PISO-CAN200/400

Linux SocketCAN CAN Bus 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

Copyright 2010 by ICP DAS. All rights are reserved.

Trademark

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

Tables of Content

1. Linux Software Installation

The PISO-CAN200/400 SocketCAN driver can be used in linux kernel 2.6.25 or later kernel 2.6.X version. For Linux O.S, the recommended installation and uninstallation steps are given in Sec 1.1 ~ 1.2

1.1 Linux SocketCAN Driver Installing Procedure

Step 1: Download the linux driver "ixcan-0.0.0.tar.gz" (or the later

ixcan package version) from ICP DAS webpage

http://www.icpdas.com/download/pci/piso-can/index.htm to the linux

host.

Step 2: You must use the '**root**' identity to compile and install linux

SocketCAN driver.

Step 3: Decompress the tarball "ixcan.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: Before user install PISO-CAN200/400 driver module (ixcan.ko),

user should check the linux kernel had supported the SocketCAN

driver modules (please refer to Figure 1-1, 1-2, 1-3).

Figure 1-1

Figure 1-2

Figure 1-3

Step 7: You can type '**./ixcan.inst**' to install the PISO-CAN200/400 SocketCAN driver module and build the network device interface "canX". Please refer to the Figure 1-4(the figure show the PISO-CAN400 "canX" interface).

Figure 1-4

Step 8: You can type 'dmesg' to check the number of CAN boards and channel. Please refer to the Figure 1-5 (the figure show the information of PISO-CAN400 boards).

Figure 1-5

1.2   Startup and Stop CAN Interface

Once the driver installed, the CAN interface has to be started and stoped like a standard net interface. Please follow the below steps to startup CAN interface:

Step 1: Use iproute2's (version 2.6.31 or later version) command 'ip' to configure CAN baud rate and startup CAN interface. Please refer to below command and Figure 1-6(can2 baud rate is 125k).

#ip link set can2 up type can bitrate 125000

#ifconfig

Figure 1-6

Step 2:   Besides using 'ip' to startup can interface, user could use the 'ip'

command to check can interface status. Please refer to below command and Figure 1-7.

 # ip -details link show can2

Figure 1-7

Step 3:   If user want to stop can interface, user could use command 'ip' to stop can interface. Please refer to below command.

 # ip link set can2 down

1.3   Linux Driver Uninstalling Procedure

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

Step 2:   Type '**./ixcan.remove**' to remove the SocketCAN driver module.

2. SocketCAN CAN Bus Library Function Description

The static library is the collection of function calls of the PISO-CAN200/400 cards for linux kernel 2.6.25(or later kernel version) system. The application structure is presented as following figure. The user application program developed by C(C++) language can call library "libsktcan.a" in user mode. And then static library will call the SocketCAN modules to access the hardware system.

```
                        ┌─────────────────────────┐
                        │    User's Application     │
                        └─────────────────────────┘
                                    │
                                    │ Function Call into Library
                                    ▼
┌──────────────────────────────────────────────────────────┐
│ Development                                                 │
│ Toolkit          ┌─────────────────────────┐              │
│                  │ Static library "libsktcan.a" │          │
│                  └─────────────────────────┘              │
│                              │                             │
│                              │ Services Call into Kernel-Mode│
│                              ▼                             │
│                  ┌─────────────────────────┐              │
│                  │  SocketCAN Driver Modules │              │
│                  └─────────────────────────┘              │
│                              │                             │
│                              │ Device Control              │
└──────────────────────────────┼────────────────────────────┘
                               ▼
                    ┌─────────────────────────┐
                    │     Hardware Devices      │
                    └─────────────────────────┘
```
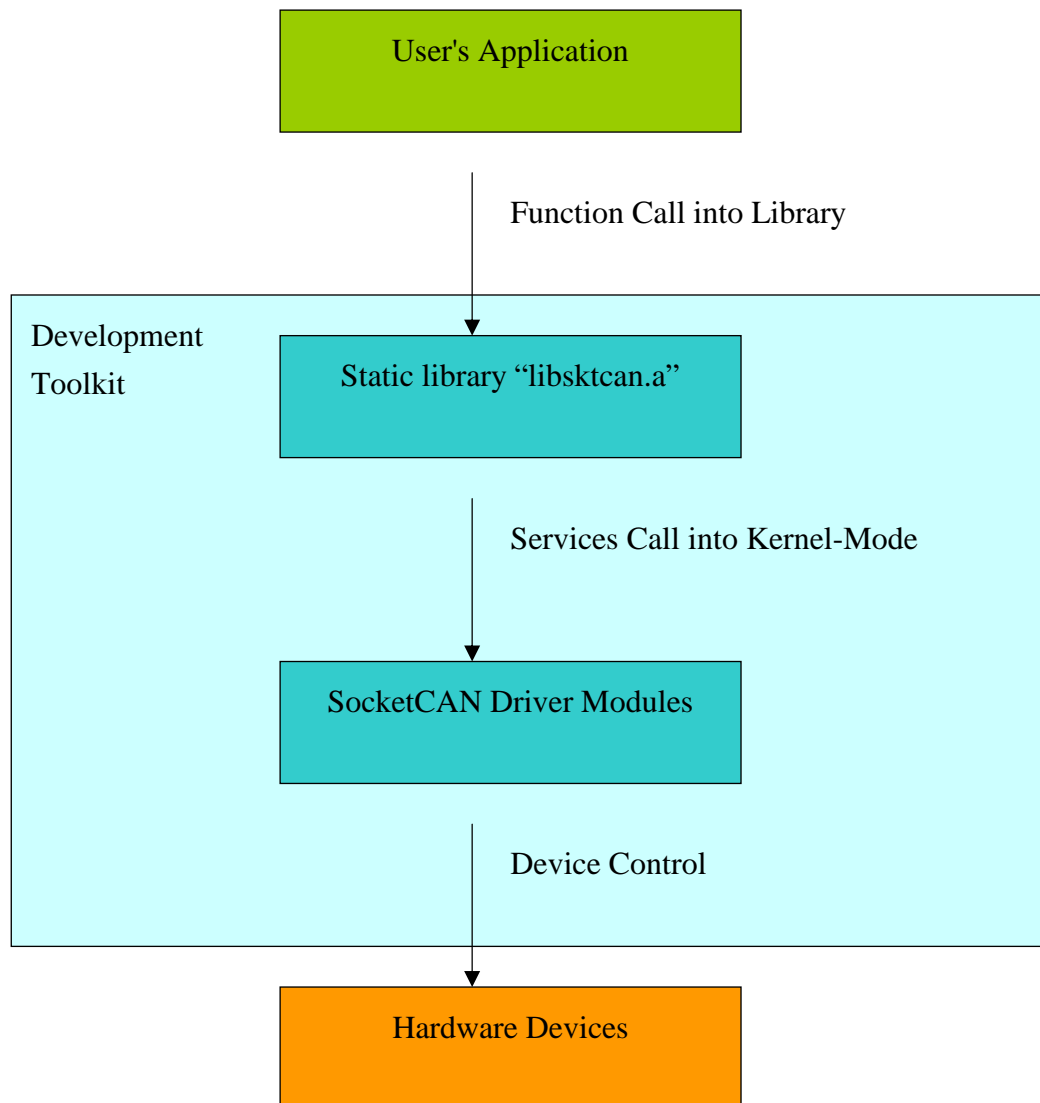
Figure 2.1

2.1 Table of Error Code and Error ID

Error Code	Error ID	Error String
0	SOCKETCAN_NOERROR	OK (No error !)
1	SOCKETCAN_OPEN_ERROR	Open SocketCAN failure
2	SOCKETCAN_BIND_ERROR	Bind SocketCAN failure
3	SOCKETCAN_CLOSE_ERROR	Close SocketCAN failure
4	SOCKETCAN_SEND_FRAME_ERROR	Send CAN Frame failure
5	SOCKETCAN_RECEIVE_FRAME_ERROR	Get CAN Frame failure

Table 2.1

2.2 Function Descriptions

Function Definition
char * SocketCAN_GetDriverVersion(void);
char * SocketCAN_GetLibraryVersion(void);
WORD SocketCAN_Open(char *canport, int *skt);
WORD SocketCAN_Close(int skt);
WORD SocketCAN_SendMsg(int skt, struct can_frame *frame);
WORD SocketCAN_ReceiveMsg(int skt, struct can_frame *frame);

Table 2.2

2.3 SocketCAN CAN Bus Library FUNCTIONS

2.3.1　SocketCAN_GetDriverVersion

- **Description:**

  To get the ixcan driver version.

- **Syntax:**

  char * SocketCAN_GetDriverVersion(Void)

- **Parameter:**

  None

- **Return:**

  The linux ixcan driver version.

2.3.2　SocketCAN_GetLibraryVersion

- **Description:**

  To get the SocketCAN CAN bus library version.

- **Syntax:**

  WORD SocketCAN_GetLibraryVersion(void)

- **Parameter:**

  None

- **Return:**

  The SocketCAN CAN bus library version.

2.3.3　SocketCAN_Open

- **Description:**

  To open CAN socket for PISO-CAN200/400 Devices.

- **Syntax:**

  WORD SocketCAN_Open(char *canport, int *skt)

- **Parameter:**

  canport : The name of CAN network interface.

  skt : To access a file descriptor for the new socket.

- **Return:**

  "SOCKETCAN_NOERROR"

  "SOCKETCAN_OPEN_ERROR"

  "SOCKETCAN_BIND_ERROR"

  Please refer to "Section 2.1 Error Code"

2.3.4 SocketCAN_Close

- **Description :**

  To close CAN Socket for PISO-CAN200/400 Devices.

- **Syntax :**

  WORD SocketCAN_Close(int skt)

- **Parameter :**

  skt : The file descriptor for the CAN socket.

- **Return:**

  "SOCKETCAN_NOERROR"

  "SOCKETCAN_CLOSE_ERROR"

  Please refer to "Section 2.1 Error Code"

2.3.5 SocketCAN_SendMsg

- **Description :**

  To send the CAN frame.

- **Syntax :**

  WORD SocketCAN_SendMsg(int skt, struct can_frame *frame)

- **Parameter :**

  skt : The file descriptor for the CAN socket.

  frame : The basic CAN frame structure.

- **Return:**

  "SOCKETCAN_NOERROR"

  "SOCKETCAN_SEND_FRAME_ERROR"

  Please refer to "Section 2.1 Error Code"

2.3.6 SocketCAN_ReceiveMsg

- **Description :**

  To receive the CAN frame.

- **Syntax :**

  WORD SocketCAN_ReceiveMsg(int skt, struct can_frame *frame)

- **Parameter :**

  skt : The file descriptor for the CAN socket.

  frame : The basic CAN frame structure.

- **Return:**

  "SOCKETCAN_NOERROR"

  "SOCKETCAN_RECEIVE_FRAME_ERROR"

Please refer to "Section 2.1 Error Code"

3. SocketCAN CAN Bus Demo For Linux

All of demo programs will not work normally if PISO-CAN200/400 SocketCAN driver would not be installed correctly. During the installation process, the install-scripts "ixcan.inst" will setup the correct SocketCAN driver. After driver (version 0.0.0 or the later driver version) compiled and installation, the related CAN bus library, demo and header files for different development environments are presented as follows.

Table 3.1

Driver Name	Directory Path	File Name	Description
ixcan-0.0.0	include	sja1000.h	SocketCAN driver header
		pisocan.h	SocketCAN CAN bus library header
	lib	libsktcan.a	The library of CAN bus.
	examples/ pisocan200_400	send_canmsg_a.c	CAN bus library Demo for sending CAN message.
		receive_canmsg_a.c	CAN bus library Demo for receiving CAN message
		receive_send_canmsg_a.c	CAN bus library Demo for sending and receiving CAN message at the same time.
		send_canmsg.c	Demo for sending CAN message.
		receive_canmsg.c	Demo for receiving CAN message

		receive_send_canmsg.c	Demo for sending and receiving CAN message

3.1   Demo code "send_canmsg.c"

This demo program is used to send CAN frame from the can interface that user assigned. Please refer to Figure 3-1.

Figure 3-1

3.2   Demo code "receive_canmsg.c"

This demo program is used to receive CAN frame from the can interface that user assigned. Please refer to Figure 3-2.

Figure 3-2

3.3 Demo code "receive_send_canmsg.c"

This demo program is used to receive and send CAN frame from the can interface that user assigned. Please refer to Figure 3-3.

```
[root@localhost pisocan200_400]# ./receive_send_canmsg can2
interface = can2, family = PF_CAN, type = SOCK_RAW, proto = CAN_RAW
CAN MsgID : 111 -- Msg Length : 8 -- Data : 01 02 03 04 05 06 07 08
```

Figure 3-3

3.4 Demo code "send_canmsg_a.c"

This demo program teach user how to use SocketCAN CAN bus library function to send CAN frame from the can interface that user assigned. Please refer to Figure 3-4.

```
[root@localhost pisocan200_400]# ./send_canmsg_a can2

Press 'Enter' to send data from interface can2

Send CAN Message ID : 0x123 Length : 8
Data[0] : 01
Data[1] : 02
Data[2] : 03
Data[3] : 04
Data[4] : 04
Data[5] : 05
Data[6] : 06
Data[7] : 07

Press 'Esc' to quit or Press 'Enter' to run again
^[
[root@localhost pisocan200_400]#
```

Figure 3-4

3.5 Demo code "receive_canmsg_a.c"

This demo program teach user how to use SocketCAN CAN bus library

function to receive CAN frame from the can interface that user assigned. Please refer to Figure 3-5.

```
[root@localhost pisocan200_400]# ./receive_canmsg_a can3

Receive CAN message from interface can3
Send CAN Message ID : 0x123 Length : 8
Data[0] : 01
Data[1] : 02
Data[2] : 03
Data[3] : 04
Data[4] : 04
Data[5] : 05
Data[6] : 06
Data[7] : 07

^C
```

Figure 3-5

3.6 Demo code "receive_send_canmsg_a.c"

This demo program teach user how to use SocketCAN CAN bus library function to receive and send can frame from the can interface that user assigned. Please refer to Figure 3-6.

```
[root@localhost pisocan200_400]# ./receive_send_canmsg_a can3
CAN MsgID : 123 -- Msg Length : 8 -- Data : 01 02 03 04 04 05 06 07
CAN MsgID : 123 -- Msg Length : 8 -- Data : 01 02 03 04 04 05 06 07
CAN MsgID : 123 -- Msg Length : 8 -- Data : 01 02 03 04 04 05 06 07
^C
```

Figure 3-6