PISO-CAN400-D/T
PISO-CAN200-D/T
DeviceNet Master Library

User's 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 assumes 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, or 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.

Contents

1.General Information

1.1 DeviceNet Introduction

The CAN (Controller Area Network) is a serial communication protocol, which efficiently supports distributed real-time control with a very high level of security. It is an especially suited for networking "intelligent" devices as well as sensors and actuators within a system or sub-system. In CAN networks, there is no addressing of subscribers or stations in the conventional sense, but instead, prioritized messages are transmitted. DeviceNet is one kind of the network protocols based on the CAN bus and mainly used for machine control network, such as textile machinery, printing machines, injection molding machinery, or packaging machines, etc. DeviceNet is a low level network that provides connections between simple industrial devices (sensors, actuators) and higher-level devices (controllers), as shown in Figure 1.1.

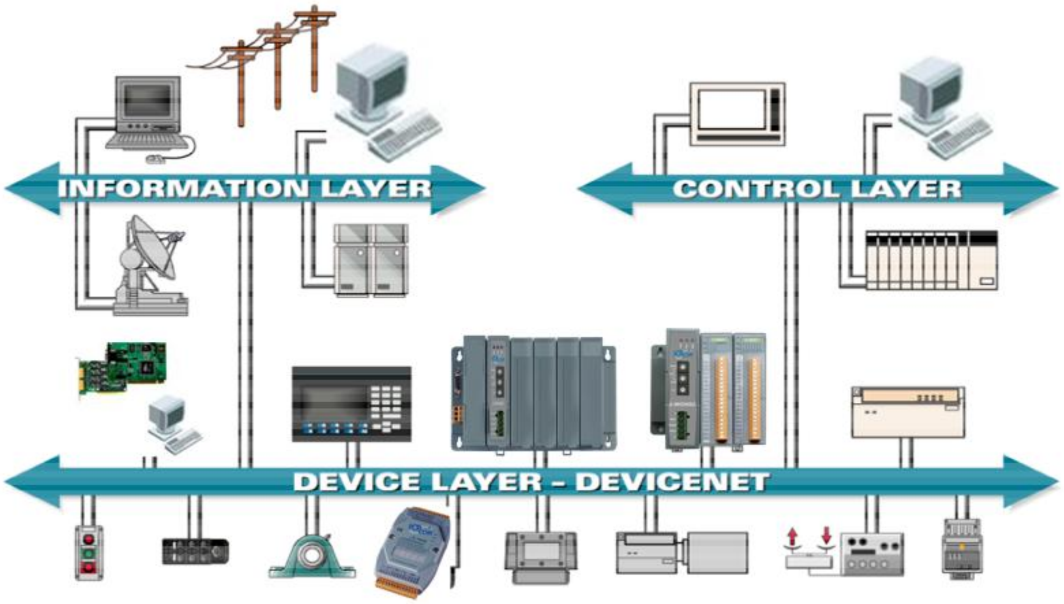


Figure 1.1 Example of the DeviceNet network

DeviceNet is a cost effective solution to one kind application of control c\area network. It reduces the connection wires between devices and provides rapid troubleshooting rejection function. The transfer rate can be up to 500Kbps within 100 meters. The transfer distance can be up to 500 meters in 125Kbps (See Table 1.1). It allows direct peer to peer data exchange between nodes in an organized and, if necessary, deterministic manner. Master/Slave connection model can be supported in the same network. Therefore, DeviceNet is able to facilitate all application communications based on a redefine a connection scheme. However, DeviceNet connection object strands as the communication path between multiple endpoints, which are application objects that is needed to share data.

Baud rate (bit/s)	Max. Bus length (m)
500 K	100
250 K	250
125 K	500

Table 1.1 The Baud rate and the Bus length

1.2 DeviceNet Applications

DeviceNet is the standardized network application layer optimized for factory automation. It is mainly used in low- and mid-volume automation systems. Some users have also implemented DeviceNet for machine control systems. The main DeviceNet application fields include the following application area (For more information, please refer to www.odva.org):

- Production cell builds and tests CPUs
- Beer brewery
- Equipment for food packing
- Fiberglass twist machine
- Sponge production plant
- Isolation wall manufacturing
- Overhead storage bin production
- Pocket-bread bakery
- Dinnerware production
- HVAC module production
- Textile machines
- Trawler automation system
- LCD manufacturing plant
- Rolling steel door production
- Bottling line
- Tight manufacturing

1.3 DeviceNet Master Library Characteristics

ICP DAS DeviceNet Master Library (DLL functions) provides users to establish DeviceNet network rapidly by Master/Slave connection model. The DeviceNet library is for PCI CAN interface card, which is PISO-CAN200/400. Using the library, users don’t need to take care of the detail of the DeviceNet protocol. The library will implement the DeviceNet protocol automatically. It can reduce the complexity of user’s DeviceNet Master Software. The library mainly supports the Predefined Master-Slave Connection Set functions to allow users to merge PISO-CAN200/400 into DeviceNet network. It can help users to establish the connection with DeviceNet slave devices easily. The general application architecture is demonstrated as Figure 1.2.

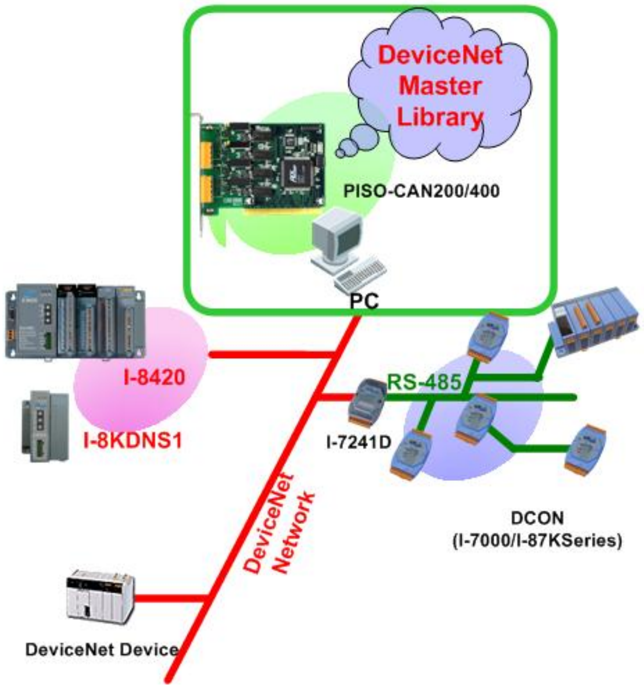


Figure 1.2 Application architecture

The library only provides the DeviceNet Master mechanism to communicate with slave devices by the Predefined Master/Slave Connection Set, which can be clarify as two forms: One is the Explicit Message and others are I/O Messages. Note that before communicating I/O data with DeviceNet slave devices, the Master device must connect to slave devices by explicit message connection to define the connection object. Here, we only provide one explicit message connection and four I/O connections as depicted in Figure 1.3.

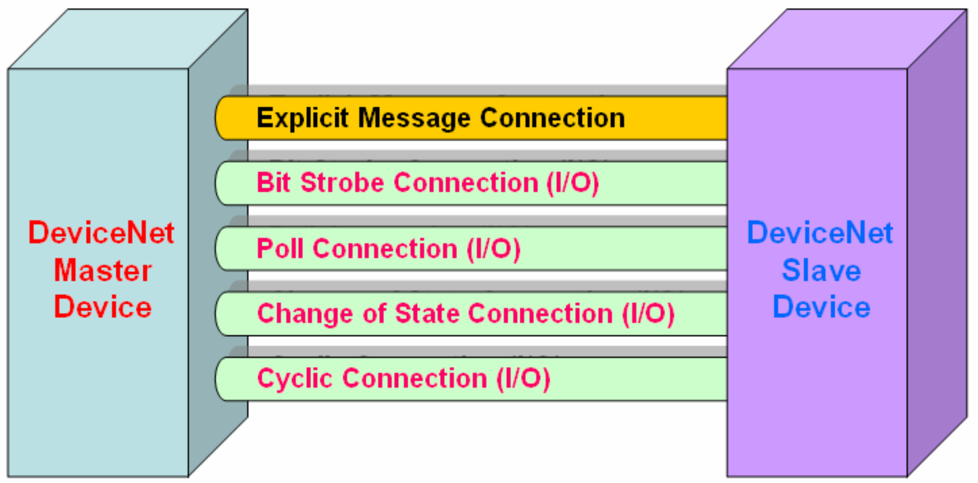

DeviceNet
Master
Device
Explicit Message Connection
Bit Strobe Connection (I/O)
Poll Connection (I/O)
Change of State Connection (I/O)
Cyclic Connection (I/O)
DeviceNet
Slave
Device

Figure 1.3 DeviceNet Messaging

The DeviceNet Communication Protocol is based on the concept of connections method. Master should create connections with slave devices based on the command of exchanging information and I/O data. To establish master control mechanism, there are only four main steps to be followed. Figure 1.4 demonstrates the basic process for the DeviceNet master communication. The every step function is described in below:

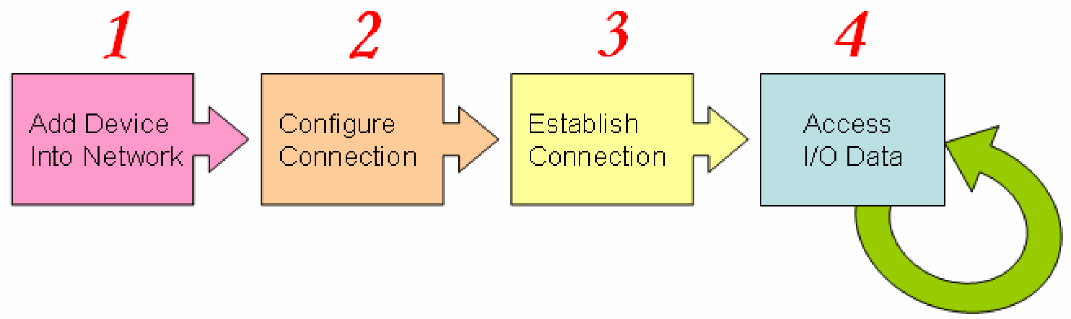


Figure 1.4 Four steps to establish connection

1. **Add device into network**
   You should provide the slave device's MAC ID to add into network.
2. **Configure connection**
   You can check the slave device's I/O connection type and the I/O data length. When configuring the I/O connection, you should provide these parameters.
3. **Establish connection**
   After configuring connections, users can start communicating with slave devices.
4. **Access I/O data**
   After communicating with slave devices, you can access the I/O data with corresponding read/write function.

After establishing the explicit connection, the connection path is then used to exchange the general information from one node to the others. And then users can create the I/O connections in the next step. Once I/O connections have been created, I/O data may be exchanged among devices in the DeviceNet network according to master device demand. Therefore, the master device can access I/O data of the slave devices by one of the four I/O connection methods. The library is not only easy to use but also providing a lot of the DeviceNet Master functions to retrieve and deliver the slave's I/O data. For more information, please refer to function description and demo programs in section 4.

1.4 Features

Hardware Features

- PCI BUS interface;
- 2500Vrms photo-isolation protection.
- Four or two independent CAN communication ports.
- Compatible with CAN specification 2.0 parts A and B.
- On-board optical isolation protection.
- Jumper select 120Ω terminator resistor for each port.
- Direct memory mapping to the CAN controllers.
- 33MHz 32bit 5V PCI bus (V2.1) plug and play technology.

DeviceNet Library Features

- Driver supported for Windows 98/ME/NT4/2000/XP
- Programmable transfer-rate 125K, 250K, 500K.
- Each Port support maximum nodes up to 64
- Support Group 2 Only Server functions
- Predefined Master-Slave Connection Set
- Support Offline Connection Set
- The maximum Fragment number is (Input/Output) up to 64
- Support I/O Operation Mode: Poll, Bit-Strobe and Change Of State/Cyclic
- Programmable MAC ID Setting

1.5 Specifications

- CAN controller: Phillips SJA1000T.
- CAN transceiver: Phillips 82C250/251.
- Signal support: CAN_H, CAN_L.
- 16 MHz CAN controller frequency.
- Connector: 5-pin screw terminal connector or 9-pin D-sub male connector.
- Isolation voltage: 2500Vrms.
- Power requirements:

  CAN400: 5V@640mA

  CAN200: 5V@380mA
- Environmental:

  Operating temp: 0~60℃
  Storage temp: -20~80℃
  Humidity: 0~90% non-condensing

  Dimensions: 130mm X 110mm

1.6 Product Check List

In addition to this manual, the package includes the following items:

- □ PISO-CAN400/CAN200 card;
- □ ADP-9 Board (for PISO-CAN400 only)
- □ Software CD ROM;
- □ User manual.

It is recommended that users should read the release note first. All of the important information needed will be provided in the release note as follows:

- □ Where you can find the software driver, utility and demo programs.
- □ How to install software & utility.
- □ Where is the diagnostic program?
- □ FAQ's and answers.

Attention !

If any of these items are missing or damaged, please contact your local field agent. Keep aside the shipping materials and carton in case you want to ship or store the product in the future.

2.Hardware Configuration

This section will describe the hardware settings of the PISO-CAN200/400, which includes the settings for both the PISO-CAN400 and the PISO-CAN200.

2.1 Board Layout

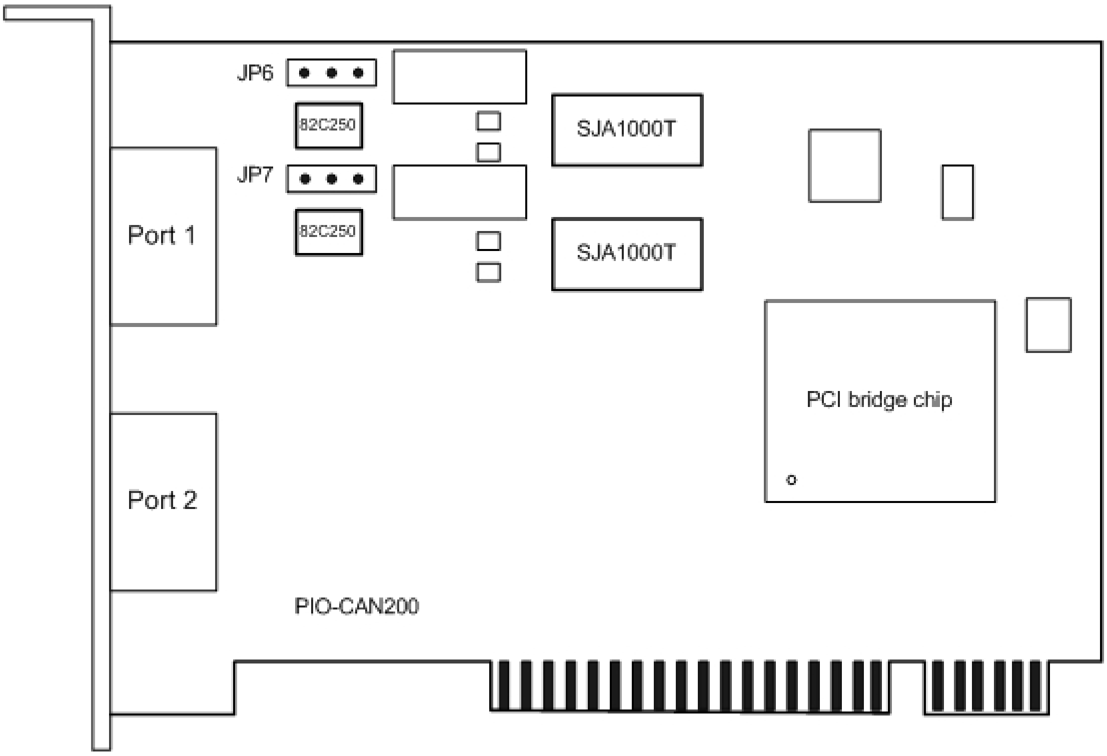


Figure2.1 PISO-CAN200 Board LAYOUT

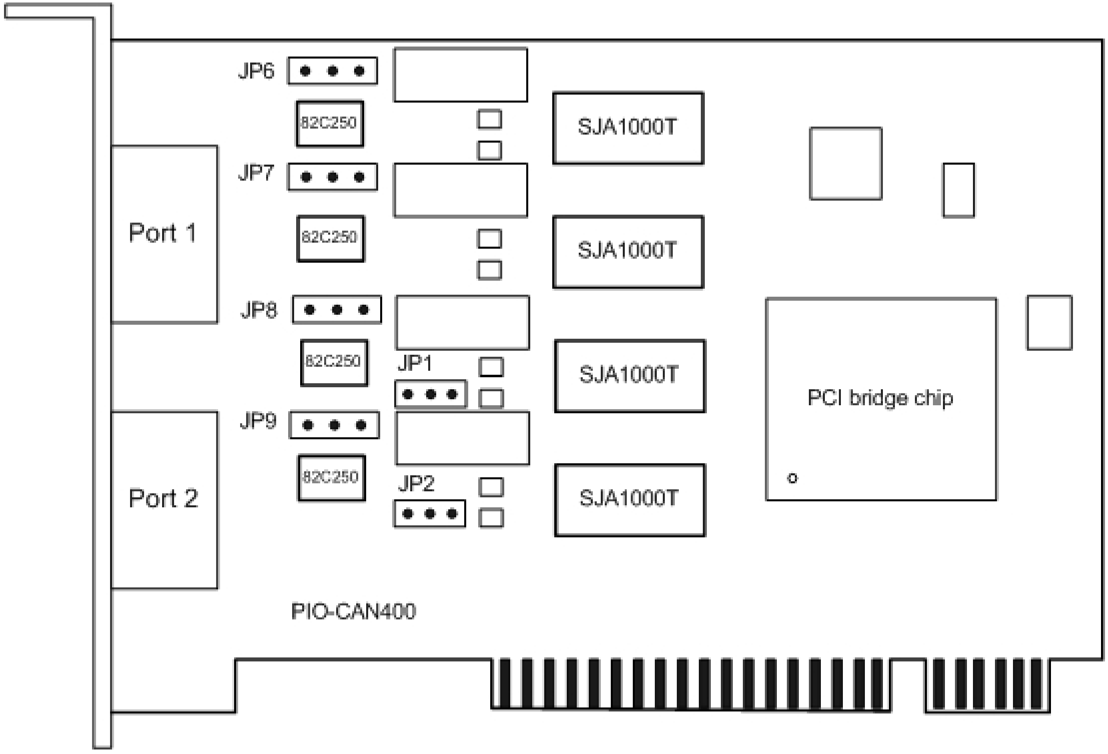


Figure2.2 PISO-CAN400 Board LAYOUT

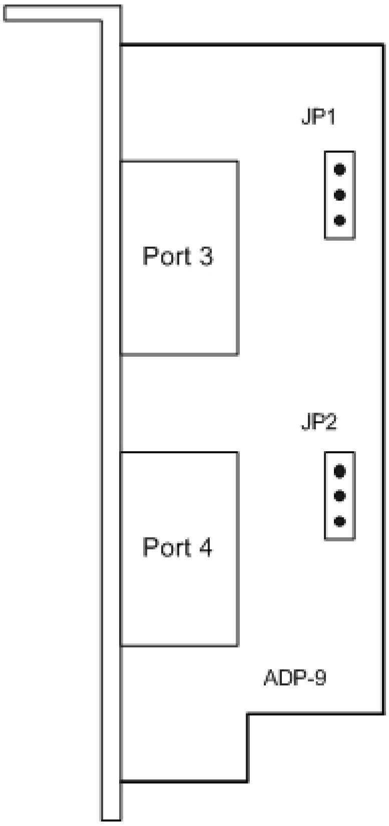
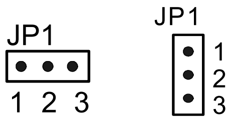
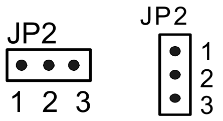
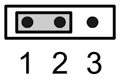
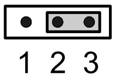
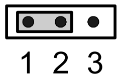
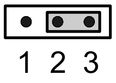
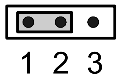
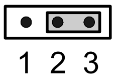
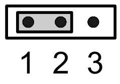
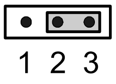


Figure2.3 ADP-9 Board LAYOUT (For PISO-CAN400 Only)

2.2 Jumper Selection

Table 2.1 Jumper Selections

Jumper	Description	Status	
JP1	CAN Port 3 Connector, connecting PISO-CAN400 board and ADP-9 board.	 Pin1: CAN_L Pin2: CAN_H Pin3: Shield	
JP2	CAN Port 4 Connector, connecting PISO-CAN400 board and ADP-9 board.	 Pin1: CAN_L Pin2: CAN_H Pin3: Shield	
		Enable	Disable
JP6	Port 1 terminator resister(120Ω) selection		
JP7	Port 2 terminator resister(120Ω) selection		
JP8	Port 3 terminator resister(120Ω) selection		
JP9	Port 4 terminator resister(120Ω) selection		

2.3 Connector Pin Assignment

The PISO-CAN400-T/PISO-CAN200-T and PISO-CAN400-D/PISO-CAN200-D are equipped with four/two sets of **5-pin screw terminal connectors** and of **9-pin D-sub male connectors** for wire connection of the CAN bus, respectively. The connector's pin assignment is specified as follows:

2.3.1 5-pin screw terminal connector

The 5-pin screw terminal connector for the CAN bus is shown in Figure 2.4 and the details for the pin assignment are presented in Table 2.2.

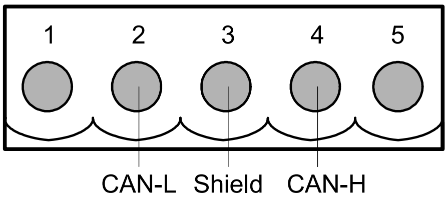


Figure2.4 5-pin screw terminal connector

Table 2.2: Pin assignment of 5-pin screw terminal connector

5-pin screw terminal connectors pin assignment	
1	No Use
2	CAN_L
3	Shield
4	CAN_H
5	No Use

2.3.2 9-pin D-sub male connectors

The 9-pin D-sub male connector of the CAN bus interface is shown in Figure 2.5 and the corresponding pin assignments are given in Table 2.3.

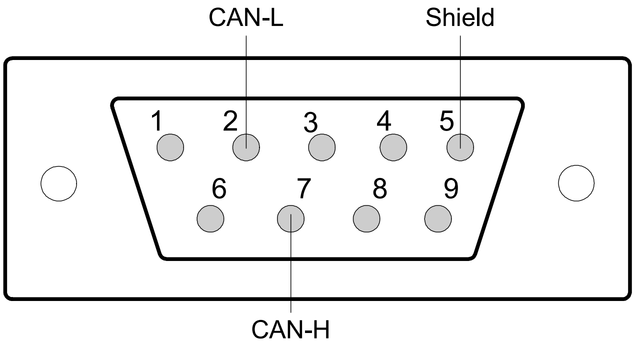


Figure2.5 9-pin D-sub male connector

Table 2.3 Pin assignment of the 9-pin D-sub male connector

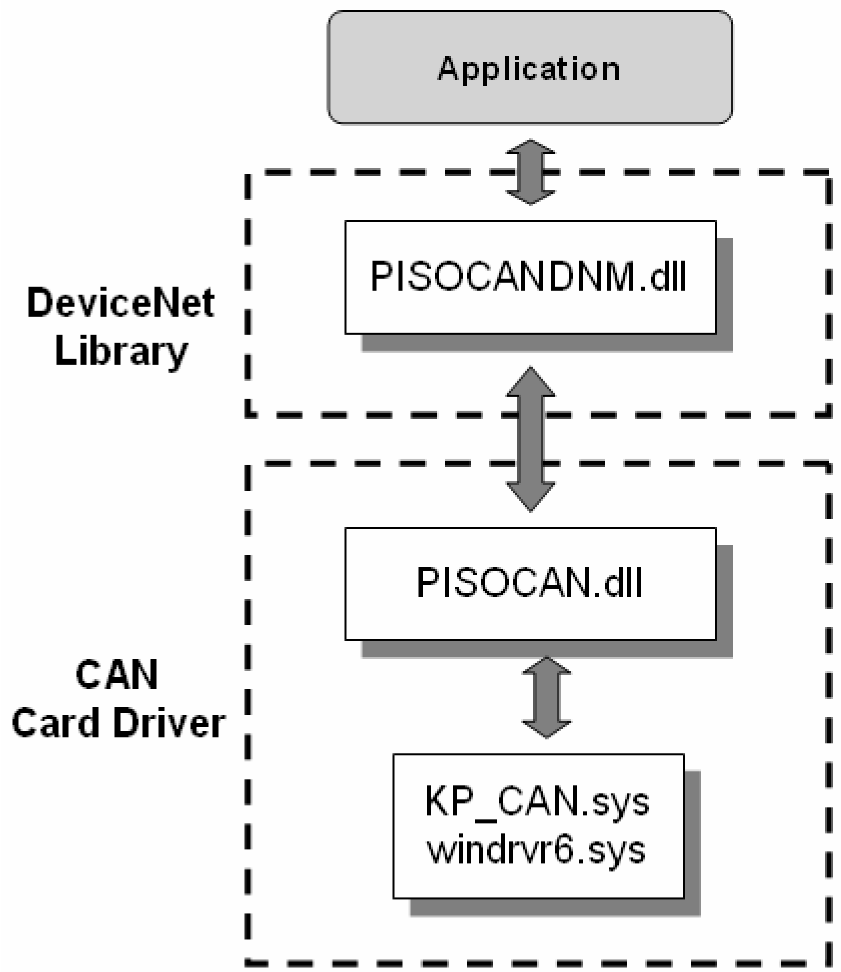
D-sub male connector pin assignment	
1	No Use
2	CAN_L
3	No Use
4	No Use
5	Shield
6	No Use
7	CAN_H
8	No Use
9	No Use

2.4 Hardware Installation

1. Configure the jumper settings on your PISO-CAN400/CAN200 in accordance with your particular requirements.
2. Shutdown your system and take off the chassis of your host computer.
3. Plug in your PISO-CAN400/CAN200 into a suitable available PCI slot.
4. Replace your chassis.
5. Plug your CAN bus cable(s) into the 5-pin screw terminal connector or the 9-pin D-sub connector.
6. When the hardware installation is complete, please turn on the computer again.

3. Software Installation

The DeviceNet DLL driver is the collection of function calls for the PISO-CAN400/CAN200 cards used in Windows 98/Me/NT/2000/XP systems. The application structure is presented in the following figure. The user's DeviceNet application programs can be developed by the following designated tools: VB, Delphi and Borland C++ Builder…etc. In these tools, the application program can call the PISOCANDNM.DLL driver to implement DeviceNet network application. And then the DeviceNet DLL driver will throughout the PISOCAN.dll into the KP_CAN.sys and windrvr6.sys to access the hardware system, as shown in the following Figure.



3.1 DLL Function Definition and Description

The software Installation for DeviceNet application has two main steps. First, the users should install the CAN card driver. Next, the users should install the DeviceNet Library. After finishing the installation process, the user can refer to demo program and follow its hardware application structure to testing the basic functions of master/slave connection. For the advance application, user can refer to the basic demo program to develop their own DeviceNet master application.

[First]：Install CAN Card Driver

The PISO-CAN400/CAN200 can be used in 98/Me/NT/2000/XP Windows environments. For these Windows operation systems, the recommended installation procedure is given as follows:

Step 1: Insert the companion CD into the CD-ROM driver and wait a few seconds until the installation program starts automatically. If it cannot be started automatically for some reason, please double-click the file **\NAPDOS\AUTO32.EXE** on this CD.

Step 2: Click the first item; Toolkits (Software) / Manuals.

Step 3: Click the item PCI Bus DAQ Card.

Step 4: Click PISO-CAN 400/200.

Step 5: Click “install Toolkit for Windows 98 (Or Me, NT, 2000, XP), which is based on the operation system you used”.

Then the setup file will start the driver installation process and copy the related material to the indicated directory and then register the driver on your computer. The driver target directory is as below for the different systems.

Windows NT/2000 – **WINNT\SYSTEM32\DRIVERS**

Windows 98/Me/XP – **WINDOWS\SYSTEM32\DRIVERS**

[Second]：Install DeviceNet Library

Step 1: Insert the companion CD into the CD-ROM driver and wait a few seconds until the installation program starts automatically. If it cannot be started automatically for some reason, please double-click the file **\NAPDOS\AUTO32.EXE** on this CD.

Step 2: Click the first item; Toolkits (Software) / Manuals.

Step 3: Click the item PCI CAN Card.

Step 4: Click PISO-CAN 400/200.

Step 5: Click “DeviceNet Lib”

Step 6: Click “install Toolkit for Windows 98 (Or Me, NT, 2000, XP), which is based on the operation system you used”.

Then the setup file will start the driver installation process and copy the related material to the indicated directory and then register the driver on your computer. The driver target directory is default as below.

C:\DAQPro\PISO-CAN\DeviceNet\

4.Function description

4.1 DLL Function Definition and Description

All the functions provided in the PISOCANDNM.dll are listed in the following table and detail information for every function is presented in the next sub-section. However, in order to make the descriptions more simply and clear, the attributes for the both the input and output parameter functions are given as **[input]** and **[output]** respectively, as shown in following table.

Keyword	Set parameter by user before calling this function?	Get the data from this parameter after calling this function?
[input]	Yes	No
[output]	No	Yes

Functions Table

No.	Function Name	Description
1	DNM_GetVersion	Get the version of the PISOCANDNM.dll
2	DNM_ActiveBoard	Activate the device
3	DNM_CloseBoard	Stop and close the kernel driver
4	DNM_Online	Make the CAN port Online
5	DNM_Offline	Make the CAN port Offline
6	DNM_ChangeMACID	Change Master MAC ID
7	DNM_ChangeBaudrate	Change the baud rate
8	DNM_AddDevice	Add devices into the DeviceNet network
9	DNM_RemoveDevice	Remove devices from the DeviceNet network
10	DNM_CofigExplicitMsg	Configure the Explicit Message connection
11	DNM_SetAttribute	Set the attribute of the device's instance
12	DNM_GetAttribute	Get the attribute of the device's instance
13	DNM_CheckExplicitMsgConnectionStatus	Check Explicit Message connection status
14	DNM_ConfigPoll	Configure the Poll connection
15	DNM_CheckPollConnectionStatus	Check the Poll connection status

Functions Table (continue)

No.	Function Name	Description
16	DNM_ReadPollInputData	Read the data from the Poll input channel
17	DNM_WritePollOutputData	Write the data to the Poll output channel
18	DNM_ConfigBitStrobe	Configure the Bit-Strobe connection
19	DNM_ReadBitStrobe	Read the data from Bit-Strobe input channel
20	DNM_CheckBitStrobeConnectionStatus	Check the Bit-Strobe connection status
21	DNM_ConfigCOS	Configure the Change-Of-State connection
22	DNM_ConfigCyclic	Configure the Cyclic connection
23	DNM_ReadCOSInputData	Read data from the COS input channel
24	DNM_WriteCOSOutputData	Write data to the COS output channel
25	DNM_ReadCyclicInputData	Read data from the Cyclic input channel
26	DNM_WriteCyclicOutputData	Write data to the Cyclic output channel
27	DNM_CheckCOSConnectionStatus	Check the COS connection status
28	DNM_CheckCyclicConnectionStatus	Check the Cyclic connection status
29	DNM_StartDevice	Start to communicate to specific device
30	DNM_StopDevice	Stop to communicate to specific device
31	DNM_StartCommunicate	Start to communicate to all devices
32	DNM_StopCommunicate	Stop to communicate to all devices
33	DNM_GetAllDeviceMACID	Get all device’s MAC ID in the network
34	DNM_GetOwnership	Get the ownership in the DeviceNet network
35	DNM_CheckFalutNode	Check any fault node in the network
36	DNM_GetAllFaultNode	Get the SerialNo and VendorID of all fault nodes in the DeviceNet network
37	DNM_ChangeFaultMACID	Change the MAC ID of specific fault node

4.2 FunctionReturn Code

Interpretation of the return code

Return Code	Error ID	Comment
0	CAN_NoError	OK
1	CAN_DriverError	Driver error
2	CAN_ActiveBoardError	This board can't be activated.
3	CAN_BoardNumberError	The board number exceeds the maximum board number (7).
4	CAN_PortNumberError	The port number exceeds the maximum port number.
5	CAN_ResetError	CAN chip hardware reset error
6	CAN_SoftResetError	CAN chip software reset error
7	CAN_InitError	CAN chip initiation error
8	CAN_ConfigError	CAN chip configure error
9	CAN_SetACRError	Set to Acceptance Code Register error
10	CAN_SetAMRError	Set to Acceptance Mask Register error
11	CAN_SetBaudRateError	Set Baud Rate error
12	CAN_EnableRxIrqFailure	Enable CAN chip receive interrupt failure
13	CAN_DisableRxIrqFailure	Disable CAN chip receive interrupt failure
14	CAN_InstallIrqFailure	Installing PCI board IRQ failure
15	CAN_RemoveIrqFailure	Removing PCI board IRQ failure
16	CAN_TransmitBufferLocked	Transmit buffer in CAN chip is locked
17	CAN_TransmitIncomplete	Previously transmission is not yet completed
18	CAN_ReceiveBufferEmpty	CAN chip RXFIFO is empty
19	CAN_DataOverrun	Data was lost because there was not enough space in CAN chip RXFIFO
20	CAN_ReceiveError	Receive data is not completed
21	CAN_SoftBufferIsEmpty	Software buffer in driver is empty
22	CAN_SoftBufferIsFull	Software buffer in driver is full

Interpretation of the return code (Continue)

Return Code	Error ID	Comment
1100	DNM_TxMessageConnectionIDError	Message ID is illegal
1101	DNM_TxMessageRTRError	RTR is out of range(0~1)
1102	DNM_TxMessageLengthError	Data length exceeds the range between 0 and 8
1200	DNM_DuplicateMACIDTimeout	It is timeout when waiting for the duplicate MAC ID message
1201	DNM_WaitBufferResponseTimeout	It is timeout when waiting for the destination device response
1202	DNM_BaudrateError	The Baud Rate exceeds the range between 0 and 2
1203	DNM_MACIDNumberError	The MAC ID exceeds the range between 0 and 63
1204	DNM_OnlineError	The Master MAC ID is duplicate in the DeviceNet network
1205	DNM_OfflineError	The Offline function fail
1206	DNM_MasterStateError	The Master is in illegal state (note1)
1300	DNM_BoardNumberError	The board number exceeds the range between 0 and 7
1301	DNM_BoardAlreadyClose	The board is already closed
1302	DNM_BoardAlreadyActive	The board already active
1303	DNM_BoardNotActive	The board does not active.
1400	DNM_PortAlreadyOnline	The port is already online
1401	DNM_PortNumberError	The port number exceed the range between 0 ~ 3(or 0~1 for 2 ports)
1500	DNM_ChangeMACIDError	Change Master MAC ID fail
1501	DNM_ChangeBaudrateError	Change Master Baud Rate error
1600	DNM_SetAttributeRequestError	Send SetAttribute request Message fail
1601	DNM_SetAttributeResponseError	The device response error when setting attribute
1602	DNM_SetAttributeResponseTimeout	The device doesn't response within specific time

Interpretation of the return code (Continue)

Return Code	Error ID	Comment
1603	DNM_SetAttributeDataIsEmpty	The data is empty when setting attribute
1604	DNM_SetAttributeACKError	SetAttribute ACK message error
1605	DNM_SetAttributeFragmentACKTimeout	It is timeout when waiting for ACK message
1700	DNM_GetAttributeRequestError	Send GetAttribute request message fail
1701	DNM_GetAttributeResponseError	The device response error when getting attribute
1702	DNM_GetAttributeResponseTimeout	The device doesn’t response within specific time.
1703	DNM_GetAttributeFragmentTimeout	The device doesn’t response within specific time
1800	DNM_DeviceAlreadyInScanList	The device is already in Master’s ScanList (note2)
1801	DNM_DeviceNotInScanList	The device is not in Master’s ScanList (note2)
1802	DNM_DeviceIsUnConnected	The device doesn’t establish the connection. Maybe the configuration has some error.
1803	DNM_DeviceIsConnecting	The device is connecting now
1900	DNM_ExplicitMsgNotConfigure	The device doesn’t configure the Explicit Message connection.
1901	DNM_PollNotConfigure	The device doesn’t configure the Poll connection.
1902	DNM_PollNoResponse	The Poll connection has no response
1903	DNM_PollResponseDataError	The Poll connection response error message.
1904	DNM_PollFragmentError	The device has fragment errors when sending Poll I/O data
1905	DNM_WritePollOutputDataLengthError	The data length is error when sending Poll I/O data

Interpretation of the return code (Continue)

Return Code	Error ID	Comment
1906	DNM_BitStrobeNotConfigure	The device doesn't configure the Bit-Strobe connection.
1907	DNM_COS_CyclicNotConfigure	The device doesn't configure the COS/Cyclic connection.
2000	DNM_CreateThreadError	The Master can not create thread.
2001	DNM_ResumeThreadError	The Master can't resume thread.
2002	DNM_SuspendThreadError	The Master can't suspend thread.
2003	DNM_TerminateThreadError	The Master can't terminate thread
2004	DNM_CloseHandleError	The Master has errors when closing the thread handle.
2005	DNM_ThreadNotExist	The Receive thread doesn't exist.
2100	DNM_CreateDeviceBitStrobeThreadError	The Master has errors when creating Bit-Strobe thread.
2101	DNM_BitStrobeRequestThreadNotExist	The Master has errors when creating Bit-Strobe requester thread.
2102	DNM_CloseBitStrobeRequestError	The Master has errors when closing Bit-Strobe requester thread.
2103	DNM_TerminateBitStrobeRequestThreadError	The Master has errors when terminating Bit Strobe requester thread.
2104	DNM_SuspendBitStrobeRequestThreadError	The Master has errors when suspending Bit Strobe requester thread
2105	DNM_ResumeBitStrobeRequestThreadError	The Master has errors when resuming Bit Strobe requester thread
2106	DNM_BitStrobeNoResponse	The Bit-Strobe connection has no response
2107	DNM_BitStrobeResponseDataError	The Bit-Strobe connection response error messages.
2200	DNM_CreateDeviceCOS_CyclicThreadError	The Master can't create COS/Cyclic thread.

Interpretation of the return code (Continue)

Return Code	Error ID	Comment
2201	DNM_COS_Cyclic_NoResponse	The COS/Cyclic connection has no response
2202	DNM_COS_Cyclic_ResponseDataError	The COS/Cyclic connection response error messages
2203	DNM_COS_Cyclic_FragmentError	The COS/Cyclic connection has fragment errors when sending COS/Cyclic I/O data
2204	DNM_WriteCOSOutputDataLengthError	The data length is error when sending COS I/O data
2205	DNM_WriteCyclicOutputDataLengthError	The data length is error when sending Cyclic I/O data.
3000	DNM_CreateDeviceExplicitMsgThreadError	The Master has errors when creating Explicit Message connection thread.
3100	DNM_CreateDevicePollThreadError	The Master has errors when creating Poll connection.
3200	DNM_CreateDeviceBitStrobeRequestThreadError	The Master has errors when creating Bit Strobe connection thread.
3300	DNM_COS_Cyclic_ConnectionDuplicate	It is illegal when configuring both COS and Cyclic connections in one device.
3500	DNM_COS_CyclicAlreadyConfigure	This message means that the COS/Cyclic connection has been configured and started device, so the Poll connection can not be configured.
4000	DNM_GetOwnershipFail	The Master can not get the ownership.
4001	DNM_MasterNotGetOwnership	The Master has not got the ownership yet.

4.3 Flow Diagram for DeviceNet Application

In this section, we will show the operation procedure of the DeviceNet for the explicit message and I/O connection in the application. In the following sub-section, we will show some flow diagrams to describe how to apply the DeviceNet library in order to build a master device. Figure 4.1 presents the “SetAttribute and GetAttribute” procedure by Explicit Message Connection. Figure 4.2 introduces the Poll Connection (I/O Connection). Figure 4.3 stands as the Bit-Strobe connection (I/O Connection). The Change Of State Connection (I/O Connection) is depicted in Figure 4.4. Figure 4.5 shows the Cyclic Connection (I/O Connection). Note that users need to follow the operation principle of the DeviceNet protocal for correctly and easily access remote data by these connection methods. For more detail information, please refer to the demo programs in section 5.

4.3.1 The “SetAttribute and GetAttribute” procedure

The flow diagram, Fig 4.1, depicts how to get/set the attribute value of DeviceNet slave devices via the explicit connection. Users can follow the procedure to implement the program.

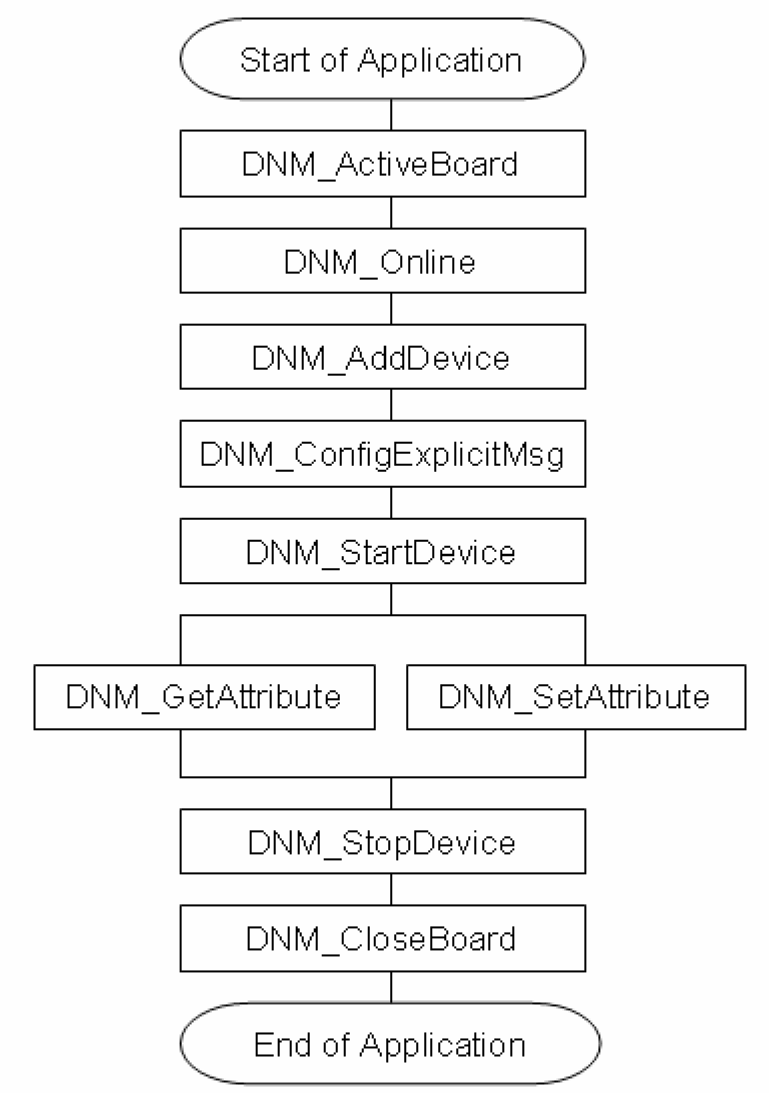


Figure 4.1 Flow Diagram of the Explicit Message Connection

4.3.2 The Poll Connection procedure

The Figure 4.2 demonstrates how to establish a poll connection with DeviceNet slave devices. Users can refer to the procedure to implement the program.

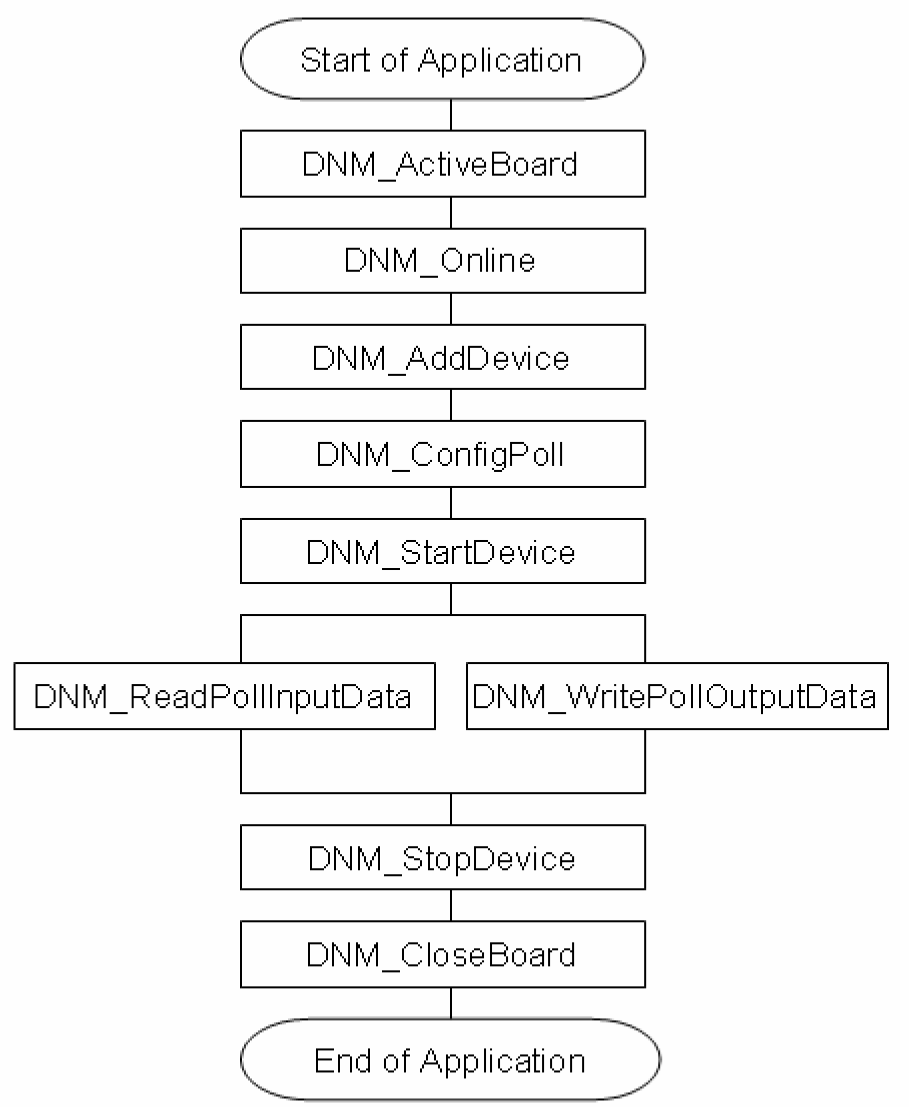


Figure 4.2 Flow Diagram of the Poll Connection

4.3.3 The Bit-Strobe Connection procedure

The flow diagram, Figure 4.3, represents how to establish a Bit-Strobe connection with DeviceNet slave devices. Users can follow the process to implement the program.

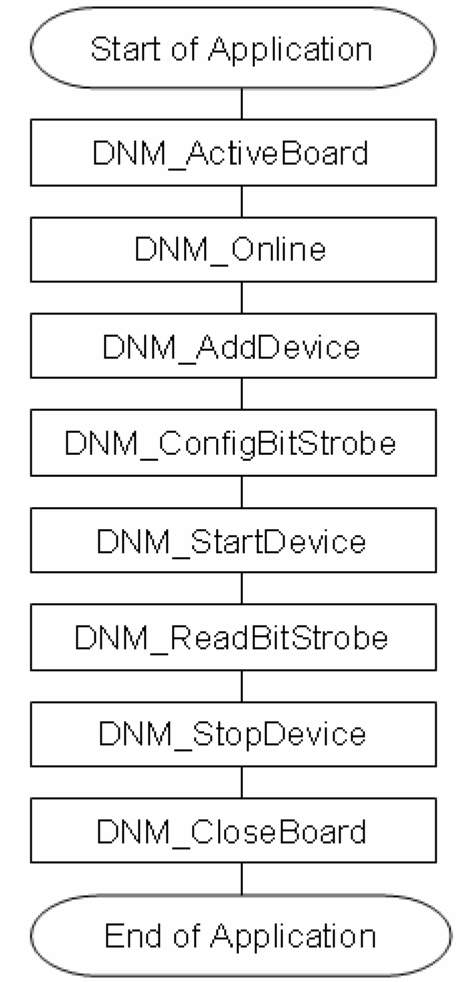


Figure 4.3 Flow Diagram of the Bit-Strobe Connection

4.3.4 The Change Of State Connection procedure

The following flow diagram presents how to establish a Change Of State connection with DeviceNet slave devices. Users can follow the process to implement the program.

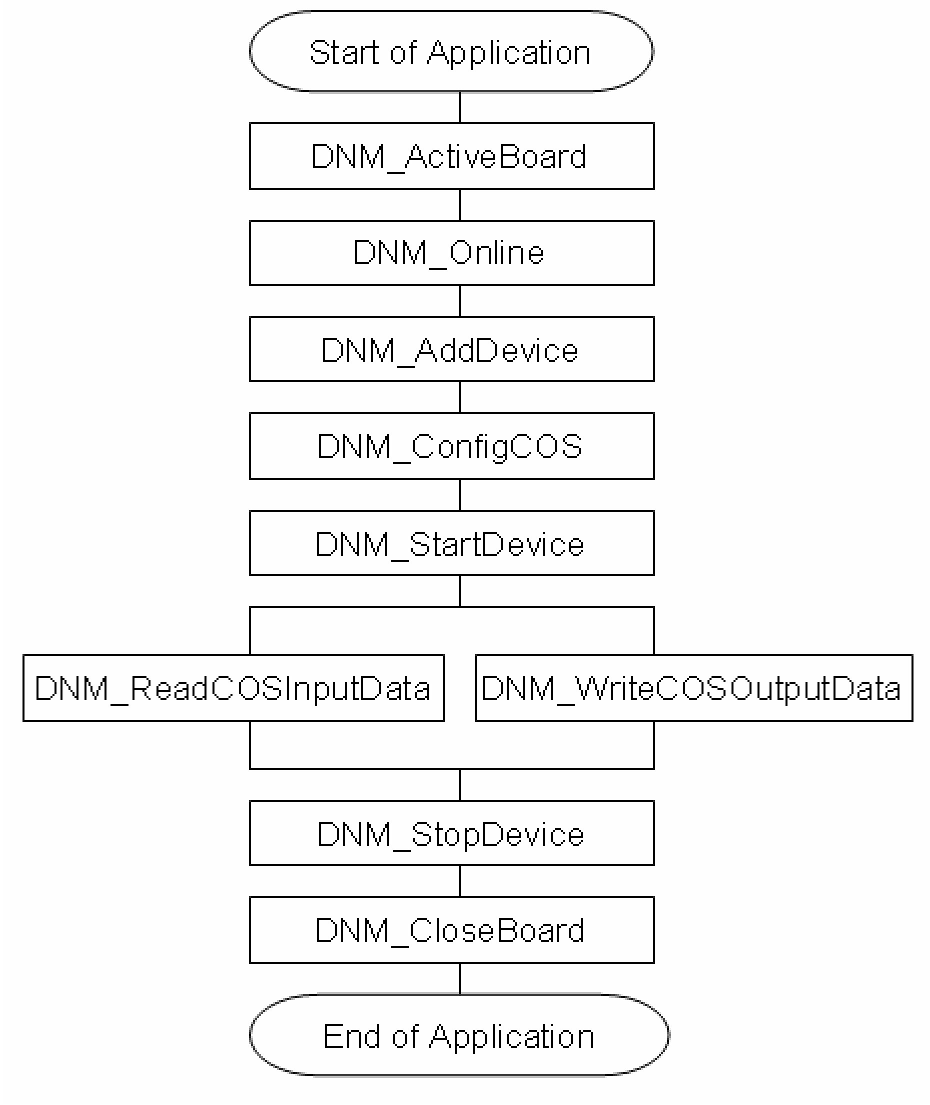


Figure 4.4 Flow Diagram of the Change Of State Connection

4.3.5 The Cyclic Connection procedure

The following flow diagram is applied to establish a Cyclic connection with DeviceNet slave devices. Users can follow the process to implement the program.

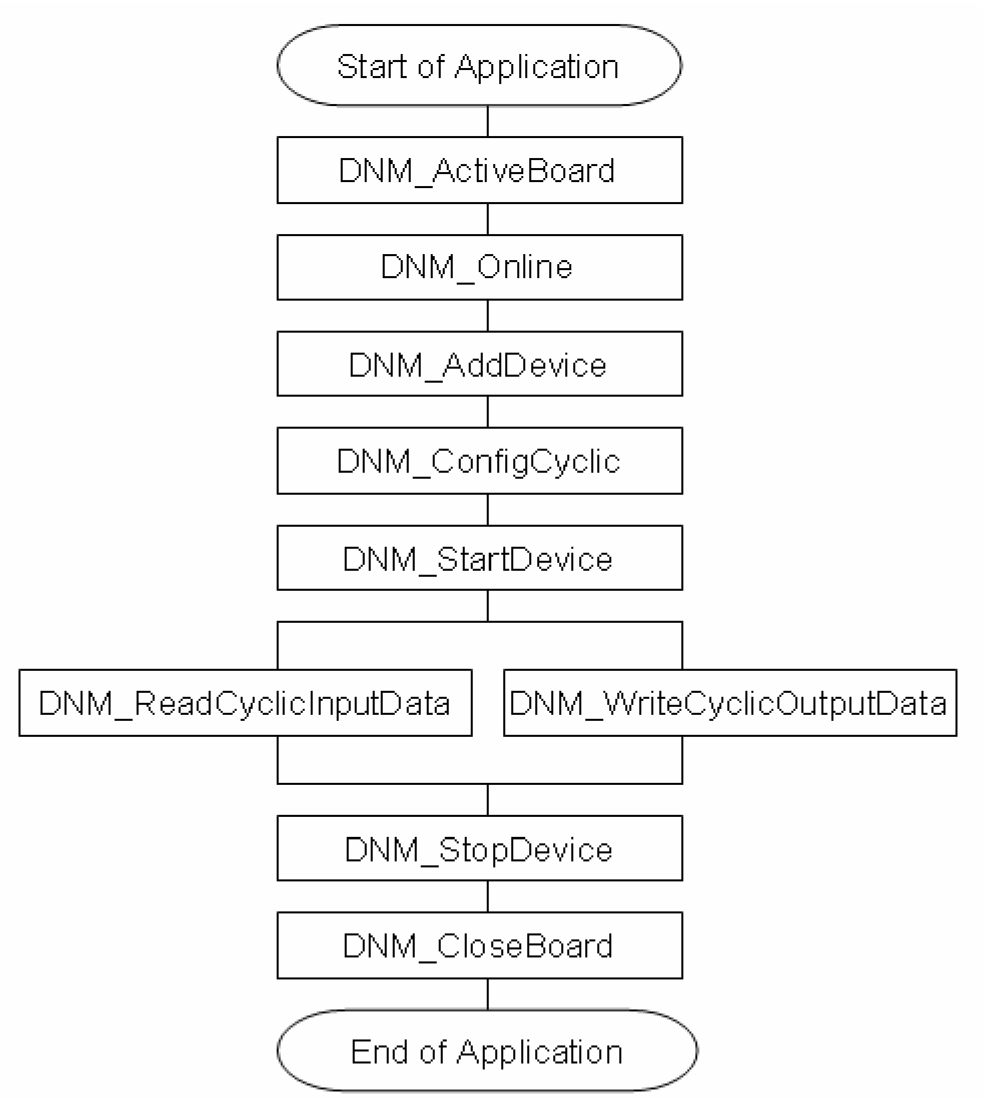
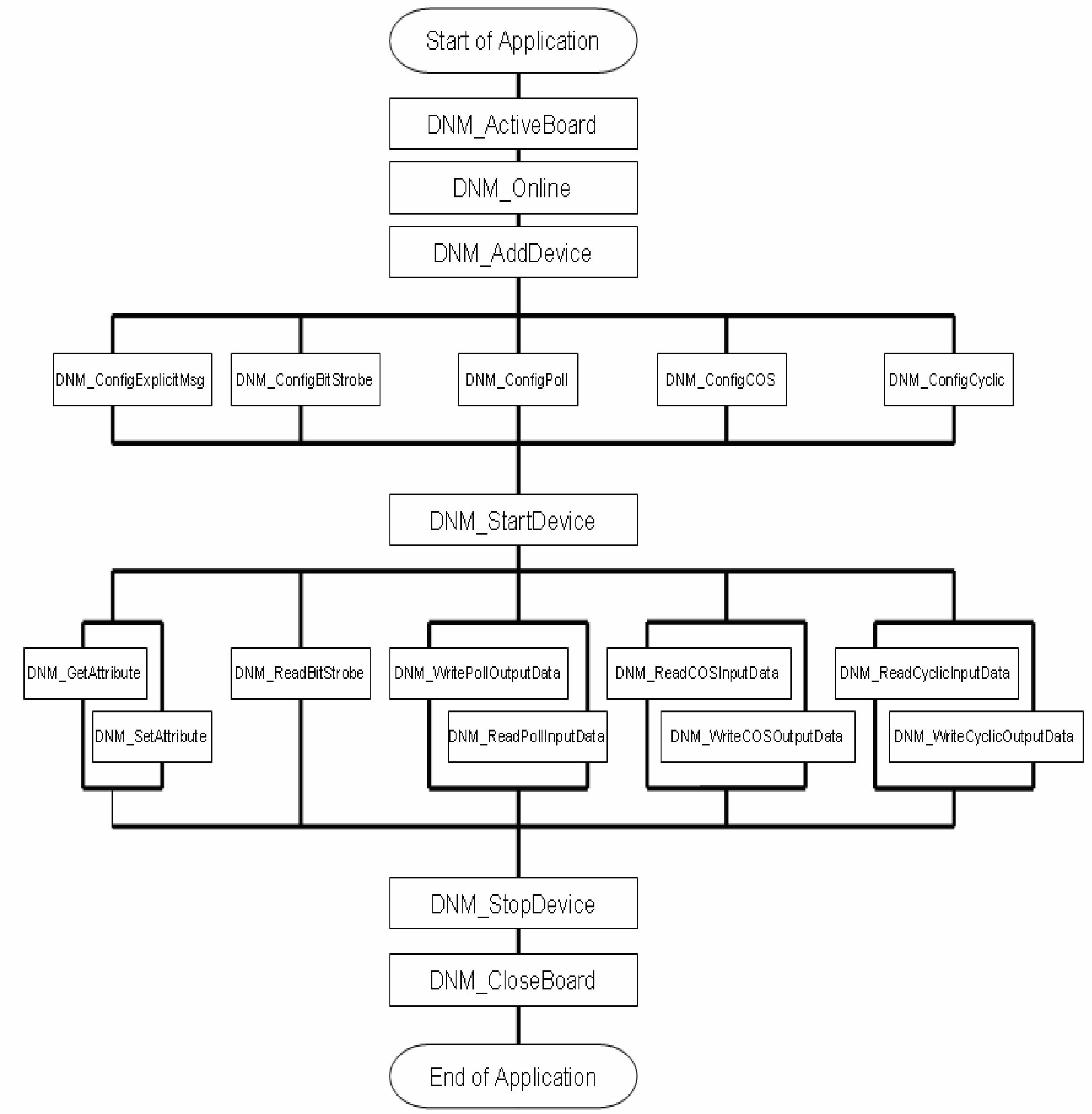


Figure 4.5 Flow Diagram of the Cyclic Connection

4.3.6 The designed flow diagram

The following flow diagram mainly presents how to program the DeviceNet application by the DeviceNet library. Users just need to integrate the function calls of deviceNet library to develop a deviceNet master device. For more information, please refer to the function description and demo program source code.



4.4 Function Description

4.4.1 DNM_GetVersion

- **Description:**

This function is used to obtain the version information of PISOCANDNM.dll driver.

- **Syntax:**

```
float DNM_GetVersion(void)
```

- **Parameter:**

None

- **Return:**

DLL version information.

4.4.2 DNM_ActiveBoard

- **Description:**

  The function can activate the PISO-CAN 200/400. It must be called once before using other functions of PISOCANDNM.dll.

- **Syntax:**

  WORD DNM_ActiveBoard(BYTE BoardNo)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the total board number
  DNM_BoardAlreadyActive: BoardNo has been activated.
  CAN_ActiveBoardError: This board can not be activated or kernel driver can not be found.

4.4.3 DNM_CloseBoard

- **Description:**

  The function can stop and close the kernel driver to release the device resource from system resource. This method must be called once before exiting the user's application program.

- **Syntax:**

  WORD DNM_CloseBoard(BYTE BoardNo)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the total board number
  DNM_BoardAlreadyClose: BoardNo has been closed.
  CAN_ActiveBoardError: This board is not activated.

4.4.4 DNM_Online

- **Description:**

  The function can make the CAN Port Online. When calling this function, the CAN Port will check duplicate MAC ID. After the function is used successfully, the CAN port become the DeviceNet master port. Then, the port will also transfer into Online State in the DeviceNet network.

- **Syntax:**

```
WORD DNM_Online(BYTE BoardNo , BYTE Port,
                           BYTE Baudrate , BYTE Macid)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  Baudrate: [input] DeviceNet baud rate (0~2)
  0→500Kbps 1→250Kbps 2→125Kbps
  Macid: [input] DeviceNet Master MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_PortAlreadyOnline: This CAN port is already online.
  DNM_MACIDNumberError: Macid exceeds the range from 0 to 63
  DNM_BaudrateError: Baudrate exceeds the range form 0 to 2
  DNM_OnlineError: The CAN port can not transfer into online state. It could be the Mac Id value is duplicated in DeviceNet or the wire is not connected to the DeviceNet network.
  CAN_DriverError: Kernel driver can not be opened.
  CAN_InitError: Initiating CAN controller failure.

4.4.5 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 also transfer into Offline State in DeviceNet network. In this status, the CAN port will not communicate with any slave devices.

- **Syntax:**

  WORD DNM_Offline(BYTE BoardNo , BYTE Port)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)

  Port: [input] CAN port number (0~1 or 0~3)

- **Return:**

  DNM_NoError: OK

  DNM_BoardNumberError: BoardNo exceeds the current total board number.

  DNM_PortNumberError: Port exceeds the current port number.

  DNM_BoardNotActive: BoardNo has not been activated.

  DNM_MasterStateError: This CAN port is not online.

  DNM_OfflineError: The CAN port can not transfer into offline state.

4.4.6 DNM_ChangeMACID

- **Description:**

  When the CAN port is online, this function can change the MAC ID of this master port.

- **Syntax:**

  WORD DNM_ChangeMACID(BYTE BoardNo , BYTE Port , BYTE Macid)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  Macid: [input] DeviceNet Master MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_MACIDNumberError: Macid exceeds the range from 0 to 63

4.4.7 DNM_ChangeBaudrate

- **Description:**

  When the CAN port is online, this function can change its Baud rate.

- **Syntax:**

```
WORD DNM_ChangeBaudrate (BYTE BoardNo , BYTE Port ,
                                    BYTE Baudrate)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  Macid: [input] DeviceNet Master MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_BaudrateError: Baud rate exceeds the range from 0 to 2
  DNM_ChangeBaudrateError: Check the DNM_Online errors

4.4.8 DNM_AddDevice

- **Description:**

  This function can add the slave devices into the scan list of the DeviceNet master port. Before using this function, the CAN 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:**

```
WORD DNM_AddDevice (BYTE BoardNo , BYTE Port ,
                                          BYTE DestMACID)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  DestMACID: [input] Destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceAlreadyInScanList: The DestMACID device has been added before.
  DNM_MACIDNumberError: DestMACID exceeds the range from 0 to 63.Maybe it is the same with the Master MACID.

4.4.9 DNM_RemoveDevice

- **Description:**

  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:**

```
WORD DNM_RemoveDevice (BYTE BoardNo , BYTE Port ,
                                    BYTE DesMACID)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  DestMACID: [input] Destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.

4.4.10 DNM_ConfigExplicitMsg

- **Description:**

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

- **Syntax:**

```
WORD DNM_ConfigExplicitMsg(BYTE BoardNo , BYTE Port ,
                           BYTE DesMACID ,
                           WORD watchdog_timeout_action)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)

Port: [input] CAN port number (CAN200): 0~1 ; (CAN400): 0~3

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 will wait until its child connections have been delete. If the connection has no child, it will be deleted.

- **Return:**

DNM_NoError: OK

DNM_BoardNumberError: BoardNo exceeds the current total board number.

DNM_PortNumberError: Port exceeds the current port number.

DNM_BoardNotActive: BoardNo has not been activated.

DNM_MasterStateError: This CAN port is not online.

DNM_DeviceNotInScanList: The DestMACID device has been removed before.

4.4.11 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 firstly.

- **Syntax:**

```
WORD DNM_SetAttribute(BYTE BoardNo,BYTE Port,BYTE DesMACID,
      WORD ClassID, BYTE InstanceID, BYTE AttributeID,
      BYTE *SetValue, UINT Length)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The device has been removed before.
DNM_SetAttributeDataIsEmpty: The SetValue is empty. Or the Length is 0.
DNM_DeviceIsUnConnected: The Explicit Msg Connection doesn't establish.
DNM_SetAttributeResponseError: The device response an error message.

DNM_SetAttributeResponseTimeout: The device doesn’t response in 1 sec.

DNM_SetAttributeFragmentACKTimeout: The device doesn’t ACK in 1 sec.

DNM_SetAttributeACKError: The device has an ACK error response.

- **Example([VC++] [BCB])**

```
BYTE  *Data = new BYTE[64];
Data[0] = 0xC4;   Data[1] = 0x09;
DNM_SetAttribute(0,0,1,5,1,9,Data,2);
```

4.4.12 DNM_GetAttribute

- **Description:**

  This function is used to get the attribute value of the specific device's instance. Before calling this function, the Explicit Message Connection, which connects to the MAC ID device, must be established first.

- **Syntax:**

```
WORD DNM_GetAttribute(BYTE BoardNo,BYTE Port,BYTE DesMACID,
         WORD ClassID,BYTE InstanceID,BYTE AttributeID,
         BYTE *GetValue , UINT *Length)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The device has been removed before.
DNM_DeviceIsUnConnected: The Explicit Msg Connection doesn't establish.
DNM_GetAttributeResponseError: The device response an error message.
DNM_GetAttributeResponseTimeout: The device has no response in 1 sec.

- **Example([VC++] [BCB])**

```
BYTE  *Data = new BYTE[64];
UINT  Length;
DNM_SetAttribute(0,0,1,5,1,9,Data,&Length);
```

4.4.13 DNM_CheckExplicitMsgConnectionStatus

- **Description:**

  The function is used to check the Explicit Message Connection Status, which connects to the specific slave.

- **Syntax:**

```
WORD DNM_CheckExplicitMsgConnectionStatus(BYTE BoardNo ,
                                          BYTE Port ,
                                          BYTE DesMACID)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The device has been removed before.
  DNM_DeviceIsUnConnected: The Explicit Msg Connection doesn’t establish.
  DNM_GetAttributeResponseError: The device response an error message.
  DNM_GetAttributeResponseTimeout: The device has no response in 1 sec.

4.4.14 DNM_ConfigPoll

- **Description:**

  This method is used to configure the Poll connection of the specific MAC ID device. The master port can get/set the data via the connection, which connects to the specific slave, 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:**

```
WORD DNM_ConfigPoll(BYTE BoardNo , BYTE Port ,
                    BYTE DesMACID ,
                    WORD produced_connection_size,
                    WORD consumed_connection_size,
                    WORD expected_packet_rate,
                    WORD watchdog_timeout_action = 0)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK

DNM_BoardNumberError: BoardNo exceeds the current total board number.

DNM_PortNumberError: Port exceeds the current port number.

DNM_BoardNotActive: BoardNo has not been activated.

DNM_MasterStateError: This CAN port is not online.

DNM_DeviceNotInScanList: The device has been removed before.

DNM_COS_CyclicAlreadyConfigure: The COS/Cyclic is configured before.

4.4.15 DNM_CheckPollConnectionStatus

- **Description:**

The function is used to check the Poll connection status, which connects to the specific MAC ID device.

- **Syntax:**

WORD DNM_CheckPollConnectionStatus(BYTE BoardNo,BYTE Port,
BYTE DesMACID)

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

DNM_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The Poll connection doesn't establish.

4.4.16 DNM_ReadPollInputData

- **Description:**

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

- **Syntax:**

```
WORD DNM_ReadPollInputData(BYTE BoardNo,BYTE Port,
                           BYTE DesMACID, BYTE *DataBuf,
                           UINT *DataLength)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  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_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.
  DNM_DeviceIsUnConnected: The Poll connection doesn’t establish.

4.4.17 DNM_WritePollOutputData

- **Description:**

  The function will set the data according with the consumed connection path of the specific MAC ID device via the Poll connection.

- **Syntax:**

```
WORD DNM_WritePollOutputData(BYTE BoardNo,BYTE Port,
                             BYTE DesMACID , BYTE *DataBuf ,
                             UINT DataLength)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The Poll connection doesn’t establish.
DNM_WritePollOutputDataLengthError: The Output Data Length is wrong.

4.4.18 DNM_ConfigBitStrobe

- **Description:**

  The function is to configure the Bit-Strobe connection of the specific MAC ID device. The master port will get the data of the MAC ID 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:**

```
WORD DNM_ConfigBitStrobe(BYTE BoardNo,BYTE Port,
                         BYTE DesMACID ,
                         WORD produced_connection_size,
                         WORD expected_packet_rate,
                         WORD watchdog_timeout_action = 0)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  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_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.

4.4.19 DNM_ReadBitStrobe

- **Description:**

The function is used to get the data according with the produced connection path of the MAC ID slave device via the Bit-strobe connection.

- **Syntax:**

```
WORD DNM_ReadBitStrobe(BYTE BoardNo , BYTE Port ,
                       BYTE DesMACID , BYTE *DataBuf ,
                       UINT *DataLength)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The Bit-Strobe connection doesn’t establish.

4.4.20 DNM_CheckBitStrobeConnectionStatus

- **Description:**

  Check the status of the Bit-Strobe connection, which is connected to the specific MAC ID device.

- **Syntax:**

```
WORD DNM_CheckBitStrobeConnectionStatus(BYTE BoardNo ,
                                        BYTE Port ,
                                        BYTE DesMACID)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.
  DNM_DeviceIsUnConnected: The Bit-Strobe connection doesn't establish.

4.4.21 DNM_ConfigCOS

- **Description:**

  This function is used to configure the Change_Of_State connection. This connection will get the data every period of time, when the poll connection 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:**

```
WORD DNM_ConfigCOS(BYTE BoardNo , BYTE Port ,
                   BYTE DesMACID ,
                    WORD produced_connection_size ,
                    WORD consumed_connection_size ,
                    BYTE isnonack = FALSE,
                    WORD expected_packet_rate = 0x9C4,
                    WORD watchdog_timeout_action = 0)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  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_NoError: OK

DNM_BoardNumberError: BoardNo exceeds the current total board number.

DNM_PortNumberError: Port exceeds the current port number.

DNM_BoardNotActive: BoardNo has not been activated.

DNM_MasterStateError: This CAN port is not online.

DNM_DeviceNotInScanList: The device has been removed before.

4.4.22 DNM_ConfigCyclic

- **Description:**

This function is used to configure the Cyclic connection. This connection will get the data according with the produced connection path of the MAC ID device channel every period of time, when the poll connection 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:**

```
WORD DNM_ConfigCyclic(BYTE BoardNo , BYTE Port ,
                                BYTE DesMACID ,
                                 WORD produced_connection_size ,
                                 WORD consumed_connection_size ,
                                 BYTE isnonack = FALSE,
                                 WORD expected_packet_rate = 0x9C4,
                                 WORD watchdog_timeout_action = 0)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK

DNM_BoardNumberError: BoardNo exceeds the current total board number.

DNM_PortNumberError: Port exceeds the current port number.

DNM_BoardNotActive: BoardNo has not been activated.

DNM_MasterStateError: This CAN port is not online.

DNM_DeviceNotInScanList: The device has been removed before.

4.4.23 DNM_ReadCOSInputData

- **Description:**

This method is used to read the data of the MAC ID slave device via the COS connection.

- **Syntax:**

```
WORD DNM_ReadCOSInputData(BYTE BoardNo,BYTE Port,
                          BYTE DesMACID , BYTE *DataBuf ,
                          UINT *DataLength)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The COS connection doesn’t establish.

4.4.24 DNM_WriteCOSOutputData

- **Description:**

  This function is used to set the data according with the consumed connection path of the MAC ID slave device via the COS connection.

- **Syntax:**

```
WORD DNM_WriteCOSOutputData(BYTE BoardNo,BYTE Port,
                            BYTE DesMACID , BYTE *DataBuf ,
                            UINT DataLength)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  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_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.
  DNM_DeviceIsUnConnected: The COS connection doesn't establish.
  DNM_WriteCOSOutputDataLengthError: The Output Data length is wrong.

4.4.25 DNM_ReadCyclicInputData

- **Description:**

This function is used to get the data according with the produced connection path of the MAC ID slave device via the cyclic connection.

- **Syntax:**

```
WORD DNM_ReadCyclicInputData(BYTE BoardNo,BYTE Port,
                             BYTE DesMACID , BYTE *DataBuf ,
                             UINT *DataLength)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The Cyclic connection doesn’t establish.

4.4.26 DNM_WriteCyclicOutputData

- **Description:**

This method is used to set the data according with the consumed connection path of the output channel of the device.

- **Syntax:**

```
WORD DNM_WriteCyclicOutputData(BYTE BoardNo,BYTE Port,
                               BYTE DesMACID,BYTE *DataBuf,
                               UINT DataLength)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The Cyclic connection doesn’t establish.
DNM_WriteCyclicOutputDataLengthError: The Output Data length is wrong.

4.4.27 DNM_CheckCOSConnectionStatus

- **Description:**

  This function is used to check the status of the COS connection which is connected to the MAC ID slave device.

- **Syntax:**

```
WORD DNM_CheckCOSConnectionStatus(BYTE BoardNo ,BYTE Port ,
                                  BYTE DesMACID)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)

  Port: [input] CAN port number (0~1 or 0~3)

  DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK

  DNM_BoardNumberError: BoardNo exceeds the current total board number.

  DNM_PortNumberError: Port exceeds the current port number.

  DNM_BoardNotActive: BoardNo has not been activated.

  DNM_MasterStateError: This CAN port is not online.

  DNM_DeviceNotInScanList: The DestMACID device has been removed before.

  DNM_DeviceIsUnConnected: The COS connection doesn't establish.

4.4.28 DNM_CheckCyclicConnectionStatus

- **Description:**

This function is used to check the status of the Cyclic connection which is connected to the MAC ID slave device.

- **Syntax:**

```
WORD DNM_CheckCyclicConnectionStatus(BYTE BoardNo,BYTE Port,
                                     BYTE DesMACID)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

DNM_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_DeviceNotInScanList: The DestMACID device has been removed before.
DNM_DeviceIsUnConnected: The Cyclic connection doesn't establish.

4.4.29 DNM_StartDevice

- **Description:**

  This function is used to start to communicate with the destination device with MAC ID.

- **Syntax:**

  WORD DNM_StartDevice(BYTE BoardNo,BYTE Port,BYTE DesMACID)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.
  DNM_DeviceIsConnecting: The device is connecting now.

4.4.30 DNM_StopDevice

- **Description:**

  This function is used to stop to communicate with the destination device.

- **Syntax:**

  WORD DNM_StopDevice(BYTE BoardNo,BYTE Port,BYTE DesMACID)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  DestMACID: [input] The destination DeviceNet MAC ID (0~63)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_DeviceNotInScanList: The DestMACID device has been removed before.
  DNM_DeviceIsUnConnected: The device is Unconnected now.

4.4.31 DNM_StartCommunicate

- **Description:**

  This function is used to start to communicate with all devices in scan list.

- **Syntax:**

  WORD DNM_StartCommunicate(BYTE BoardNo , BYTE Port)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.

4.4.32 DNM_StopCommunicate

- **Description:**

  This function is used to stop to communicate with all destination devices in scan list.

- **Syntax:**

  WORD DNM_StopCommunicate(BYTE BoardNo , BYTE Port)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.

4.4.33 DNM_GetAllDeviceMACID

- **Description:**

  This function is to get the MAC IDs of all devices in DeviceNet network.

- **Syntax:**

```
WORD DNM_GetAllDeviceMACID(BYTE BoardNo,BYTE Port,
                           BYTE *AllMACID,BYTE *DeviceCount)
```

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)
  AllMACID: [output] All of the destination DeviceNet MAC IDs
  DeviceCount: [output] The amount of the devices in the network.

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.

4.4.34 DNM_GetOwnership

- **Description:**

  This function is used to make the master port to get the Ownership in DeviceNet network.

- **Syntax:**

  WORD DNM_GetOwnership(BYTE BoardNo , BYTE Port)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)

- **Return:**

  DNM_NoError: OK
  DNM_BoardNumberError: BoardNo exceeds the current total board number.
  DNM_PortNumberError: Port exceeds the current port number.
  DNM_BoardNotActive: BoardNo has not been activated.
  DNM_MasterStateError: This CAN port is not online.
  DNM_GetOwnershipFail: This port can not get the offline ownership

4.4.35 DNM_CheckFaultNode

- **Description:**

  This function is used to check any fault nodes in DeviceNet network.

- **Syntax:**

  WORD DNM_CheckFaultNode(BYTE BoardNo , BYTE Port)

- **Parameter:**

  BoardNo: [input] PISO-CAN400/200 board number (0~7)
  Port: [input] CAN port number (0~1 or 0~3)

- **Return:**

  This function will return the amount of the fault node (0~63). The maximum number could not larger than 63.

  DNM_BoardNumberError: BoardNo exceeds the current total board number.

  DNM_PortNumberError: Port exceeds the current port number.

  DNM_BoardNotActive: BoardNo has not been activated.

  DNM_MasterStateError: This CAN port is not online.

  DNM_MasterNotGetOwnership: This port does not get the offline ownership

4.4.36 DNM_GetAllFaultNode

- **Description:**

This function is used to get the Serial No and the Vendor ID of all fault nodes in DeviceNet network.

- **Syntax:**

```
WORD DNM_GetAllFaultNode(BYTE BoardNo , BYTE Port ,
                         WORD *VendorID , DWORD *SerialNo ,
                         BYTE *NodeCount)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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:**

The value 3 : NoResponse
DNM_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_MasterNotGetOwnership: This port does not get the offline ownership

4.4.37 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:**

```
WORD DNM_ChangeFaultMACID(BYTE BoardNo , BYTE Port ,
                          WORD VendorID,DWORD SerialNo,
                          BYTE NewMACID)
```

- **Parameter:**

BoardNo: [input] PISO-CAN400/200 board number (0~7)
Port: [input] CAN port number (0~1 or 0~3)
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_NoError: OK
DNM_BoardNumberError: BoardNo exceeds the current total board number.
DNM_PortNumberError: Port exceeds the current port number.
DNM_BoardNotActive: BoardNo has not been activated.
DNM_MasterStateError: This CAN port is not online.
DNM_MasterNotGetOwnership: This port does not get the offline ownership

5.Demo Programs for Windows

All of demo programs will not work normally if PISO-CAN200/400 driver would not be installed correctly. During the installation process of the driver, the install-shields will register the correct kernel driver to the operation system and copy the DLL driver and demo programs to the correct position based on the driver software package you have selected (Win98,Me,NT,win2000,XP). After completing the driver and DeviceNet library installation, the related demo programs, development library and declaration header files for different development environments are installed in the system as follows.

```
|--\Demo                              →demo program
   |--\BCB6                           → for Borland C++ Builder 6
   |    |--\DNM.H                     → Header file
   |       \BCBPISOCANDNM.LIB         → Linkage library for BCB
   |
   |--\VB6                            → for Visual Basic 6
      |--\PISOCANDNM.BAS              →Declaration file
```

The list of demo programs:

DEMO1: Devices communicate by the Explicit Message Connection.

DEMO2: Devices communicate by the I/O Connection.

A brief introduction to the demo programs

DEMO1:

Demo1 is the example used for starting the DeviceNet communication. This demo program is designed to communicate with slave device through Explicit Message Connection. Users apply the GetAttribute and SetAttribute function to access the information data of the slave devices. Before exercising this demo, the user needs to finish the wire connection between the Master and slave device. (See figure 5.1)

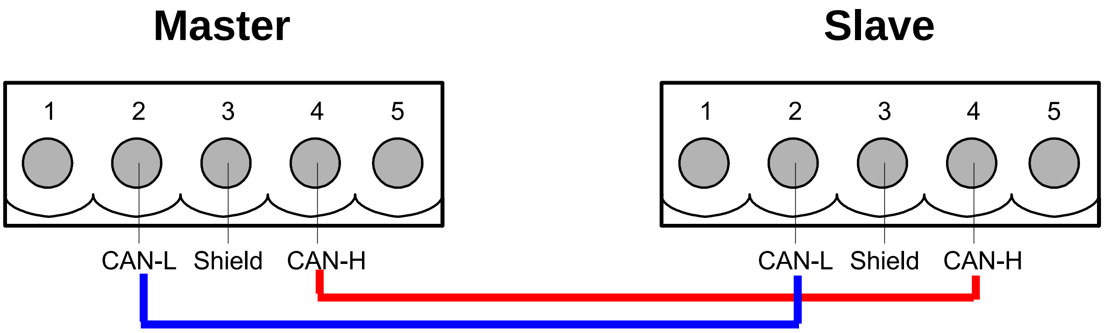


Figure 5.1: The wire connection

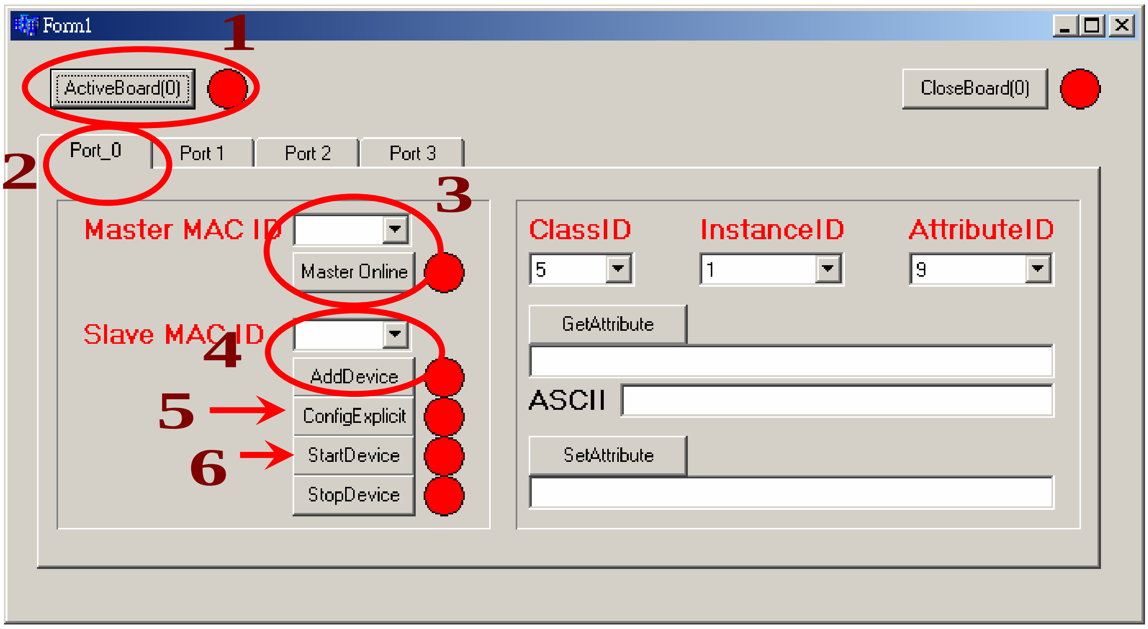


Figure 5.2: the form of demo1

Based on this demo (See figure 5.2), the users can get the attribute value of the slave device. And users can set the attribute value of the slave device by keying the DeviceNet message into the SetAttribute frame area and then click the “SetAttribute” button to send out the DeviceNet message to the slave device through the DeviceNet network. For example, you can select ClassID = 1, InstanceID = 1, AttributeID = 7, then press “GetAttribute” button; you will get the slave product name in the ASCII field.

DEMO2:

In demo 2, we provide a demonstration on how to communicate with slave device through the I/O Connection. The users should check the IO connection mode that the slave device supports, then set the input and output data length. Pressing the “Config XXX Connection” button to finish the configuration. The next step is to press “start device” button to start to communicate with slave device via the I/O connection. If the I/O connection is established, the user can read the data from the slave device by pressing “Read input data” button. At the same time, the user can write the data to the slave device.

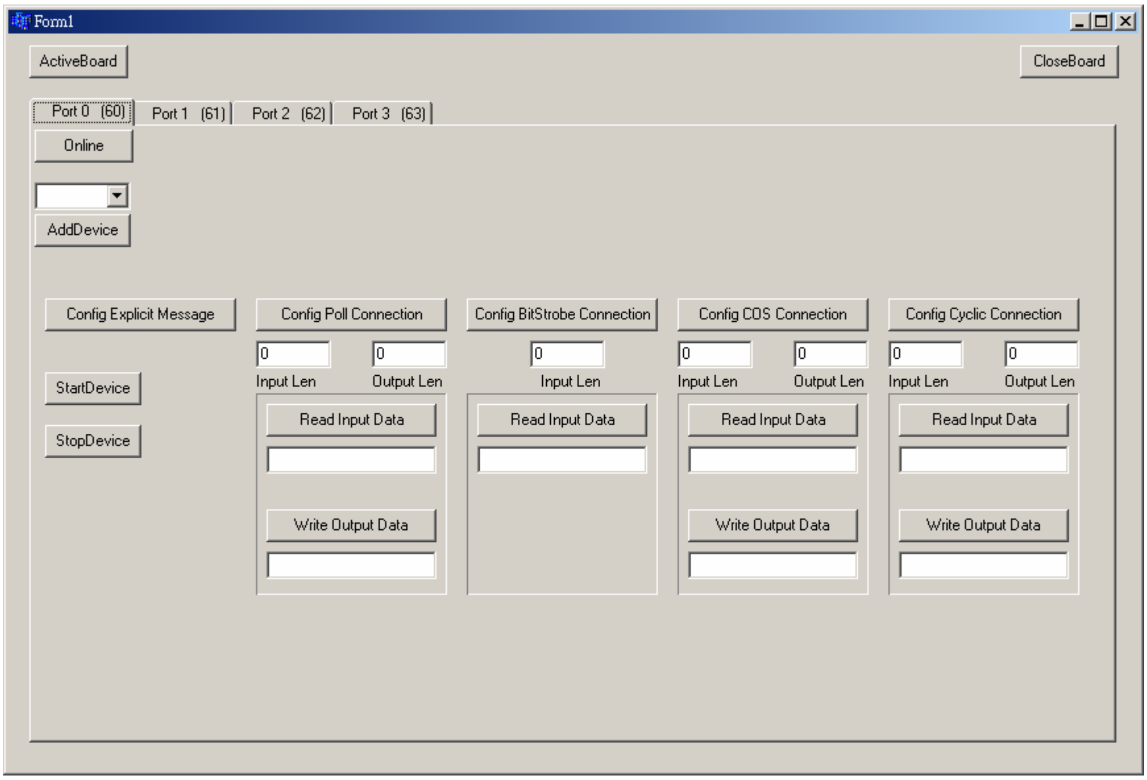


Figure 5.3: The form of demo2 program