PISO-CAN FD Series

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.

Trademark

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

Tables of Content

1. Linux Software Installation

The PISO-CAN FD SocketCAN driver can be used in x86_64 Linux OS(kernel 5.10.x or later version). For Linux O.S, the recommended installation and uninstall steps are given in Sec 1.1 ~ 1.2

1.1 Linux SocketCAN Driver Installing Procedure

Step 1: Download the Linux driver “ixcanfd-**X.X.X**.tar.gz” from the below ICP DAS webpage to the Linux host.

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

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

Step 3: Decompress the tarball “ixcanfd.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-CAN SocketCAN driver module user should check the Linux kernel had supported the SocketCAN driver modules (please refer to Figure 1-1, 1-2, 1-3).

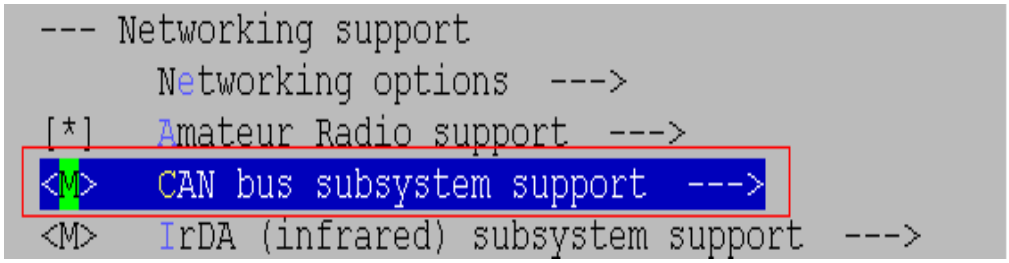

Figure 1-1

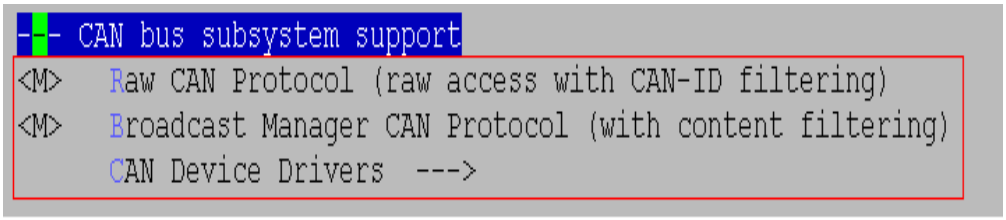

Figure 1-2

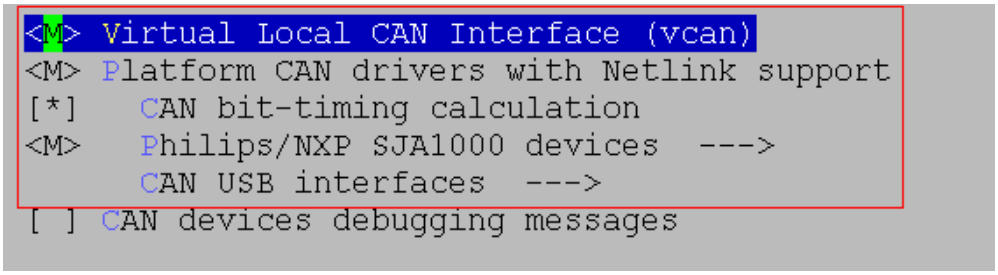

Figure 1-3

Step 7: You can type '**./ixcanfd.inst**' to install the PISO-CAN200U-FD SocketCAN driver module and build the network device interface "canX".

```
root@icpdas:/tmp/ixcanfd# ./ixcanfd.inst
IxCANFD Installation v 0.0.0
Check Kernel version... 5.x
Load module can
Load module can-dev
Load module can-raw
Load module ixcanfd_mcp251xfd
Load module ixcanfd

IxCAN Device Interface

(can0: no entry)
(can1: no entry)
```

Step 8: You can type ‘dmesg' to check the number of CAN boards and channel. The command “dmesg” show the CAN card “Board” number、”Port” number and interface name).

```
root@icpdas:/tmp/ixcanfd# dmesg
…
CAN device driver interface
can: raw protocol
icpdas-ixcanfd 0000:04:00.0 (unnamed net_device) (uninitialized): Detected MCP2518FD, but
firmware specifies a MCP2518FD. Fixing up.
icpdas-ixcanfd 0000:04:00.0: Board #1 : Channel #1 Port #1 (can0) at 0x00000000f6fad11a, irq 17
icpdas-ixcanfd 0000:04:00.0 (unnamed net_device) (uninitialized): Detected MCP2518FD, but
firmware specifies a MCP2518FD. Fixing up.
icpdas-ixcanfd 0000:04:00.0: Board #1 : Channel #2 Port #2 (can1) at 0x00000000f6fad11a, irq 17
```

1.2 Startup and Stop SocketCAN Interface

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

Step 1: Using iproute2’s (version 2.6.31 or later version) command ‘ip’ to set “can1” and “can2” baud rate 1000k and command “ifconfig” to show the SocketCAN interface.

```
root@icpdas:/tmp/ixcanfd# ip link set can0 up type can bitrate 1000000
root@icpdas:/tmp/ixcanfd# ip link set can0 up type can bitrate 1000000
root@icpdas:/tmp/ixcanfd# ifconfig
```

Step 2: Besides using ‘ip’ to startup can interface, user could use the ‘ip’ command to check can interface status.

```
root@icpdas:/tmp/ixcanfd# ip -details link show can0
19: can0: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
DEFAULT group default qlen 10
    link/can  promiscuity 0
    can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 1000000 sample-point 0.750
          tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
          mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
          mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
          clock 40000000numtxqueues 1 numrxqueues 1 gso_max_size 65536
          gso_max_segs 65535

root@icpdas:/tmp/ixcanfd# ip -details link show can1
20: can1: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UP mode
          DEFAULT group default qlen 10
    link/can  promiscuity 0
    can state ERROR-ACTIVE (berr-counter tx 0 rx 0) restart-ms 0
          bitrate 1000000 sample-point 0.750
          tq 25 prop-seg 14 phase-seg1 15 phase-seg2 10 sjw 1
          mcp251xfd: tseg1 2..256 tseg2 1..128 sjw 1..128 brp 1..256 brp-inc 1
          mcp251xfd: dtseg1 1..32 dtseg2 1..16 dsjw 1..16 dbrp 1..256 dbrp-inc 1
          clock 40000000numtxqueues 1 numrxqueues 1 gso_max_size 65536
          gso_max_segs 65535
```

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

```
root@icpdas:/tmp/ixcanfd# ip link set can0 down
root@icpdas:/tmp/ixcanfd# ip link set can1 down
```

1.3 Linux Driver Uninstalling Procedure

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

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

```
root@icpdas:/tmp/ixcanfd#  ./ixcanfd.remove
IxCANFD Removement v 0.0.0
Check Kernel version... 5.10
Unload module ixcan
Unload module ixcanfd_mcp251xfd
Unload module can_raw
Unload module can-dev
Unload module can
```

2. SocketCAN CAN Bus Library Function Description

The static library is the collection of function calls of the PISO-CAN FD cards for Linux kernel 5.10.x or later kernel version. 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.

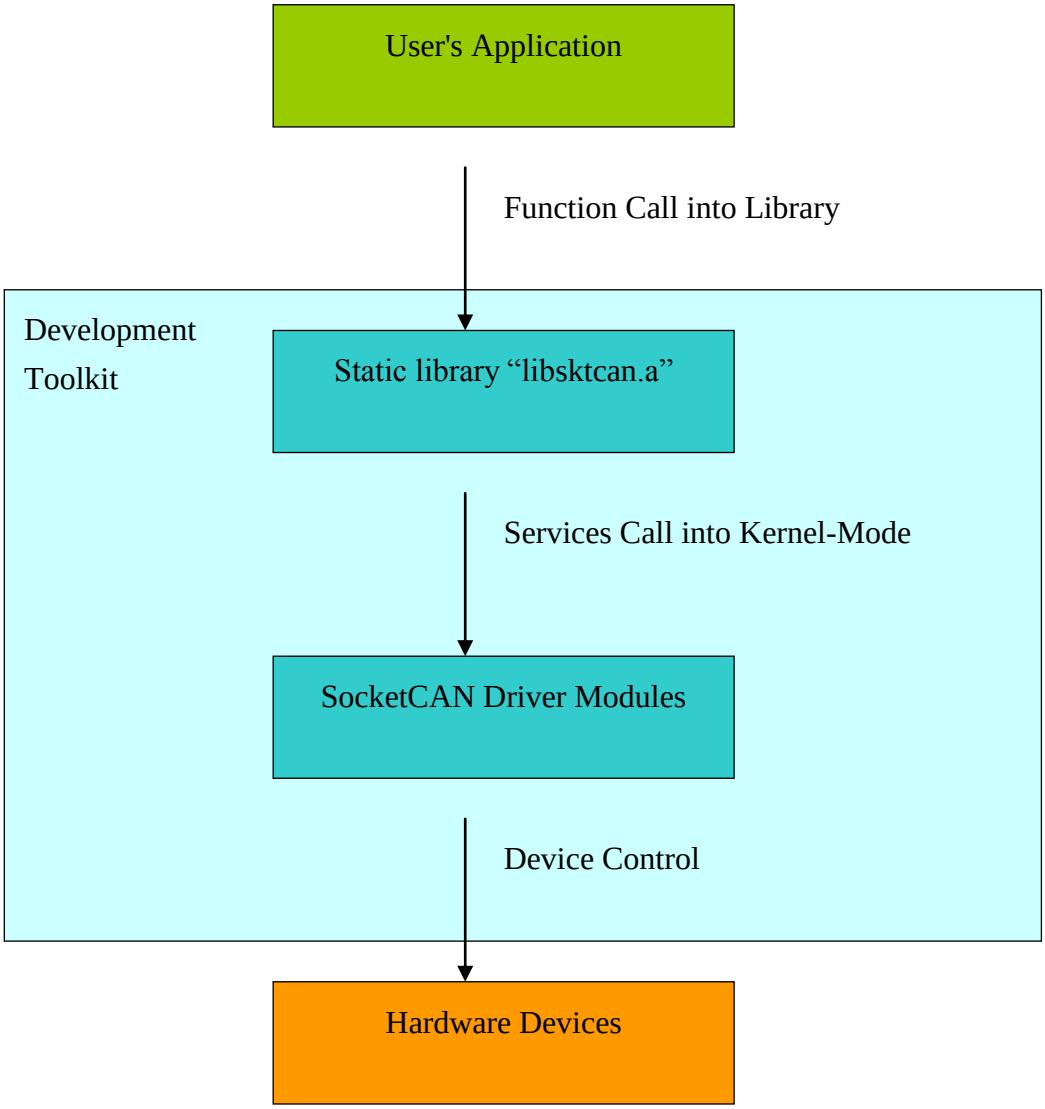

Figure 2.1

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_SendMsgWithResend(int skt, struct can_frame *frame, DWORD resend_count);
WORD SocketCAN_ReceiveMsg(int skt, struct can_frame *frame);
WORD SocketCAN_ReceiveMsgNoWait(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 ixcanfd driver version.
- **Syntax:**
  char * SocketCAN_GetDriverVersion(Void)
- **Parameter:**
  None
- **Return:**
  The Linux ixcanfd 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-CAN 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-CAN 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_SendMsgWithResend

- **Description :**
  To send the CAN frame. Besides if PISO-CAN send CAN frame error, the function would resend CAN frame.
- **Syntax :**
  WORD SocketCAN_SendMsg(int skt, struct can_frame *frame, DWORD resend_count)
- **Parameter :**
  skt : The file descriptor for the CAN socket.
  frame : The basic CAN frame structure.
  resend_count : The times of resending CAN frame when PISO-

CAN send CAN frame error. The max resend times is 60000.

- **Return:**
  “SOCKETCAN_NOERROR”
  “SOCKETCAN_SEND_FRAME_ERROR”
  Please refer to "Section 2.1 Error Code"

2.3.7 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"

2.3.8 SocketCAN_ReceiveMsgNoWait

- **Description :**
  If no CAN frame, the function would return failure without waiting.
- **Syntax :**
  WORD SocketCAN_ReceiveMsgNoWait(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-CAN SocketCAN driver would not be installed correctly. During the installation process, the install-scripts “ixcanfd.inst” will setup the correct SocketCAN driver. After driver (version 0.1.0 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
ixcanfd-0.1.0	include	mcp251xfd.h	SocketCAN driver header
		pisocan.h	SocketCAN CAN bus library header
	lib	libsktcan.a	The library of CAN bus.
	examples/ pisocanfd	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

3.1 Demo code “send_canmsg.c”

This demo program is used to send CAN frame from the can interface “can0” that user assigned.

```
root@icpdas:/tmp/ixcanfd#  ./send_canmsg can0
interface = can0, family = PF_CAN, type = SOCK_RAW, proto = CAN_RAW

Press 'Enter' to send data from interface can0

Send CAN Message ID : 0x123 Length : 8 Data : 01 02 03 04 05 06 07 08

Press 'Esc' to quit or Press 'Enter' to run again
```

3.2 Demo code “receive_canmsg.c”

This demo program is used to receive CAN frame from the can interface “can1” that user assigned.

```
root@icpdas:/tmp/ixcanfd#  ./receive_canmsg can1
interface = can1, family = PF_CAN, type = SOCK_RAW, proto = CAN_RAW

Receive CAN message from interface can1
Receive CAN Message ID : 0x123 Length : 8 Data : 01 02 03 04 05 06 07 08
```

3.3 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 “can0” that user assigned.

```
root@icpdas:/tmp/ixcanfd#  ./send_canmsg_a can0
interface = can0, family = PF_CAN, type = SOCK_RAW, proto = CAN_RAW

Press 'Enter' to send data from interface can0

Send CAN Message ID : 0x123 Length : 8 Data : 01 02 03 04 05 06 07 08

Press 'Esc' to quit or Press 'Enter' to run again
```

Figure 3-4

3.4 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 “can1” that user assigned.

```
root@icpdas:/tmp/ixcanfd#  ./receive_canmsg_a can1
interface = can1, family = PF_CAN, type = SOCK_RAW, proto = CAN_RAW

Receive CAN message from interface can1
Receive CAN Message ID : 0x123 Length : 8 Data : 01 02 03 04 05 06 07 08
```