PISO-CAN

Linux SocketCAN DeviceNet 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-CAN 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 uninstall steps are given in Sec 1.1 ~ 1.2

1.1 Linux SocketCAN Driver Installing Procedure

Step 1: Download the linux driver “ixcan-0.**X**.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-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).

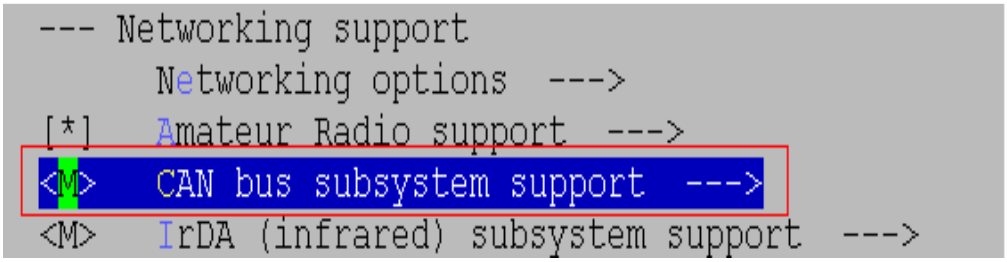

Figure 1-1

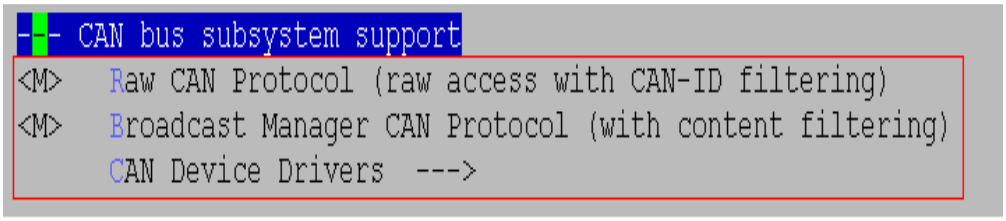

Figure 1-2

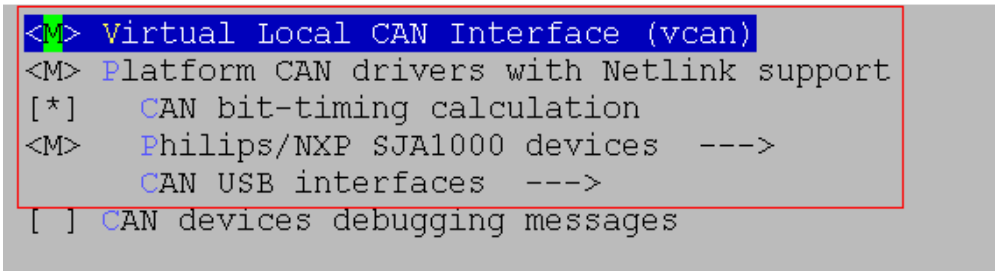

Figure 1-3

Step 7: You can type '**./ixcan.inst**' to install the PISO-CAN SocketCAN driver module and build the network device interface "canX". Please refer to the Figure 1-4(the figure show the PISO-CAN100/200 "canX" interface).

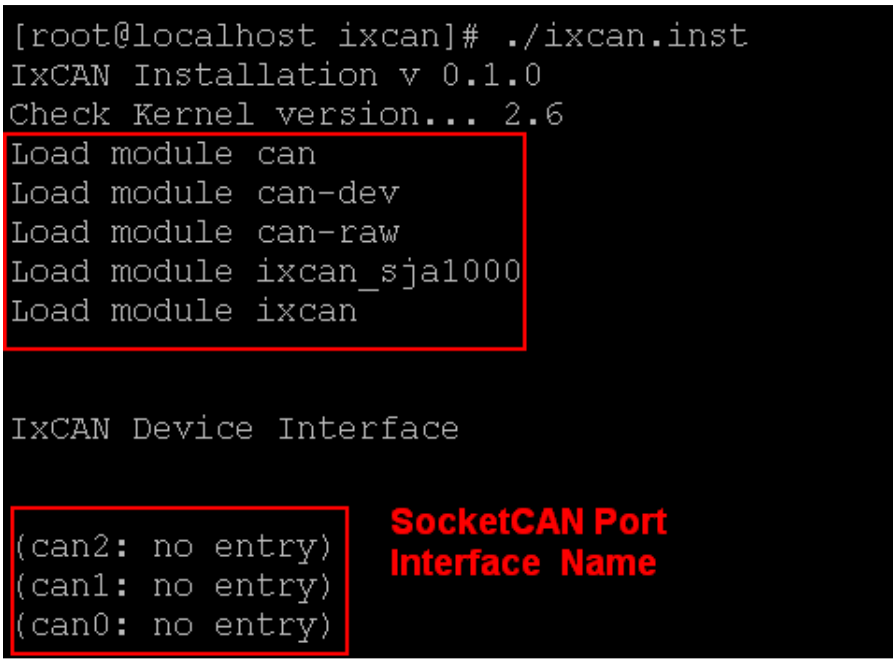

Figure 1-4

Step 8: You can type ‘dmesg' to check the number of CAN boards and channel. Please refer to the Figure 1-5 (“dmesg” show the CAN card’s “Board” number、”Port” number and interface name).

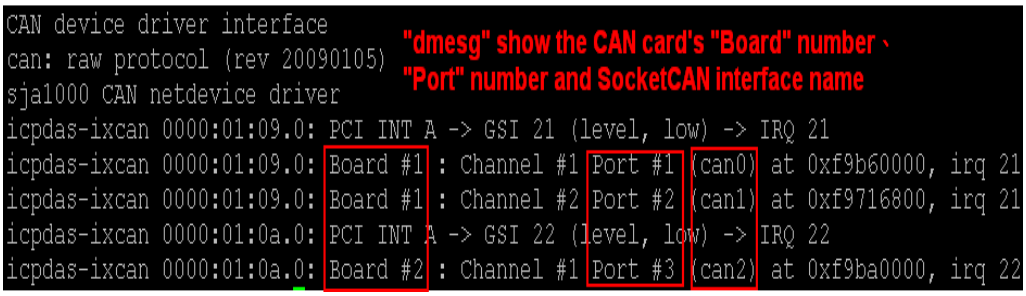

Figure 1-5

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: 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 1000k).

#ip link set can2 up type can bitrate 1000000

#ifconfig

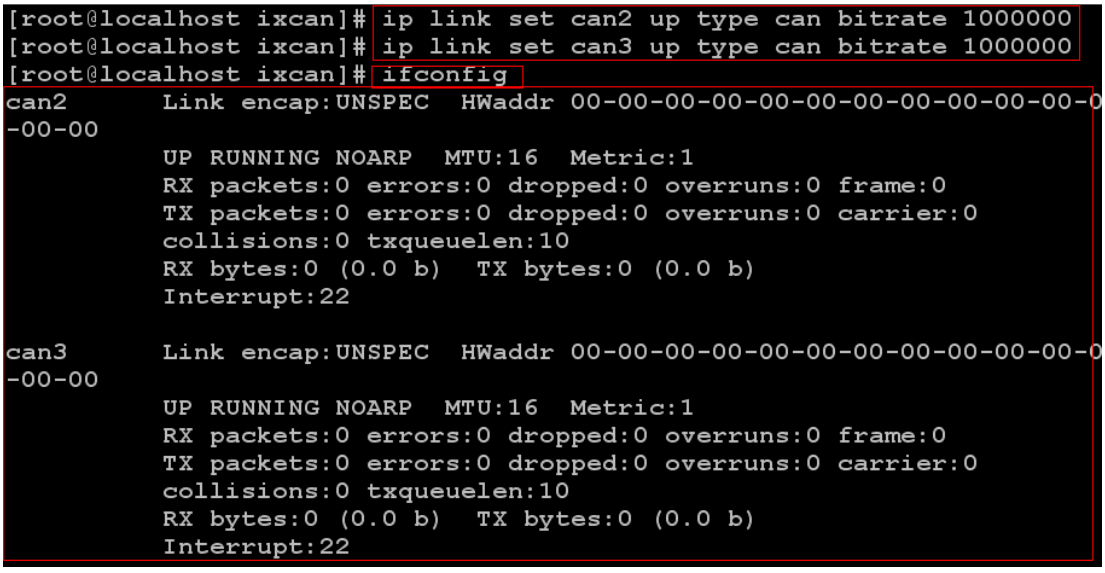

Figure 1-6

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

```
[root@localhost ixcan]# ip -details link show can2
37: can2: <NOARP,UP,LOWER_UP,ECHO> mtu 16 qdisc pfifo_fast state UNKNOWN qlen 10
    link/can
    can state ERROR-ACTIVE restart-ms 0
    bitrate 1000000 sample-point 0.750
    tq 125 prop-seg 2 phase-seg1 3 phase-seg2 2 sjw 1
    sja1000: tseg1 1..16 tseg2 1..8 sjw 1..4 brp 1..64 brp-inc 1
    clock 8000000
```

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 DeviceNet Library Function Description

The SocketCAN DeviceNet library is the collection of function calls of the PISO-CAN cards for linux kernel 2.6.25(or later kernel version) system. The application structure is presented as following figure 2-1. The user application program developed by C(C++) language can call library “libsktdnm.a” in user mode. And then static library will call the SocketCAN modules to access the hardware system.

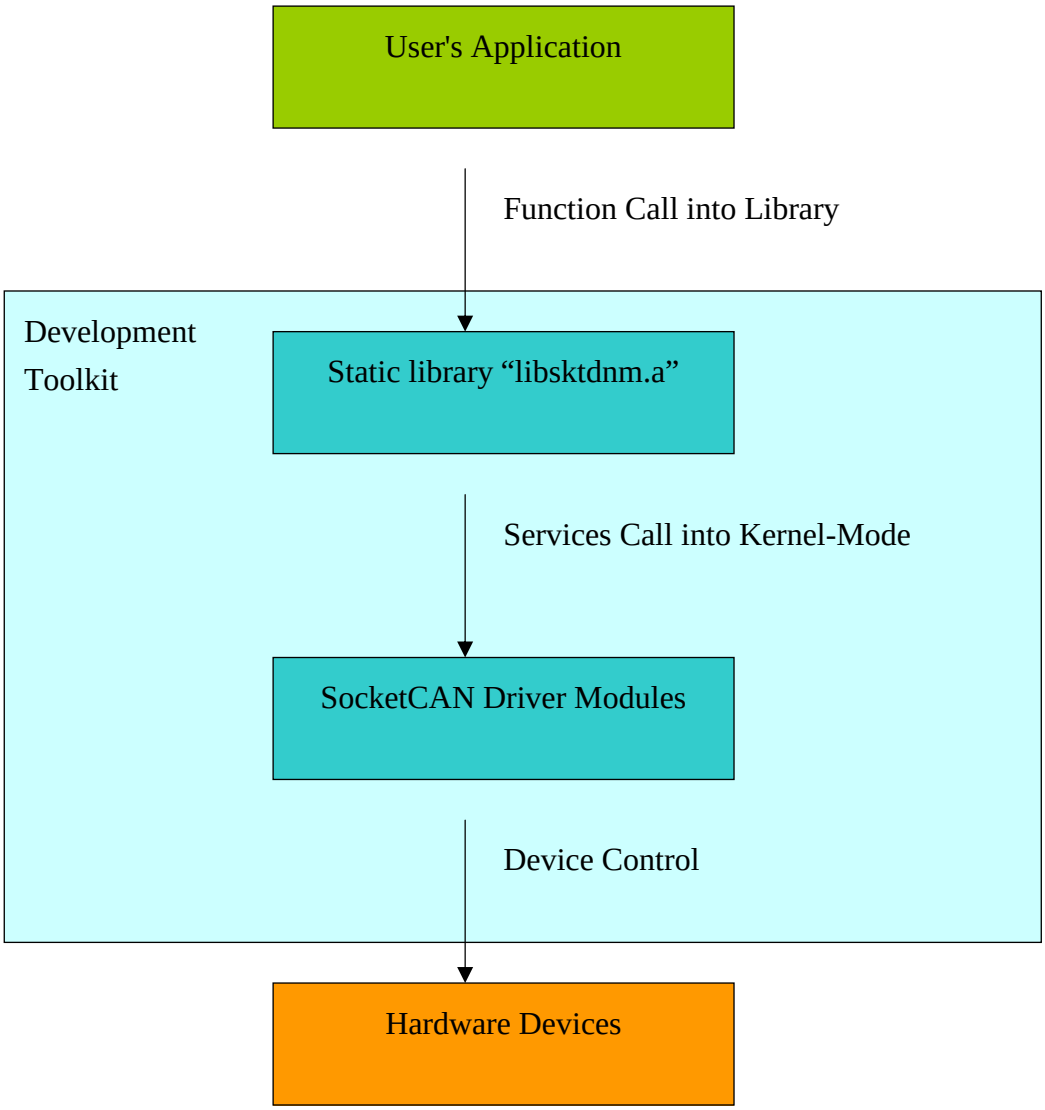

Figure 2-1

2.1 Table of Error Code and Error ID

Error Code	Error ID	Error String
0	DNM_NOERROR	OK (No error !)
1	DNM_OPEN_ERROR	Open CAN DeviceNet master device failure
2	DNM_CLOSE_ERROR	Close CAN DeviceNet master device failure
3	DNM_PORT_ALREADY_ONLINE	CAN port is already online
4	DNM_MACID_ERROR	The MAC id exceed the range between 0 ~ 63
5	DNM_PORT_INIT_ERROR	CAN port initial error
6	DNM_PORT_CONFIG_ERROR	Configure CAN port error
7	DNM_ENABLERXIRQ_ERROR	Enable CAN port interrupt error
8	DNM_SENDMSG_ERROR	Send CAN message error
9	DNM_ONLINE_ERROR	Master MAC ID is duplicate in the DeviceNet network
10	DNM_DUPLICATE_MACID_TIMEOUT	It is timeout when waiting for the duplicate MAC ID message
11	DNM_ADDTHREAD_ERROR	Create the thread of connection object error
12	DNM_RESUMETHREAD_ERROR	Resume thread error
13	DNM_MASTER_STATE_ERROR	The master is in illegal state
14	DNM_OFFLINE_ERROR	The offline function fail
15	DNM_CHANGE_MACID_ERROR	Change master MAC id error
16	DNM_CHANGE_BAUDRATE_ERROR	Change master baud rate error
17	DNM_DEVICE_IN_SCANLIST	The device is in master scan list
18	DNM_DEVICE_NOT_IN_SCANLIST	The device isn’t in master scan list
19	DNM_EXPLICITMSG_NOT_CONFIG	The device doesn’t configure explicit message connection

Error Code	Error ID	Error String
20	DNM_CONFIG_EXPLICITMSG_ERROR	Configure slave explicit message connection error
21	DNM_POLL_NOT_CONFIG	The device doesn't configure poll I/O connection
22	DNM_CONFIG_POLL_ERROR	Configure slave poll I/O connection error
23	DNM_WRITE_POLL_OUTPUT_DATALENGTH_ERROR	The data length is error when sending poll I/O data
24	DNM_BITSTROBE_NOT_CONFIG	The device doesn't configure bit-strobe I/O connection
25	DNM_CONFIG_BITSTROBE_ERROR	Configure slave bit-strobe I/O connection error
26	DNM_COS_CYCLIC_NOT_CONFIG	The device doesn't configure COS/Cyclic I/O connection
27	DNM_CONFIG_COS_CYCLIC_ERROR	Configure slave COS/Cyclic I/O connection error
28	DNM_STROBE_REQUEST_THREAD_NOT_EXIST	The master has errors when creating Bit-Strobe requester thread
29	DNM_SUSPEND_STROBE_REQUEST_THREAD_ERROR	The master has errors when suspending Bit Strobe requester thread
30	DNM_THREAD_CANCEL_ERROR	The master has errors when cancel connection-object thread
31	DNM_SET_ATTRIBUTE_DATA_IS_EMPTY	The data is empty when setting attribute
32	DNM_DEVICE_IS_UNCONNECTED	The device doesn't establish the connection. Maybe the configuration has some error.
33	DNM_DEVICE_IS_CONNECTING	The device is connecting now.
34	DNM_SET_ATTRIBUTE_RESPONSE_TIMEOUT	The device doesn't response within specific time.
35	DNM_SET_ATTRIBUTE_FRAGMENT_ACK_TIMEOUT	It is timeout when waiting for ACK message

Error Code	Error ID	Error String
36	DNM_SET_ATTRIBUTE_RESPONSE_ERROR	The device response error when setting attribute
37	DNM_SET_ATTRIBUTE_ACK_ERROR	Setting attribute ACK message error
38	DNM_WAIT_BUFFER_RESPONSE_TIMEOUT	It is timeout when waiting for the destination device response
39	DNM_GET_ATTRIBUTE_RESPONSE_TIMEOUT	The device doesn't response within specific time.
40	DNM_GET_ATTRIBUTE_RESPONSE_ERROR	The device response error when getting attribute
41	DNM_CREATE_EXPLICITMSG_THREAD_ERROR	The master create explicit thread error
42	DNM_DEVICE_ERROR	MACID Error or Power off.
43	DNM_COS_CYCLIC_ALREADY_CONFIG	This message means that the COS/Cyclic connection has been configured and started device, so the poll connection can not be configured.
44	DNM_COS_CYCLIC_CONNECT_DUPLICATE	It is illegal when configuring both COS and Cyclic connections in one device.
45	DNM_SET_POLL_INPUT_LEN_ERROR	The poll input length you set doesn't match with the device length.
46	DNM_SET_POLL_OUTPUT_LEN_ERROR	The poll output length you set doesn't match with the device length.
47	DNM_SET_BITSTROBE_INPUT_LEN_ERROR	The Bit-Strobe input length you set doesn't match with the device length.
48	DNM_WRITE_COS_OUTPUT_LEN_ERROR	The data length is error when sending COS I/O data.
49	DNM_WRITE_CYCLIC_OUTPUT_LEN_ERROR	The data length is error when sending Cyclic I/O data.

Error Code	Error ID	Error String
50	DNM_SET_COS_INPUT_LEN_ERROR	The COS input length doesn't match with the device length.
51	DNM_SET_CYCLIC_INPUT_LEN_ERROR	The Cyclic input length you set doesn't match with the device length.
52	DNM_SET_COS_OUTPUT_LEN_ERROR	The COS output length you set doesn't match with the device length.
53	DNM_SET_CYCLIC_OUTPUT_LEN_ERROR	The Cyclic output length you set doesn't match with the device length.
54	DNM_GET_OWNERSHIP_ERROR	The Master can not get the ownership.
55	DNM_MASTER_NOT_GET_OWNERSHIP	The Master has not got the ownership yet.
56	DNM_BITSTROBE_NO_RESPONSE	The Bit-Strobe connection has no response
57	DNM_BITSTROBE_RESPONSE_DATA_ERROR	The Bit-Strobe connection response error messages.
58	DNM_POLL_NO_RESPONSE	The Poll connection has no response
59	DNM_POLL_RESPONSE_DATA_ERROR	The Poll connection response error message.
60	DNM_POLL_FRAGMENT_ERROR	The device has fragment errors
61	DNM_COS_CYCLIC_NO_RESPONSE	The COS/Cyclic connection has no response
62	DNM_COS_CYCLIC_RESPONSE_DATA_ERROR	The COS/Cyclic connection response error messages
63	DNM_COS_CYCLIC_FRAGMENT_ERROR	The COS/Cyclic connection has fragment errors
64	DNM_NORESPONSE	The slave device no response

Table 2.1

2.2 Function Descriptions

Function Definition
char * DNM_GetDriverVersion(void)
char * DNM_GetLibraryVersion(void)
DWORD DNM_Open(BYTE BoardNo, BYTE Port)
DWORD DNM_Close(BYTE BoardNo, BYTE Port)
DWORD DNM_Online(BYTE BoardNo , BYTE Port , DWORD Baudrate , DWORD Macid)
DWORD DNM_Offline(BYTE BoardNo , BYTE Port)
DWORD DNM_ChangeMACID(BYTE BoardNo , BYTE Port , DWORD Macid)
DWORD DNM_AddDevice(BYTE BoardNo , BYTE Port , DWORD DesMacID)
DWORD DNM_RemoveDevice(BYTE BoardNo, BYTE Port, DWORD DesMacID)
DWORD DNM_ConfigExplicitMsg(BYTE BoardNo , BYTE Port , DWORD DesMacID , WORD watchdog_timeout_action)
DWORD DNM_SetAttribute(BYTE BoardNo , BYTE Port , DWORD DesMacID , WORD ClassID , BYTE InstanceID , BYTE AttributeID , BYTE *SetValue , DWORD Length)
DWORD DNM_GetAttribute(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD ClassID, BYTE InstanceID, BYTE AttributeID, BYTE *GetValue, DWORD *Length)
DWORD DNM_CheckExplicitMsgConnectionStatus(BYTE BoardNo , BYTE Port , DWORD DesMacID)
DWORD DNM_ConfigPoll(BYTE BoardNo , BYTE Port , DWORD DesMacID , WORD produced_connection_size,WORD consumed_connection_size,WORD expected_packet_rate,WORD watchdog_timeout_action)
DWORD DNM_ReadPollInputData(BYTE BoardNo, BYTE Port, DWORD DesMacID ,BYTE *DataBuf ,DWORD *DataLength)
DWORD DNM_WritePollOutputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD DataLength)
DWORD DNM_CheckPollConnectionStatus(BYTE BoardNo , BYTE Port , DWORD DesMacID)
DWORD DNM_ConfigBitStrobe(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD produced_connection_size, WORD expected_packet_rate, WORD watchdog_timeout_action)

Function Definition
DWORD DNM_ReadBitStrobe(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD *DataLength)
DWORD DNM_CheckBitStrobeConnectionStatus(BYTE BoardNo , BYTE Port , DWORD DesMacID)
DWORD DNM_ConfigCOS(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD produced_connection_size, WORD consumed_connection_size, BYTE isnonack, WORD expected_packet_rate, WORD watchdog_timeout_action)
DWORD DNM_ConfigCyclic(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD produced_connection_size, WORD consumed_connection_size, BYTE isnonack, WORD expected_packet_rate, WORD watchdog_timeout_action)
DWORD DNM_ReadCOSInputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD *DataLength)
DWORD DNM_WriteCOSOutputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD DataLength)
DWORD DNM_ReadCyclicInputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD *DataLength)
DWORD DNM_WriteCyclicOutputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD DataLength)
DWORD DNM_CheckCOSConnectionStatus(BYTE BoardNo, BYTE Port, DWORD DesMacID)
DWORD DNM_CheckCyclicConnectionStatus(BYTE BoardNo, BYTE Port, DWORD DesMacID)
DWORD DNM_StartDevice(BYTE BoardNo, BYTE Port, DWORD DesMacID)
DWORD DNM_StopDevice(BYTE BoardNo, BYTE Port, DWORD DesMacID)
DWORD DNM_StartCommunicate(BYTE BoardNo, BYTE Port)
DWORD DNM_StopCommunicate(BYTE BoardNo, BYTE Port)
DWORD DNM_GetOwnership(BYTE BoardNo, BYTE Port)
DWORD DNM_CheckFaultNode(BYTE BoardNo, BYTE Port)
DWORD DNM_GetAllFaultNode(BYTE BoardNo, BYTE Port, WORD *VendorID, DWORD *SerialNo, BYTE *NodeCount)
DWORD DNM_ChangeFaultMACID(BYTE BoardNo, BYTE Port, WORD VendorID, DWORD SerialNo, BYTE NewMacID)
DWORD DNM_GetAllDeviceMACID(BYTE BoardNo, BYTE Port, BYTE *AllMacID, BYTE *DeviceCount)
DWORD DNM_GetDeviceStatus(BYTE BoardNo,BYTE Port,BYTE DesMacID)

Function Definition
DWORD DNM_SetPollFrequency(WORD Frequency)
DWORD DNM_SetBitStrobeFrequency(WORD Frequency)

Table 2.2

2.3 PISO-CAN Series DeviceNet Library Functions

2.3.1 DNM_GetDriverVersion

- **Description:**
  To get the linux IxPCI driver version.
- **Syntax:**
  char * DNM_GetDriverVersion(void)
- **Parameter:**
  None
- **Return:**
  The linux driver of ixpci version.

2.3.2 DNM_GetLibraryVersion

- **Description:**
  To get the linux CAN DeviceNet library version.
- **Syntax:**
  char * DNM_GetLibraryVersion(void)
- **Parameter:**
  None
- **Return:**
  The CAN DeviceNet master library version.

2.3.3 DNM_Open

- **Description:**
  To open the device file of PISO-CAN series DeviceNet master.
- **Syntax:**
  DWORD DNM_Open(BYTE BoardNo, BYTE Port)
- **Parameter:**
  BoardNo: The board number of PISO-CAN board(refer to figure 1.5)
  Port: The port number of PISO-CAN board(refer to figure 1.5)
- **Return:**
  “DNM_NOERROR”
  “DNM_OPEN_ERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.4 DNM_Close

- **Description :**

  To close PISO-CAN series DeviceNet master device file.

- **Syntax :**

  WORD DNM_Close(BYTE BoardNo, BYTE Port)

- **Parameter :**

  BoardNo: The board number of PISO-CAN board(refer to figure 1.5)
  Port: The port number of PISO-CAN board(refer to figure 1.5)

- **Return:**

  “DNM_NOERROR”
  “DNM_CLOSE_ERROR”
  “DNM_OPEN_ERROR”
  (Please refer to "Section 2.1 Error Code").

2.3.5 DNM_Online

- **Description :**

  The function can make the CAN Port online. When calling this function, the CAN port will check the duplicate MAC ID. After the function work well, the CAN port become the DeviceNet master port. Then, the port will also transfer into online state in the DeviceNet network.

- **Syntax :**

  DWORD DNM_Online(BYTE BoardNo , BYTE Port , DWORD Baudrate, DWORD Macid);

- **Parameter :**

  BoardNo: [input] PISO-CAN board number(refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  Baudrate: [input] DeviceNet baud rate (125, 250 and 500)
  Macid: [input] DeviceNet Master MAC ID (0~63)

- **Return:**

  “DNM_OPEN_ERROR”
  “DNM_PORT_ALREADY_ONLINE”
  “DNM_MACID_ERROR”
  “DNM_BAUDRATE_ERROR “
  “DNM_PORT_INIT_ERROR “
  “DNM_PORT_CONFIG_ERROR “
  “DNM_ENABLERXIRQ_ERROR “
  “DNM_SENDMSG_ERROR “

“DNM_ONLINE_ERROR “
“DNM_ADDTHREAD_ERROR “
“DNM_RESUMETHREAD_ERROR “
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.6 DNM_Offline

- **Description :**
  The method makes the CAN port offline. After calling this function, the CAN Port will disable the functions of the DeviceNet master device. It will transfer into offline state in DeviceNet network. In this status, the CAN port will not communicate with any slave devices.
- **Syntax :**
  DWORD DNM_Offline(BYTE BoardNo , BYTE Port)
- **Parameter :**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR”
  “DNM_ADDTHREAD_ERROR “
  “DNM_RESUMETHREAD_ERROR “
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code").

2.3.7 DNM_ChangeMACID

- **Description :**
  When the CAN port is online, this function can change the MAC ID of this master port.
- **Syntax :**
  DWORD DNM_ChangeMACID(BYTE BoardNo , BYTE Port , DWORD Macid)
- **Parameter :**
  BoardNo: [input] PISO-CAN series board number(refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  Macid: [input] DeviceNet Master MAC ID (0~63)
- **Return:**

“DNM_OPEN_ERROR”
“DNM_MASTER_STATE_ERROR”
“DNM_MACID_ERROR “
“DNM_CHANGE_MACID_ERROR “
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.8 DNM_AddDevice

- **Description :**
  This function can add the slave devices into the scan list of the CAN port of DeviceNet master. Before using this function, the port must be online. The master port will use the scan list to store the information of the device. And the master port just can communicate with the slave devices in this scan list.
- **Syntax :**
  DWORD DNM_AddDevice(BYTE BoardNo , BYTE Port , DWORD DesMacID)
- **Parameter :**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] Destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR “
  “DNM_DEVICE_IN_SCANLIST “
  “DNM_MACID_ERROR “
  “DNM_NOERROR “
  (Please refer to "Section 2.1 Error Code").

2.3.9 DNM_RemoveDevice

- **Description :**
  The function would remove the slave device from the scan list of this DeviceNet master port. And the Master will also remove the device information from scan list.
- **Syntax :**
  DWORD DNM_RemoveDevice(BYTE BoardNo, BYTE Port, DWORD

DesMacID)

- **Parameter :**

  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] Destination DeviceNet MAC ID (0~63)

- **Return:**

  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST “
  “DNM_NOERROR “
  (Please refer to "Section 2.1 Error Code").

2.3.10 DNM_ConfigExplicitMsg

- **Description :**

  The function is used to configure the Explicit Message connection. This is the based connection of the DeviceNet network. Before calling the function, the destination slave device must have been added into the master scan list by using the “DNM_AddDevice” function.

- **Syntax :**

  DWORD DNM_ConfigExplicitMsg(BYTE BoardNo , BYTE Port , DWORD DesMacID , WORD watchdog_timeout_action)

- **Parameter :**

  BoardNo: [input] PISO-CAN board number(refer to figure 1.5)
  Port: [input] CAN port number(refer to figure 1.5)
  DestMACID: [input] Destination DeviceNet MAC ID (0~63)
  watchdog_timeout_action: [input] The action when watchdog is timeout.

  According to the DeviceNet specification, the Explicit Message Connection can be set 1 or 3.

  1： Auto Delete. This means when watchdog is timeout, the connection will be deleted automatically. This is the default value.

  3： Deferred Delete. This means when watchdog is timeout, the connection will transfer to the deferred state. It would wait until its child connections have been delete. If the connection has no child, it will be deleted.

- **Return:**

“DNM_OPEN_ERROR”
“DNM_MACID_ERROR”
“DNM_MASTER_STATE_ERROR”
“DNM_DEVICE_NOT_IN_SCANLIST”
“DNM_CONFIG_EXPLICITMSG_ERROR“
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.11 DNM_SetAttribute

- **Description :**
  The method is used to set the attribute of the specific device’s instance. Before calling this function, the Explicit Message connection must be established first.
- **Syntax :**
  DWORD DNM_SetAttribute(BYTE BoardNo , BYTE Port , DWORD DesMacID , WORD ClassID , BYTE InstanceID , BYTE AttributeID , BYTE *SetValue , DWORD Length)
- **Parameter :**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number(refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  ClassID: [input] The Class ID of the destination device.
  InstanceID:[input] The Instance ID of the destination device.
  AttributeID: [input] The Attribute ID of the destination device.
  SetValue: [input] The value will be set to the destination device.
  Length: [input] The length (in byte) of the SetValue.
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_SET_ATTRIBUTE_DATA_IS_EMPTY“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_SENDMSG_ERROR”
  “DNM_SET_ATTRIBUTE_RESPONSE_TIMEOUT”
  “DNM_SET_ATTRIBUTE_FRAGMENT_ACK_TIMEOUT“
  “DNM_SET_ATTRIBUTE_RESPONSE_ERROR“

“DNM_SET_ATTRIBUTE_ACK_ERROR“
“DNM_SET_ATTRIBUTE_FRAGMENT_ACK_TIMEOUT“
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.12 DNM_GetAttribute

- **Description :**
  This function is used to get the attribute value of the specific instance id of device. Before calling this function, the Explicit Message connection must be established first.
- **Syntax :**
  DWORD DNM_GetAttribute(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD ClassID, BYTE InstanceID, BYTE AttributeID, BYTE *GetValue, DWORD *Length)
- **Parameter :**
  BoardNo: [input] PISO-CAN board number(refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  ClassID: [input] The Class ID of the destination device.
  InstanceID: [input] The Instance ID of the destination device.
  AttributeID: [input] The Attribute ID of the destination device.
  GetValue: [output] The variable will return the data form the device.
  Length: [output] The length (in byte) of the GetValue.
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_SENDMSG_ERROR”
  “DNM_GET_ATTRIBUTE_RESPONSE_TIMEOUT”
  “DNM_GET_ATTRIBUTE_RESPONSE_ERROR”
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code").

2.3.13 DNM_CheckExplicitMsgConnectionStatus

- **Description :**

The function is used to check the connection status of Explicit Message.

- **Syntax :**
  DWORD DNM_CheckExplicitMsgConnectionStatus(BYTE BoardNo , BYTE Port , DWORD DesMacID)
- **Parameter :**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] Destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_EXPLICITMSG_NOT_CONFIG“
  “DNM_DEVICE_ERROR”
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code").

2.3.14 DNM_ConfigPoll

- **Description :**
  This method is used to configure the Poll connection of the specific slave device. The master port can get/set the data via the connection, according to the produced/consumed connection path of this slave device. The connection timeout and watchdog timeout action can be set from “expected_packet_rate” and “watchdog_timeout_action” parameters.
- **Syntax :**
  DWORD DNM_ConfigPoll(BYTE BoardNo , BYTE Port , DWORD DesMacID , WORD produced_connection_size,WORD consumed_connection_size,WORD expected_packet_rate,WORD watchdog_timeout_action);
- **Parameter :**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  produced_connection_size: [input] The data length of the input channel of the device.
  consumed_connection_size: [input] The data length of the input

channel of the device.
expected_packet_rate: [input] The default value is 0X9C4 (2500)
watchdog_timeout_action: [input] The default value is 0

- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_ALREADY_CONFIG”
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code").

2.3.15 DNM_CheckPollConnectionStatus

- **Description:**
  The function is used to check the Poll connection status.
- **Syntax:**
  DWORD DNM_CheckPollConnectionStatus(BYTE BoardNo , BYTE Port , DWORD DesMacID);
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_POLL_NOT_CONFIG“
  “DNM_SET_POLL_INPUT_LEN_ERROR“
  “DNM_SET_POLL_OUTPUT_LEN_ERROR“
  “DNM_DEVICE_ERROR“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code").

2.3.16 DNM_ReadPollInputData

- **Description:**

This function is to get the data according with the produced connection path of the specific slave device via the Poll connection.

- **Syntax:**
  DWORD DNM_ReadPollInputData(BYTE BoardNo, BYTE Port, DWORD DesMacID ,BYTE *DataBuf ,DWORD *DataLength)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  DataBuf: [output] The data of the input channel
  DataLength: [output] The length (in Byte) of the data
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST”
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.17 DNM_WritePollOuputData

- **Description:**
  The function will set the data according with the path of consumed connection of the specific device via the Poll connection.
- **Syntax:**
  DWORD DNM_WritePollOutputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD DataLength)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  DataBuf: [input]The data of the output channel
  DataLength: [input]The length(in Byte) of the data
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“

“DNM_DEVICE_NOT_IN_SCANLIST“
“DNM_DEVICE_IS_UNCONNECTED“
“DNM_WRITE_POLL_OUTPUT_DATALENGTH_ERROR”
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code")

2.3.18 DNM_ConfigBitStrobe

- **Description:**
  The function can configure the Bit-Strobe connection of the specific device. The master port will get the data of the slave devices every period of time via this connection. The connection timeout and watchdog timeout action can be set to “expected_packet_rate” and “watchdog_timeout_action” parameters.
- **Syntax:**
  DWORD DNM_ConfigBitStrobe(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD produced_connection_size, WORD expected_packet_rate, WORD watchdog_timeout_action)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  produced_connection_size: [input] The data length of the input channel of the device.
  expected_packet_rate: [input] The default value is 0X9C4 (2500)
  watchdog_timeout_action: [input] The default value is 0
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST “
  “DNM_NOERROR“
  (Please refer to "Section 2.1 Error Code")

2.3.19 DNM_ReadBitStrobe

- **Description:**
  The function is used to get the data according with the produced connection path of the specific slave device via the Bit-Strobe

connection.

- **Syntax:**
  DWORD DNM_ReadBitStrobe(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD *DataLength)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  DataBuf: [output] The data of the input channel
  DataLength: [output] The length (in Byte) of the data
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.20 DNM_CheckBitStrobeConnectionStatus

- **Description:**
  The function can check the status of the Bit-Strobe connection, which is connected to the specific slave device.
- **Syntax:**
  DWORD DNM_CheckBitStrobeConnectionStatus(BYTE BoardNo , BYTE Port , DWORD DesMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63).
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST”
  “DNM_BITSTROBE_NOT_CONFIG”
  “DNM_SET_BITSTROBE_INPUT_LEN_ERROR”

“DNM_DEVICE_ERROR”
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code")

2.3.21 DNM_ConfigCOS

- **Description:**
  This function is used to configure the COS connection. This connection will get the data every period of time, when the connection of Poll does not exist. If the Poll connection exists, the connection will set the data when needed and get the data when the specific status changed. The connection timeout and watchdog timeout action can be set from “expected_packet_rate” and “watchdog_timeout_action” parameters.
- **Syntax:**
  DWORD DNM_ConfigCOS(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD produced_connection_size, WORD consumed_connection_size, BYTE isnonack, WORD expected_packet_rate, WORD watchdog_timeout_action)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  produced_connection_size: [input] The data length of the input channel
  consumed_connection_size: [input] The data length of the output channel
  isnonack:[input] The default value is FALSE
  expected_packet_rate: [input] The default value is 0X9C4 (2500)
  watchdog_timeout_action: [input] The default value is 0;
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_CONNECT_DUPLICATE“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.22 DNM_ConfigCyclic

- **Description:**

  This function is used to configure the Cyclic connection. The connection will get the data according to the produced connection path of the specific slave device channel every period of time, when the connection of Poll does not exist. If the Poll connection exists, the master port will get the data every period of time and could set the data via the connection. The connection timeout and watchdog timeout action can be set from “expected_packet_rate” and “watchdog_timeout_action” parameters.

- **Syntax:**

  DWORD DNM_ConfigCyclic(BYTE BoardNo, BYTE Port, DWORD DesMacID, WORD produced_connection_size, WORD consumed_connection_size, BYTE isnonack, WORD expected_packet_rate, WORD watchdog_timeout_action)

- **Parameter:**

  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  produced_connection_size: [input] The data length of the input channel.
  consumed_connection_size: [input]The data length of the output channel.
  isnonack:[input] The default value is FALSE
  expected_packet_rate: [input] The default value is 0X9C4 (2500)
  watchdog_timeout_action: [input] The default value is 0.

- **Return:**

  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_CONNECT_DUPLICATE“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.23 DNM_ReadCOSInputData

- **Description:**

  This method is used to read the data of the specific slave device via the

COS connection.

- **Syntax:**
  DWORD DNM_ReadCOSInputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD *DataLength)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  DataBuf: [output] The data of the input channel
  DataLength: [output]The length(in Byte) of the data
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_NOT_CONFIG“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.24 DNM_WriteCOSOutputData

- **Description:**
  This function is used to set the data according to the consumed connection path of the specific slave device via the COS connection.
- **Syntax:**
  DWORD DNM_WriteCOSOutputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD DataLength)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  DataBuf: [input] The data of the output channel
  DataLength: [input] The length (in Byte) of the data
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“

“DNM_DEVICE_NOT_IN_SCANLIST“
“DNM_WRITE_COS_OUTPUT_LEN_ERROR“
“DNM_DEVICE_IS_UNCONNECTED“
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code")

2.3.25 DNM_ReadCyclicInputData

- **Description:**
  This function is used to get the data according to the produced connection path of the specific slave device via the Cyclic connection.
- **Syntax:**
  DWORD DNM_ReadCyclicInputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD *DataLength)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
  DataBuf: [output] The data of the input channel
  DataLength: [output] The length(in Byte) of the data.
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_NOT_CONFIG“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.26 DNM_WriteCyclicOutputData

- **Description:**
  This method is used to set the data according to the consumed connection path of the output channel of the specific slave device.
- **Syntax:**
  DWORD DNM_WriteCyclicOutputData(BYTE BoardNo, BYTE Port, DWORD DesMacID, BYTE *DataBuf, DWORD DataLength)
- **Parameter:**

BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
Port: [input] CAN port number (refer to figure 1.5)
DestMACID: [input] The destination DeviceNet MAC ID (0~63)
DataBuf: [input] The data of the output channel
DataLength: [input] The length (in Byte) of the data

- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_WRITE_CYCLIC_OUTPUT_LEN_ERROR“
  “DNM_DEVICE_IS_UNCONNECTED“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.27 DNM_CheckCOSConnectionStatus

- **Description:**
  This function is used to check the status of the COS connection which is connected to the specific slave device.
- **Syntax:**
  DWORD DNM_CheckCOSConnectionStatus(BYTE BoardNo, BYTE Port, DWORD DesMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_NOT_CONFIG“
  “DNM_SET_COS_INPUT_LEN_ERROR“
  “DNM_SET_COS_OUTPUT_LEN_ERROR“
  “DNM_DEVICE_ERROR”
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.28 DNM_CheckCyclicConnectionStatus

- **Description:**
  This function is used to check the status of the Cyclic connection which is connected to the specific slave device.
- **Syntax:**
  DWORD DNM_CheckCyclicConnectionStatus(BYTE BoardNo, BYTE Port, DWORD DesMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_COS_CYCLIC_NOT_CONFIG“
  “DNM_SET_COS_INPUT_LEN_ERROR“
  “DNM_SET_COS_OUTPUT_LEN_ERROR“
  “DNM_DEVICE_ERROR”
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.29 DNM_StartDevice

- **Description:**
  This function is used to start to communicate with the destination slave device.
- **Syntax:**
  DWORD DNM_StartDevice(BYTE BoardNo, BYTE Port, DWORD DesMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”

“DNM_MASTER_STATE_ERROR“
“DNM_DEVICE_NOT_IN_SCANLIST“
“DNM_DEVICE_IS_CONNECTING“
“DNM_EXPLICITMSG_NOT_CONFIG“
“DNM_POLL_NOT_CONFIG“
“DNM_BITSTROBE_NOT_CONFIG“
“DNM_COS_CYCLIC_NOT_CONFIG“
“DNM_COS_CYCLIC_CONNECT_DUPLICATE“
“DNM_ADDTHREAD_ERROR“
“DNM_RESUMETHREAD_ERROR“
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code")

2.3.30 DNM_StopDevice

- **Description:**
  This function is used to stop to communicate with the destination slave device.
- **Syntax:**
  DWORD DNM_StopDevice(BYTE BoardNo, BYTE Port, DWORD DesMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_NOT_IN_SCANLIST“
  “DNM_DEVICE_IS_UNCONNECT“
  “DNM_EXPLICITMSG_NOT_CONFIG“
  “DNM_POLL_NOT_CONFIG“
  “DNM_BITSTROBE_NOT_CONFIG“
  “DNM_COS_CYCLIC_NOT_CONFIG“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.31 DNM_StartCommunicate

- **Description:**
  This function is used to start to communicate with all devices in scan list.
- **Syntax:**
  DWORD DNM_StartCommunicate(BYTE BoardNo, BYTE Port)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5).
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_IS_CONNECTING“
  “DNM_EXPLICITMSG_NOT_CONFIG“
  “DNM_POLL_NOT_CONFIG“
  “DNM_BITSTROBE_NOT_CONFIG“
  “DNM_COS_CYCLIC_NOT_CONFIG“
  “DNM_COS_CYCLIC_CONNECT_DUPLICATE“
  “DNM_ADDTHREAD_ERROR“
  “DNM_RESUMETHREAD_ERROR“
  “DNM_NOERROR”
  (Please refer to "Section 2.1 Error Code")

2.3.32 DNM_StopCommnuicate

- **Description:**
  This function is used to stop to communicate with all destination devices in scan list.
- **Syntax:**
  DWORD DNM_StopCommunicate(BYTE BoardNo, BYTE Port)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_DEVICE_IS_UNCONNECT“
  “DNM_EXPLICITMSG_NOT_CONFIG“
  “DNM_POLL_NOT_CONFIG“

“DNM_BITSTROBE_NOT_CONFIG“
“DNM_COS_CYCLIC_NOT_CONFIG“
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.33 DNM_GetAllDeviceMACID

- **Description:**
  This function is to get the MAC IDs of all devices in DeviceNet network.
- **Syntax:**
  DWORD DNM_GetAllDeviceMACID(BYTE BoardNo, BYTE Port, BYTE *AllMacID, BYTE *DeviceCount)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  AllMACID: [output] All of the destination DeviceNet MAC IDs
  DeviceCount: [output] The amount of the devices in the network
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_SENDMSG_ERROR”
  “DNM_NOERROR”
  (Please refer to Section 2.1 Error Code").

2.3.34 DNM_GetOwnership

- **Description:**
  This function is used to make the master port to get the ownership in DeviceNet network.
- **Syntax:**
  DWORD DNM_GetOwnership(BYTE BoardNo, BYTE Port)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_GET_OWNERSHIP_ERROR”
  “DNM_NOERROR”

(Please refer to "Section 2.1 Error Code").

2.3.35 DNM_CheckFaultNode

- **Description:**
  This function is used to check any fault nodes in DeviceNet network.
- **Syntax:**
  DWORD DNM_CheckFaultNode(BYTE BoardNo, BYTE Port)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_MASTER_NOT_GET_OWNERSHIP”
  “DNM_SENDMSG_ERROR”
  The amount of the fault node (0~63)
  (Please refer to "Section 2.1 Error Code").

2.3.36 DNM_GetAllFaultNode

- **Description:**
  This function is used to get the Serial No and the Vendor ID of all fault nodes in DeviceNet network.
- **Syntax:**
  DWORD DNM_GetAllFaultNode(BYTE BoardNo, BYTE Port, WORD *VendorID, DWORD *SerialNo, BYTE *NodeCount)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  VendorID: [output] The VendorID of all fault nodes
  SerialNo: [output] The SerialNo of all fault nodes
  NodeCount: [output] The amount of all fault nodes
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_MASTER_NOT_GET_OWNERSHIP”
  “DNM_NORESPONSE”
  “DNM_SENDMSG_ERROR”

“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.37 DNM_ChangeFaultMACID

- **Description:**
  The function is to change the MAC ID of the fault node in DeviceNet network if the slave device has the settable MAC ID.
- **Syntax:**
  DWORD DNM_ChangeFaultMACID(BYTE BoardNo, BYTE Port, WORD VendorID, DWORD SerialNo, BYTE NewMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  VendorID: [input] The VendorID of all fault nodes
  SerialNo: [input] The SerialNo of all fault nodes
  NewMACID: [input] The new MAC ID of the fault node
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MASTER_STATE_ERROR“
  “DNM_MASTER_NOT_GET_OWNERSHIP”
  “DNM_SENDMSG_ERROR”
  “DNM_NOERROR”
  (Please refer to Section 2.1 Error Code").

2.3.38 DNM_GetDeviceStatus

- **Description:**
  This function is used to check the status of the certain slave device.
- **Syntax:**
  DWORD DNM_GetDeviceStatus(BYTE BoardNo,BYTE Port,BYTE DesMacID)
- **Parameter:**
  BoardNo: [input] PISO-CAN board number (refer to figure 1.5)
  Port: [input] CAN port number (refer to figure 1.5)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)
- **Return:**
  “DNM_OPEN_ERROR”
  “DNM_MACID_ERROR”

“DNM_MASTER_STATE_ERROR“
“DNM_DEVICE_NOT_IN_SCANLIST“
“DNM_EXPLICITMSG_NOT_CONFIG“
“DNM_POLL_NOT_CONFIG”
“DNM_BITSTROBE_NOT_CONFIG”
“DNM_COS_CYCLIC_NOT_CONFIG”
“DNM_DEVICE_ERROR”
“DNM_NOERROR”
(Please refer to "Section 2.1 Error Code").

2.3.39 DNM_SetPollFrequency

- **Description:**
  This function is used to set the frequency of the Poll connection. The frequency is the refresh rate of Poll I/O signal. The frequency range is from 1 to 1000 Hz.
- **Syntax:**
  DWORD DNM_SetPollFrequency(WORD Frequency)
- **Parameter:**
  Frequency: [input] the Poll I/O refresh frequency (1~1000)
- **Return:**
  The frequency you have set.

2.3.40 DNM_SetBitStrobeFrequency

- **Description:**
  This function is used to set the frequency of the Bit-Strobe connection. The frequency is the refresh rate of Bit-Strobe I/O signal. The frequency range is from 1 to 1000 Hz.
- **Syntax:**
  DWORD DNM_SetBitStrobeFrequency(WORD Frequency)
- **Parameter:**
  Frequency: [input] the Bit-Strobe I/O refresh frequency (1~1000)
- **Return:**
  The frequency you have set.

3. PISO-CAN Series DeviceNet Library 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 “ixcan.inst” will setup the correct SocketCAN driver. After driver (version 0.3.1 or the later driver version) compiled and installation, the related CAN DeviceNet library, demo and header files for different development environments are presented as follows.

Driver Name	Directory Path	File Name	Description
ixcan-0.3.1	include	sja1000.h	SocketCAN driver header
		pisodnm.h	The Linux SocketCAN DeviceNet library header
	lib	libsktdnm.a	The static library of DeviceNet.
	examples/ pisocan	candnm_a.c	CAN DeviceNet Demo.

Table 3.1

3.1 Demo code “candnm_a”

Using the CAN DeviceNet master library function to provide “Explicit Message”, “Poll”, “BitStrobe” and “COS/Cyclic CAN DeviceNet object connection. Before user referred to below steps for DeviceNet application, user should install the hardware (the connection of cable) and software (the linux SocketCAN driver of PISO-CAN series board) of DeviceNet master (PISO-CAN series) and slave device (for example the CAN-8224) well.