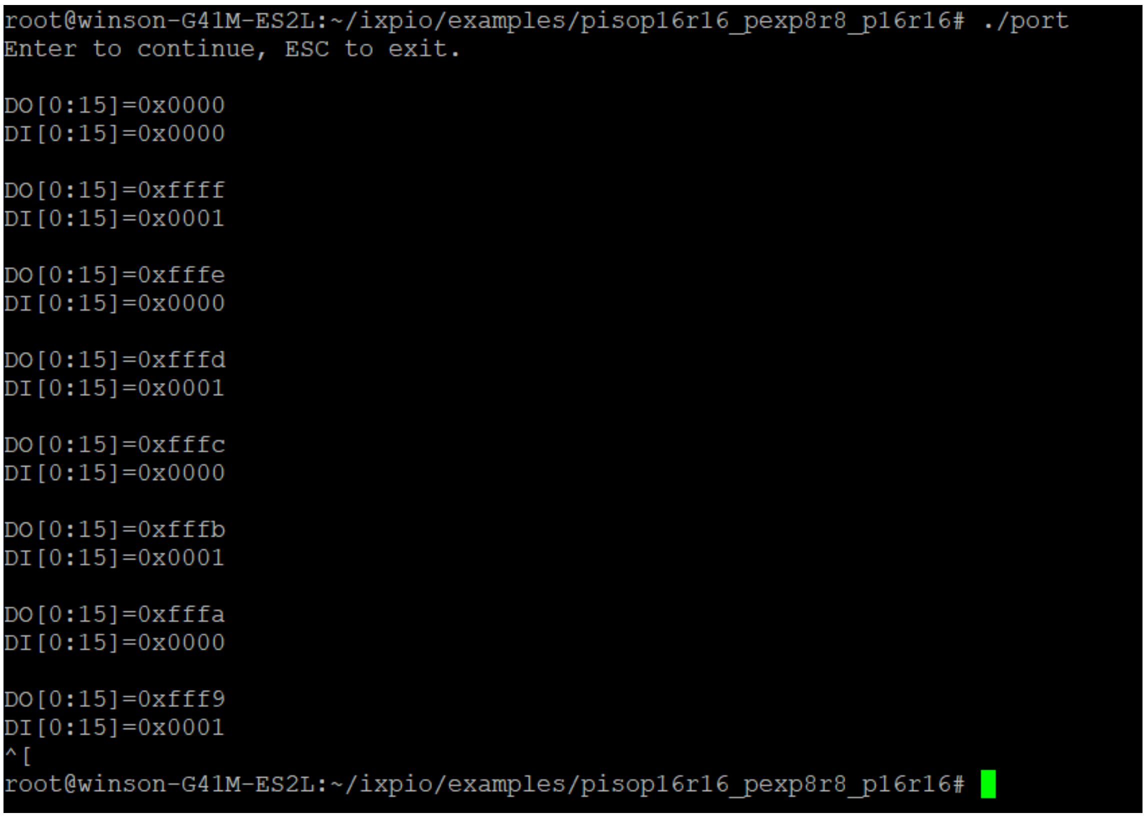
Table 3.1

Driver Name	Directory Path	File Name	Description
ixpio	Include	piodio.h	PIO/PISO library header
	lib	libpio.a libpio_64.a	PIO/PISO static library
	examples/ pisop16r16_pexp8r8_p16r16	port.c	DI and DO demo (ch0~ch15)
		port2.c	DI and DO demo (ch8~ch15)
		port_a.c	DI and DO demo with library
		card_id.c	Read Card ID
		card_id_a.c	Read Card ID with library

3.1 Demo code “port.c”

This demo program is used to output data to DO0~DO15 and read data from DI0~DI15

In Figure 3.1, use DI0 to test. When DI0 read value, then present 0x1, else is 0.



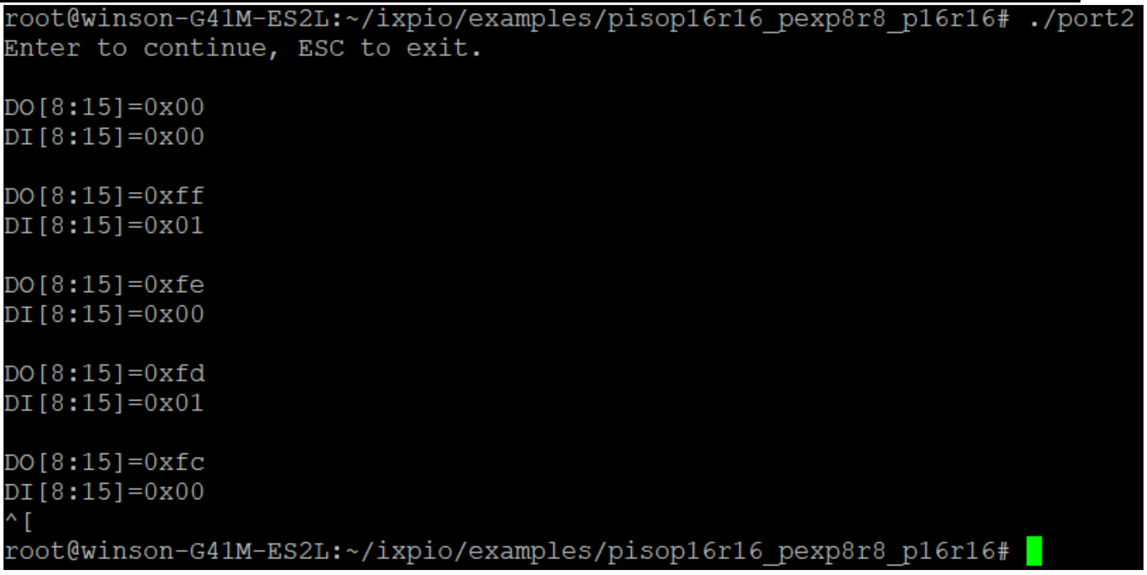
```
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16# ./port
Enter to continue, ESC to exit.

DO[0:15]=0x0000
DI[0:15]=0x0000

DO[0:15]=0xffff
DI[0:15]=0x0001

DO[0:15]=0xfffe
DI[0:15]=0x0000

DO[0:15]=0xfffd
DI[0:15]=0x0001

DO[0:15]=0xfffc
DI[0:15]=0x0000

DO[0:15]=0xfffb
DI[0:15]=0x0001

DO[0:15]=0xfffa
DI[0:15]=0x0000

DO[0:15]=0xfff9
DI[0:15]=0x0001
^[
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16#
```

Figure 3.1

3.2 Demo code “port2.c”

This demo program is used to output data to DO8~DO15 and read data from DI8~DI15

In Figure 3.2, use DI0 to test. When DI0 read value, then present 0x1, else is 0



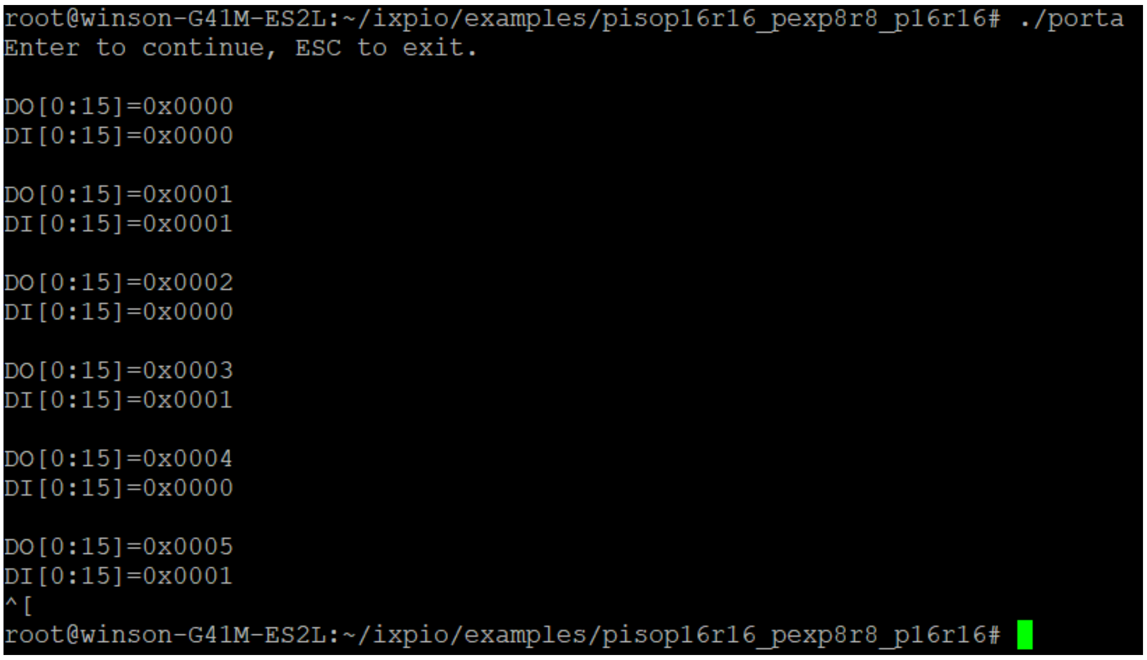
```
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16# ./port2
Enter to continue, ESC to exit.

DO[8:15]=0x00
DI[8:15]=0x00

DO[8:15]=0xff
DI[8:15]=0x01

DO[8:15]=0xfe
DI[8:15]=0x00

DO[8:15]=0xfd
DI[8:15]=0x01

DO[8:15]=0xfc
DI[8:15]=0x00
^[
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16#
```

Figure 3.2

3.3 Demo code "port_a.c"

This demo is using the static library. Output data to DO0~DO15 and read data from DI0~DI15

In Figure 3.3 use DI0 to test. When DI0 read value, then present 0x1, else is 0.

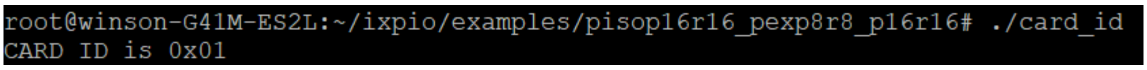


```
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16# ./porta
Enter to continue, ESC to exit.

DO[0:15]=0x0000
DI[0:15]=0x0000

DO[0:15]=0x0001
DI[0:15]=0x0001

DO[0:15]=0x0002
DI[0:15]=0x0000

DO[0:15]=0x0003
DI[0:15]=0x0001

DO[0:15]=0x0004
DI[0:15]=0x0000

DO[0:15]=0x0005
DI[0:15]=0x0001
^[
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16#
```

Figure 3.3

3.4 Demo code “card_id.c”

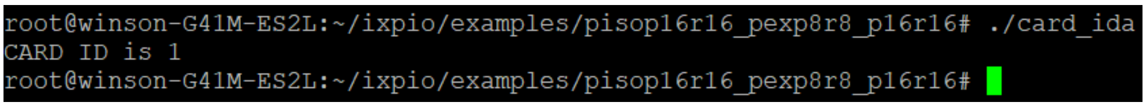
Read Card ID.



```
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16# ./card_id
CARD ID is 0x01
```

3.5 Demo code “card_id_a.c”

This demo is using the static library. Read Card ID.



```
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16# ./card_ida
CARD ID is 1
root@winson-G41M-ES2L:~/ixpio/examples/pisop16r16_pexp8r8_p16r16#
```