PISO-A64/C64/P64

User 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 Contents

1. Linux Software Installation

The PISO-A64/C64/P64 can be used in linux kernel 2.4.X and 2.6.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-0.20.4.tar.gz”(or the later driver version) in the directory “NAPDOS\Linux” of the companion CD to the linux host that you want to install driver.

Step 2: Decompress the tarball “ixpio-0.20.4.tar.gz”.

Step 3: Type `cd' to the directory containing the package's source code and type ./configure' to configure the package for your system.

Step 4: Type `make' to compile the package.

Step 5: 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 Libary Function Description

The static libary is the collection of function calls of the PIO-DIO cards for linux kernel 2.4.x and 2.6.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 libary will call the module ixpio to access the hardware system.

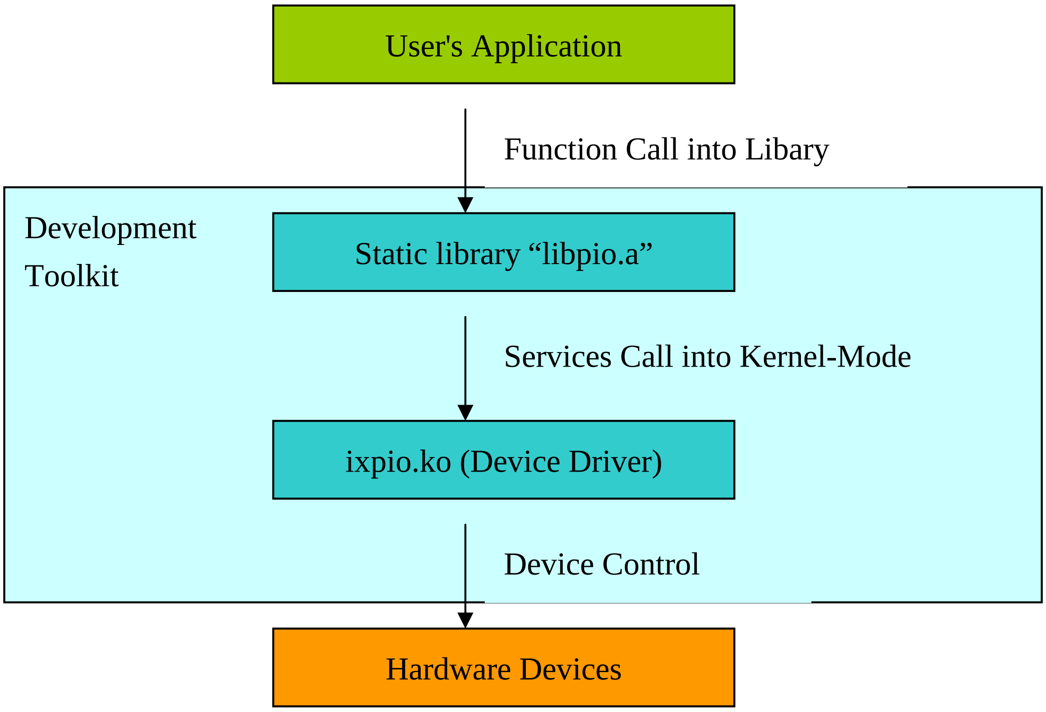

Figure 2.1

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
5	PIODA_DIGITAL_OUTPUT_ERROR	Digital output error
6	PIODA_DIGITAL_INPUT_ERROR	Digital input error

2.2 Function Descriptions

Table 2.2

Function Definition
WORD PIODA_GetDriverVersion(void);
WORD PIODA_GetLibaryVersion(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 *);

2.3 Digital I/O FUNCTIONS

2.3.1 PIODA_GetDriverVersion

- **Description:**

  To show the version number of PIO/PISO linux driver.

- **Syntax:**

  WORD PIODIO_GetDriverVersion(Void)

- **Parameter:**

  None

- **Return:**

  The code “PIODA_NOERROR”(Please refer to "Section 2.1 Error Code")

2.3.2 PIODA_GetLibaryVersion

- **Description:**
  To show the version number of PIO/PISO linux static libary..
- **Syntax:**
  WORD PIODIO_GetLibaryVersion(void)
- **Parameter:**
  None
- **Return:**
  The code “PIODA_NOERROR”(Please refer to "Section 2.1 Error Code")

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:**
  The code “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:**

  The code "PIODA_MODULE_NAME_GET_ERROR" or "PIODA_NOERROR"(Please refer to "Section 2.1 Error Code").

2.3.6 PIODA_Digital_Output

- **Description :**

  This subroutine for PISO-A64/C64 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 value of DO port . Please refer to "Table 2.3" DO port ID.

Table 2.3

PISO-A64 Port ID	PISO-C64 Port ID	DO Port
PISOA64_DOA	PISOC64_DOA	DO 0 ~ DO 7
PISOA64_DOB	PISOC64_DOB	DO 8 ~ DO 15
PISOA64_DOC	PISOC64_DOC	DO 16 ~ DO 23
PISOA64_DOD	PISOC64_DOD	DO 24 ~ DO 31
PISOA64_DOE	PISOC64_DOE	DO 32 ~ DO 39
PISOA64_DOF	PISOC64_DOF	DO 40 ~ DO 47
PISOA64_DOG	PISOC64_DOG	DO 48 ~ DO 55
PISOA64_DOH	PISOC64_DOH	DO 56 ~ DO 63

data : 8 bits data.

- **Return:**

  If returned value = PIODA_NOERROR, it means that sending data to I/O port successfully. Otherwise, please refer to "Section 2.1 Error Code".

2.3.7 PIODA_Digital_Input

- **Description :**

  This subroutine for PISO-P64 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 value of DI port. Please refer to “Table 2.4” DI port ID.

Table 2.4

PISO-P64 Port ID	DI Port
PISOP64_DIA	DI 0 ~ DI 7
PISOP64_DIB	DI 8 ~ DI 15
PISOP64_DIC	DI 16 ~ DI 23
PISOP64_DID	DI 24 ~ DI 31
PISOP64_DIE	DI 32 ~ DI 39
PISOP64_DIF	DI 40 ~ DI 47
PISOP64_DIG	DI 48 ~ DI 55
PISOP64_DIH	DI 56 ~ DI 63

di_data : A variable address used to storage the 8 bits input data.

- **Return:**

  If returned value = PIODA_NOERROR, it means that reading data from DI port successfully. Otherwise, please refer to "Section 2.1 Error Code".

3. PISO-A64/C64/P64 Demo Programs

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 0.20.4 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-0.20.4	Include	piodio.h	PIO/PISO library header
	lib	libpio.a	PIO/PISO static libary
	examples/pisop64	port.c	Digital input demo
		port_a.c	DI demo with libary
	examples/pisoa64 examples/pisoc64	port.c	Digital output demo
		port_a.c	DO demo with libary

3.1 pisoa64/pisoc64 Demo code “port.c”

This demo program is used to output digital from DO 8 ~ 15.

3.2 pisoa64/pisoc64 Demo code “port_a.c”

This demo program coded by using the static library “libpio.a”. It is used to output digital from DO 8 ~ 15.

3.3 pisop64 Demo code “port.c”

This demo program is used to output digital from DI 8 ~ 15.

3.4 pisop64 Demo code “port_a.c”

This demo program coded by using the static library “libpio.a”. It is used to output digital from DI 0 ~ 7.