# PISO-DNM100-D/T
# PISO-DNM100U-D/T
# DeviceNet Master PCI Board
## User's Manual



**PISO-DNM100U-T**



**PISO-DNM100U-D**

## 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

Copyright 2009 by ICP DAS Co., LTD. All rights reserved worldwide.

## Trademark

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# Revision

| Version | Firmware Version | Date | Author | Description |
|---|---|---|---|---|
| 3.4 | 2.6 | 2019/ 01/10 | Johney | 1. Add Venders' DeviceNet Slave.<br>2. Update Windows version. |
| 3.3 | 2.5 | 2015/ 12/24 | Johney | 1. Add Venders' DeviceNet Slave.<br>2. Update pin assignment. |
| 3.2 | 2.5 | 2013/ 06/06 | Johney | Add new function<br>1. DNM100_GetAttributeW<br>2. DNM100_SetAttributeW<br>3. DNM100_SendExplicitMSG_W |
| 3.1 | 2.4 | 2011 01/10 | Johney | Add new functions<br>* DNM100_ReadInputArea<br>* DNM100_WriteOutputArea<br>* DNM100_ReadbackOutputArea |
| 3.0 | 2.2 | 2009/ 12/25 | Johney | Add the information of the PISO-DNM100U board. |
| 2.02 | 2.1 | 2009 07/14 | Johney | Add the function DNM100_ReadbackOutputData |
| 2.01 | 2.0 | 2009 01/16 | Johney | Correct the function name. GW7304_xxx --> DNM100_xxx |
| 2.0 | 2.0 | 2008 12/15 | Johney | Reduce the number of the API function from 66 to 36. The new firmware is more efficient. |
| 1.0 | 1.0 | 2008 02/15 | Johney | This manual is for the PISO-DNM100 board. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# Contents

**REVISION** ..... **2**

**1. GENERAL INFORMATION** ..... **6**

*1.1 DeviceNet Introduction* ..... 6
*1.2 DeviceNet Applications* ..... 8
*1.3 PISO-DNM100(U) with Vendor's DeviceNet Slaves* ..... 9
*1.4 PISO-DNM100(U) Architecture* ..... 11
*1.5 DeviceNet Master Characteristics* ..... 12
*1.6 PISO-DNM100(U) Firmware Characteristics* ..... 15
*1.7 Features* ..... 17
*1.8 Specifications* ..... 18
*1.9 Block Diagram* ..... 19
*1.10 Product Check List* ..... 20

**2. HARDWARE CONFIGURATION** ..... **21**

*2.1 Board Layout* ..... 21
*2.2 Jumper Selection* ..... 22
*2.3 Connector Pin Assignment* ..... 24
*2.3.1 5-pin screw terminal connector* ..... 24
*2.3.2 9-pin D-sub male connector* ..... 25
*2.3.3 Wire connection* ..... 26
*2.4 Indicator LED* ..... 27
*2.4.1 Green LED* ..... 27
*2.4.2 Red LED* ..... 27
*2.5 Hardware Installation* ..... 28

**3. DRIVER INSTALLATION AND SOFTWARE APPLICATION** ..... **29**

*3.1 Driver Installation of the PISO-DNM100(U)* ..... 30
*3.2 Flow Diagram for Searching Devices* ..... 35
*3.3 Flow Diagram for Slave Configuration* ..... 36
*3.4 Flow Diagram for On-line Adding/Removing Device* ..... 37
*3.5 Flow Diagram for "SetAttribute" and "GetAttribute"* ..... 39
*3.6 Flow Diagram for I/O Connection* ..... 40
*3.7 Input and Output I/O Data Area (Advanced Option)* ..... 41

**4. FUNCTION DESCRIPTION** ..... **44**

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

*4.1 DLL FUNCTION DEFINITION AND DESCRIPTION* ..... 45
*4.2 FUNCTION RETURN CODE* ..... 49
*4.3 FUNCTION DESCRIPTION* ..... 53
4.3.1 DNM100_GetBoardInfo ..... 53
4.3.2 DNM100_TotalDNM100Board ..... 54
4.3.3 DNM100_ActiveBoard ..... 55
4.3.4 DNM100_CloseBoard ..... 56
4.3.5 DNM100_GetDLLVersion ..... 57
4.3.6 DNM100_GetFirmwareVersion ..... 58
4.3.7 DNM100_ResetFirmware ..... 59
4.3.8 DNM100_GetMasterMACID ..... 60
4.3.9 DNM100_SetMasterMACID ..... 61
4.3.10 DNM100_GetBaudRate ..... 62
4.3.11 DNM100_SetBaudRate ..... 63
4.3.12 DNM100_GetMasterStatus ..... 64
4.3.13 DNM100_GetSlaveStatus ..... 65
4.3.14 DNM100_StartDevice ..... 66
4.3.15 DNM100_StopDevice ..... 67
4.3.16 DNM100_StartAllDevice ..... 68
4.3.17 DNM100_StopAllDevice ..... 69
4.3.18 DNM100_AddDevice ..... 70
4.3.19 DNM100_RemoveDevice ..... 71
4.3.20 DNM100_AddIOConnection ..... 72
4.3.21 DNM100_RemoveIOConnection ..... 73
4.3.22 DNM100_GetAttribute ..... 74
4.3.23 DNM100_GetAttributeW ..... 75
4.3.24 DNM100_IsGetAttributeOK ..... 76
4.3.25 DNM100_GetAttributeValue ..... 77
4.3.26 DNM100_SetAttribute ..... 78
4.3.27 DNM100_SetAttributeW ..... 79
4.3.28 DNM100_IsSetAttributeOK ..... 80
4.3.29 DNM100_ClearAllConfig ..... 81
4.3.30 DNM100_SearchAllDevices ..... 82
4.3.31 DNM100_SearchSpecificDevice ..... 83
4.3.32 DNM100_IsSearchOK ..... 84
4.3.33 DNM100_GetSearchedDevices ..... 85
4.3.34 DNM100_GetDeviceInfoFromScanList ..... 86
4.3.35 DNM100_GetScanList ..... 87

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

4.3.36 DNM100_ImportEEPROM ..... 89
4.3.37 DNM100_ReadInputData ..... 90
4.3.38 DNM100_WriteOutputData ..... 91
4.3.39 DNM100_SendExplicitMSG ..... 92
4.3.40 DNM100_SendExplicitMSG_W ..... 93
4.3.41 DNM100_IsExplicitMSGRespOK ..... 94
4.3.42 DNM100_GetExplicitMSGRespValue ..... 95
4.3.43 DNM100_ReadbackOutputData ..... 96
4.3.44 DNM100_ReadInputArea (Advanced Option) ..... 97
4.3.45 DNM100_WriteOutputArea (Advanced Option) ..... 99
4.3.46 DNM100_ReadbackOutputArea (Advanced Option) ..... 101

***5. DEMO PROGRAMS FOR WINDOWS ..... 103***

*5.1 A BRIEF INTRODUCTION TO THE DEMO PROGRAMS ..... 103*
*5.2 WIRE CONNECTION OF THE CAN BUS ..... 104*
*5.3 VC_DEMO1 INTRODUCTION ..... 105*
*5.4 VC_DEMO2 INTRODUCTION ..... 108*
*5.5 BCB_DEMO1 INTRODUCTION ..... 110*
*5.6 BCB_DEMO2 INTRODUCTION ..... 112*

***6. LABVIEW DRIVER INTRODUCTION ..... 114***

*6.1 SOFTWARE INSTALLATION ..... 114*
*6.2 FUNCTION DESCRIPTION ..... 116*
*6.3 LABVIEW DEMO INTRODUCTION ..... 120*

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 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.

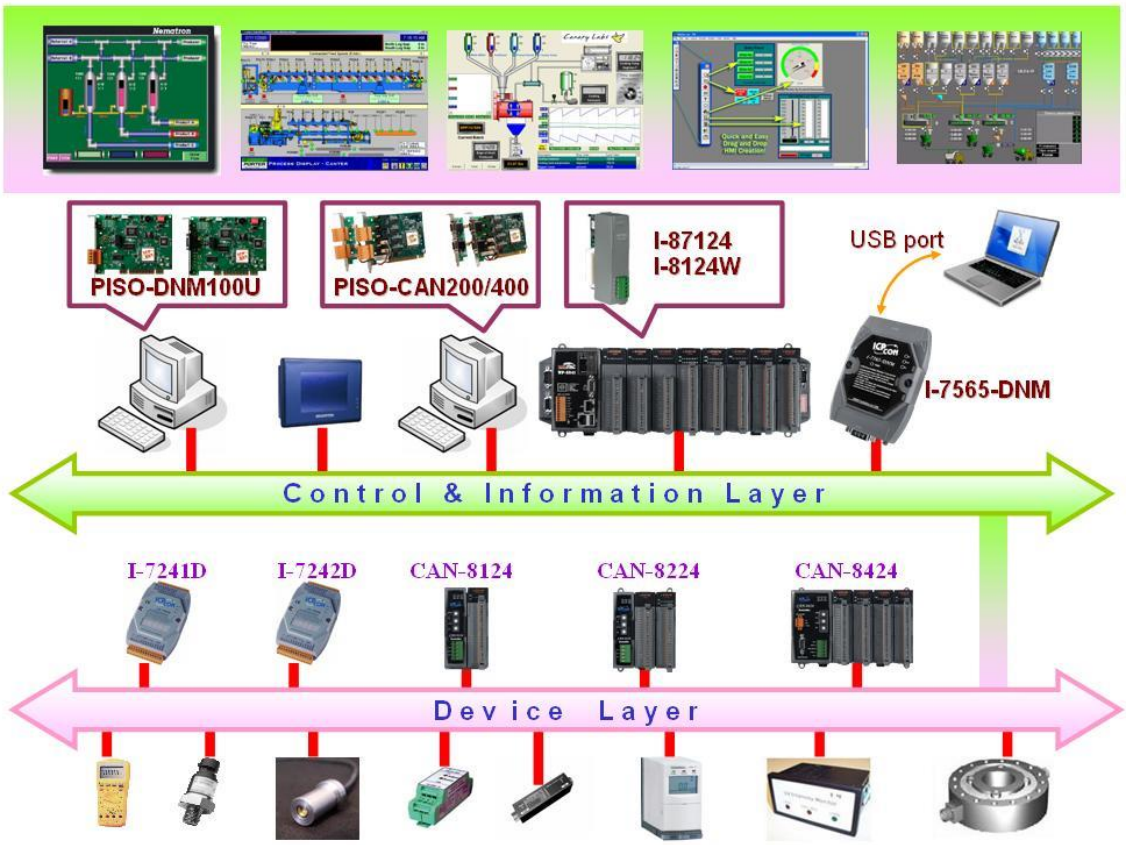



Figure 1.1 Example of the DeviceNet network

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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 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

PISO-DNM100(U) can represent an economic solution of DeviceNet application and be a DeviceNet master device on the DeviceNet network. PISO-DNM100(U) supports Group 2 only Server and UCMM functions to communication with slave devices. It has an independent CAN bus communication port with the ability to cover a wide range of DeviceNet applications. Besides, PISO-DNM100(U) uses the new CAN controller Phillips SJA1000T and transceiver 82C250, which provide bus arbitration, error detection with auto correction and re-transmission function. It can be installed on almost any windows-based system. It is popularly applied in the industrial automation, building automation, vehicle, marine, and embedded control network. Therefore, that is an easy way to develop the DeviceNet network with PISO-DNM100(U).

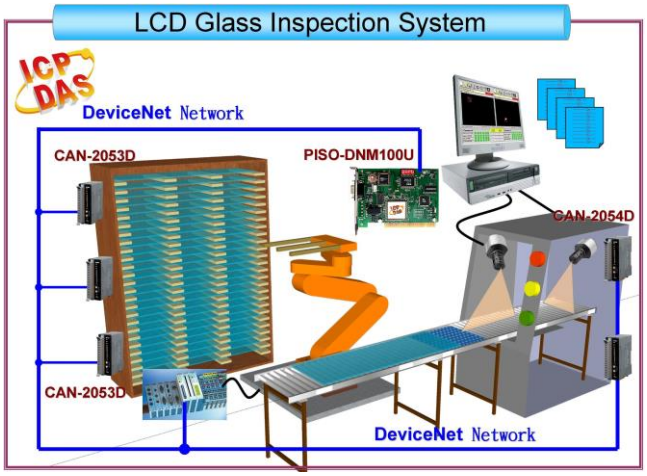
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

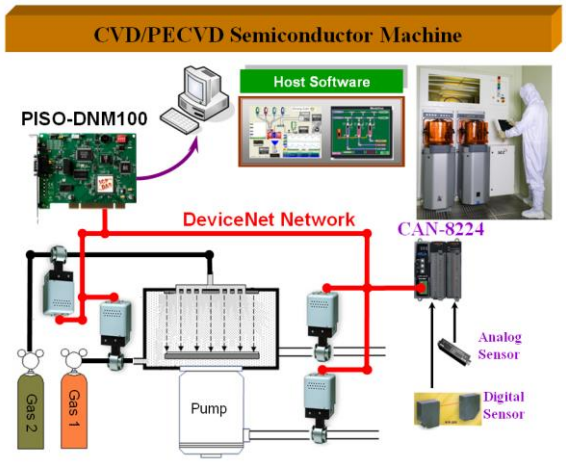
## 1.2 DeviceNet Applications

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 PISO-DNM100(U) with Vendor's DeviceNet Slaves

We have communicated with the following DeviceNet slaves.

- Allen-Bradley(AB) **PowerFlex** series DeviceNet Inverters.
- BECKHOFF **CX1500-B520** series DeviceNet I/O modules.
- BECKHOFF **BK5250** series DeviceNet I/O modules.
- MKS **683** series DeviceNet exhaust throttles.
- MKS **MFC (Mass Flow Controller)** series DeviceNet devices.
- MKS **DELTA-II FRC (Flow Ratio Controller)** series DeviceNet devices.
- MKS **DC Power Generator (OPT- xxx)** series DeviceNet devices.
- OMRON **DRT1-ID/ODxx** series DeviceNet I/O modules.
- OMRON **DRT2-MDxx** series DeviceNet I/O modules.
- COSMOS **PS-7** series DeviceNet gas detectors.
- CELERITY **UNIT IFC-125** series DeviceNet devices.
- Allen-Bradley(AB) **PowerFlex AC** Drives
- Allen-Bradley(AB) **PowerFlex AC** Drives with DriveLogix
- Allen-Bradley(AB) **PowerFlex DC** Drives
- OMRON **DRT2-ID08(-1)/MD16**
- OMRON **DRT2-OD08/MD16-1**
- OMRON **DRT2-OD08-1**
- OMRON **DRT2-ID16(-1)/OD16(-1)**
- OMRON **GRT1-DRT**
- OMRON **C200HW-DRT21**
- Swagelok **MS-VCM-D-6-0**, **MS-VCM-D-6-2** Digital DeviceNet Valve
- Swagelok **SS-PTX-D-G500-S4-K** Digital DeviceNet Pressure-Temperature Transducer
- Swagelok **SS-PTX-D-G500-SM-K** Digital DeviceNet Pressure-Temperature Transducer
- Weidmueller **SAI-AU M12 DN 16DI/8DO**

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

- Weidmueller **SAI-AU M12 DN GW 16DI**
- Weidmueller **SAI-AU M12 DN AI/AO/DI**
- ADVANCED ENERGY Apex **RF generators** and power-delivery
- SMC **ITV series** Electro-Pneumatic Regulator
- SMC **Directional Control Valves**
- SICK **DME500 series** Distance Sensor
- MTS **Temposonics R-Series** Position Sensors
- PFEIFFER VACUUM **HiPace series turbopumps**
- Allen-Bradley(AB) 1769-ADN DeviceNet Adapter
- MKS Type T3B and T3P Valves With DeviceNet Interface
- ABB DSQC 350A DeviceNet/Allen Bradley remote gateway
- MTS Temposonics R-Series sensor
- MKS MicroNode I/O series module
- LINCOLN Power Wave AC/DC 1000 device
- Festo CTEU-DN Bus nodes
- MKS Baratron Type DMB Capacitance Manometer
- Sevenstar CS200/CS220 Mass Flow Controller

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 1.4 PISO-DNM100(U) Architecture

The PISO-DNM100(U) provides users to establish DeviceNet network rapidly by Master/Slave connection model. The PISO-DNM100(U) is a high-performance DeviceNet master board with one CPU inside. This architecture of the PISO-DNM100(U) almost doesn't cost CPU resource and really increases the work efficiency on DeviceNet network. Applying the PISO-DNM100(U), users don't need to take care of the detail of the DeviceNet protocol. The inside firmware implements the DeviceNet protocol to help users to establish the connection with DeviceNet slave devices easily. The illustration about the idea is shown as Figure 1.2.

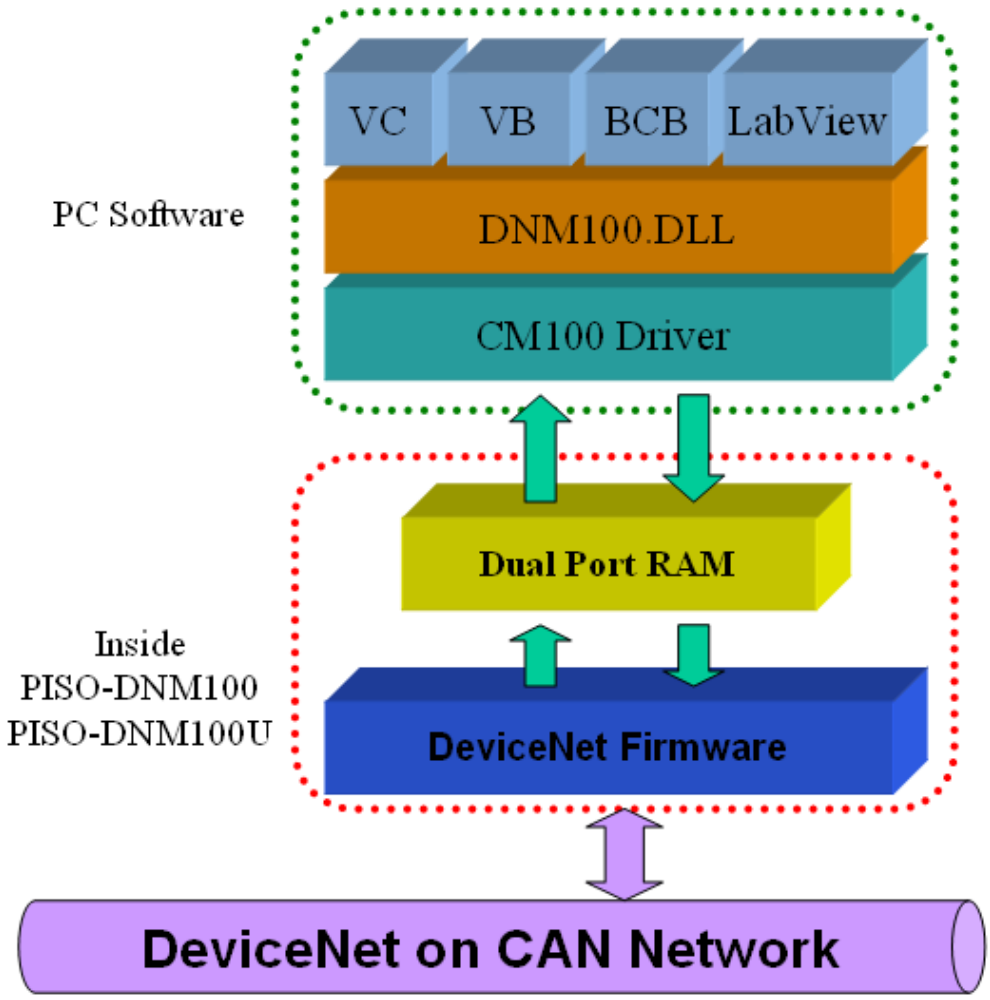



Figure 1.2 PISO-DNM100(U) illustration.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 1.5 DeviceNet Master Characteristics

Using the API functions, users don't need to take care of the detail of the DeviceNet protocol. It can reduce the complexity of user's DeviceNet Master Software. The firmware mainly supports the Predefined Master-Slave Connection Set and UCMM functions to allow users to merge third party's DeviceNet devices into the DeviceNet network. It can help users to establish the connection with DeviceNet slave devices easily. The general application architecture is demonstrated as Figure 1.3.

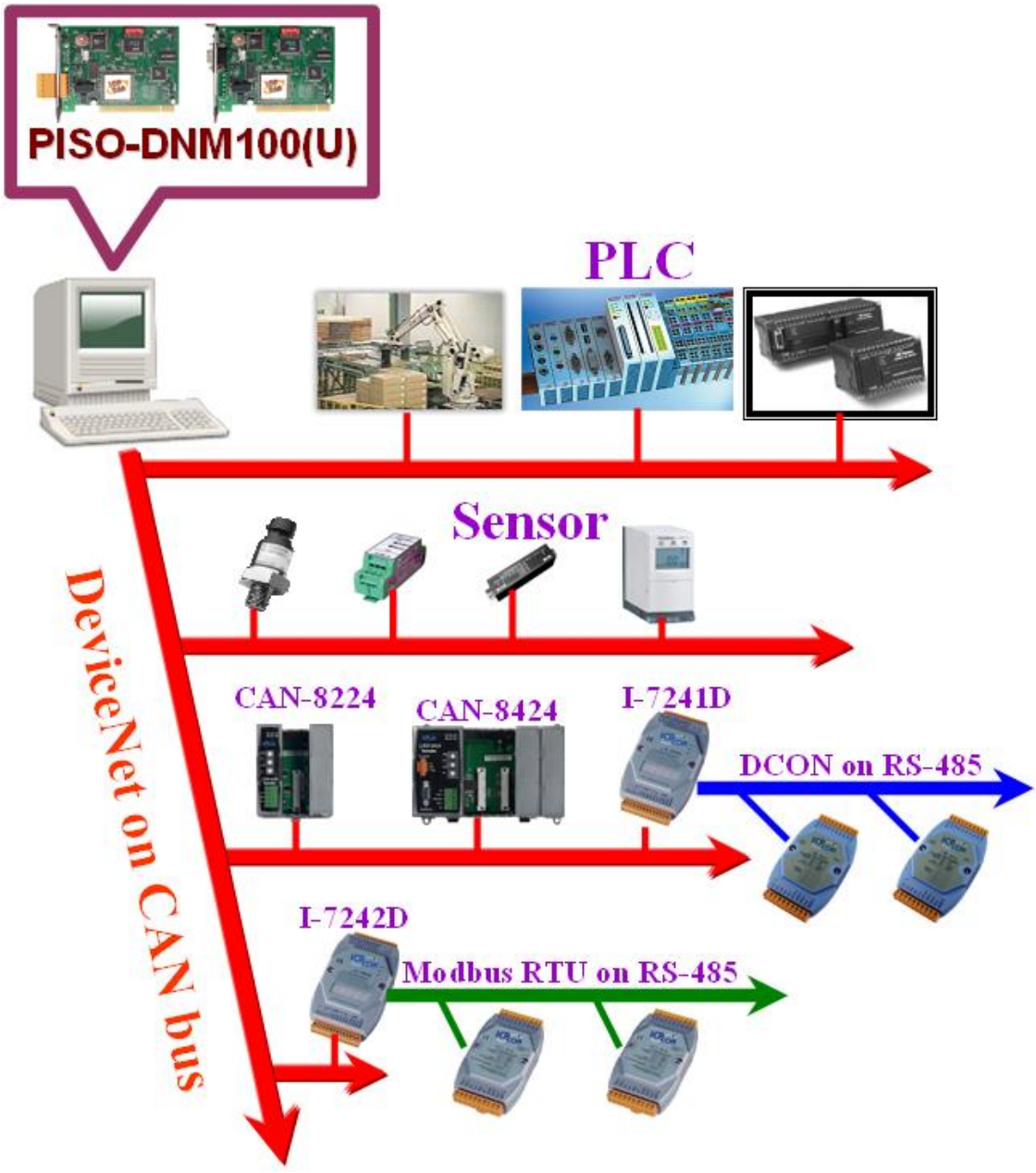



Figure 1.3 Application architecture

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

The DeviceNet protocol firmware provides the DeviceNet Master mechanism to communicate with slave devices by the Predefined Master/Slave Connection Set and UCMM Connection Set. In the DeviceNet communication protocol can be clarify as two forms: One is the Explicit Message and others are I/O Messages. Here, we only provide one explicit message connection and four I/O connections as depicted in Figure 1.4.

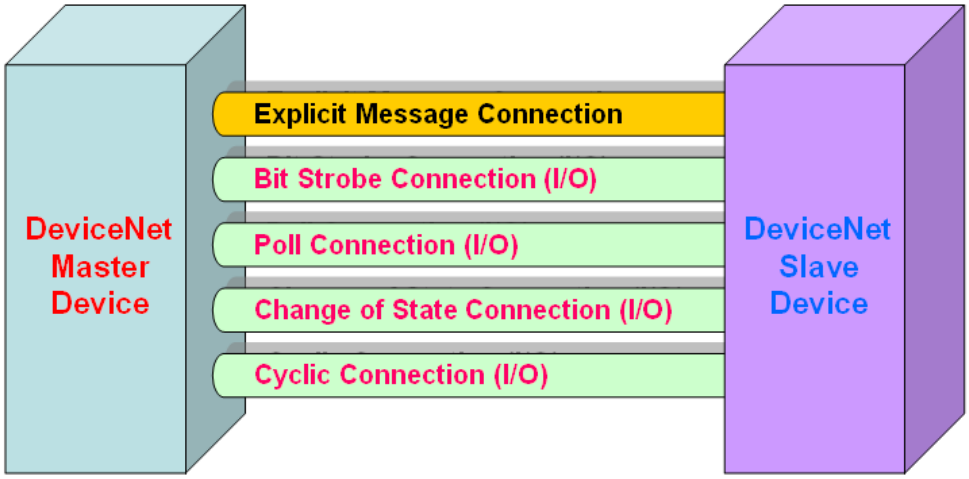



Figure 1.4 DeviceNet Messaging

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 the master control mechanism, there are only four main steps to be followed. Figure 1.5 demonstrates the basic process for the DeviceNet master communication. The every step function is described in below:

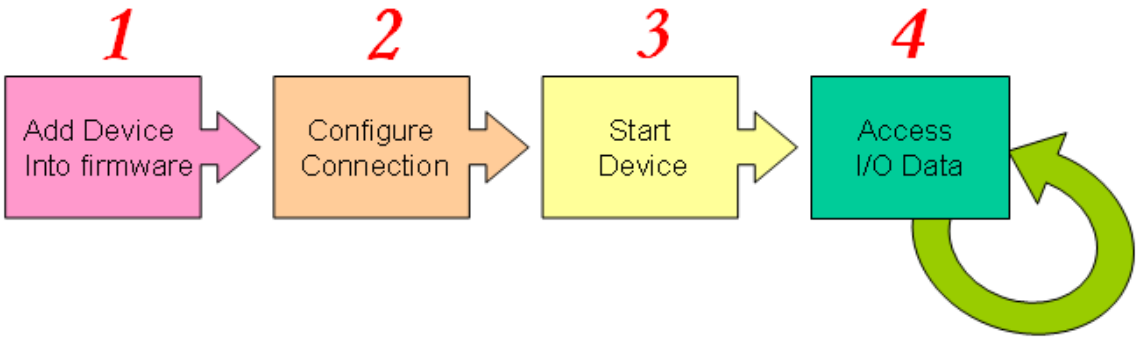



Figure 1.5 Four steps to establish connection

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

1. **Add device into firmware**

   You should provide the slave device's MAC ID to add into firmware by using API function.

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. **Start Device**

   After configuring connections, users should start device by using API function. The master will communicate with the slave device.

4. **Access I/O data**

   After communicating with slave devices, you can access the I/O data with corresponding read/write function.

After adding the device into the firmware, the master will wait for the I/O configuration information. Then users can create the I/O connections in the next step. Once I/O connections have been created and started, 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 API functions are 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 functions description and demo programs in section 4.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 1.6 PISO-DNM100(U) Firmware Characteristics

The PISO-DNM100(U) is a high-performance DeviceNet master board. The firmware inside the board implements DeviceNet protocol automatically when the board is active. The firmware always listens to the bus and receives the message at the same time. It works as shown in Figure 1.6.

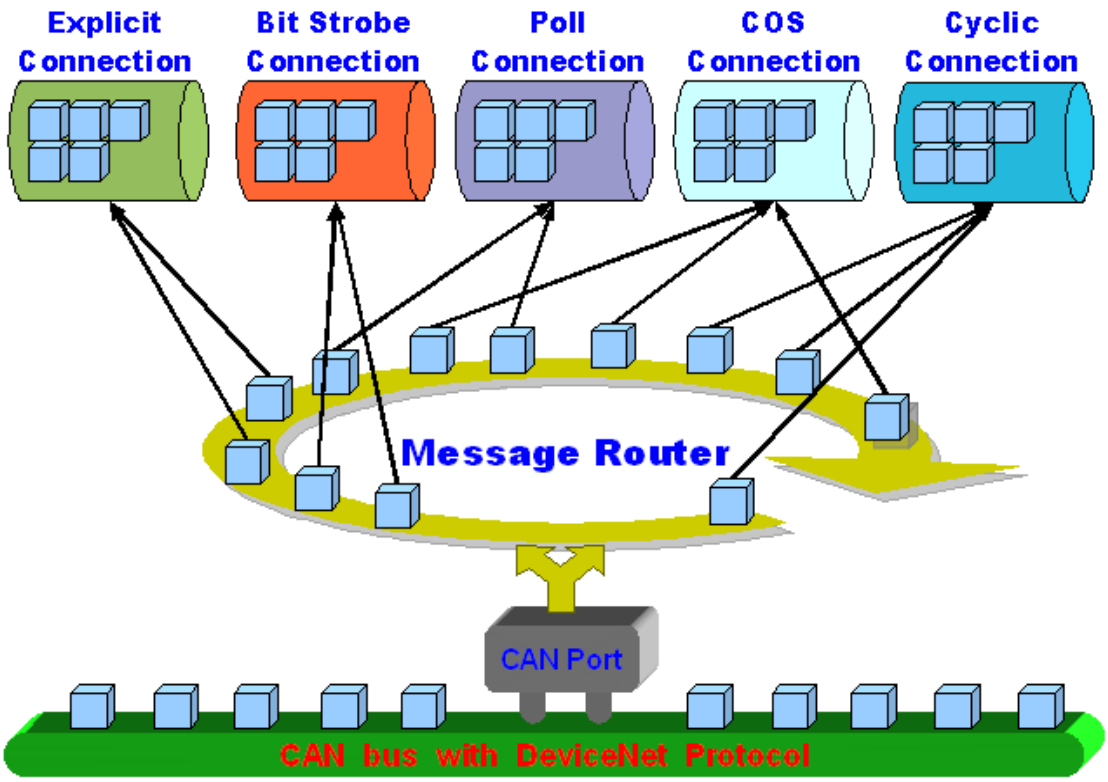



Figure 1.6 Message Router

The PISO-DNM100(U) firmware has a “ScanList” to store the remote slave devices information. After power off, the information still exists in the EEPROM. When the users turn on the PC next time, the “ScanList” will be loaded from EEPROM. The users can easily use the DLL functions to configure it, including adding devices or removing devices. It works as shown in Figure 1.7. There is more information about the library functions in chapter 4.

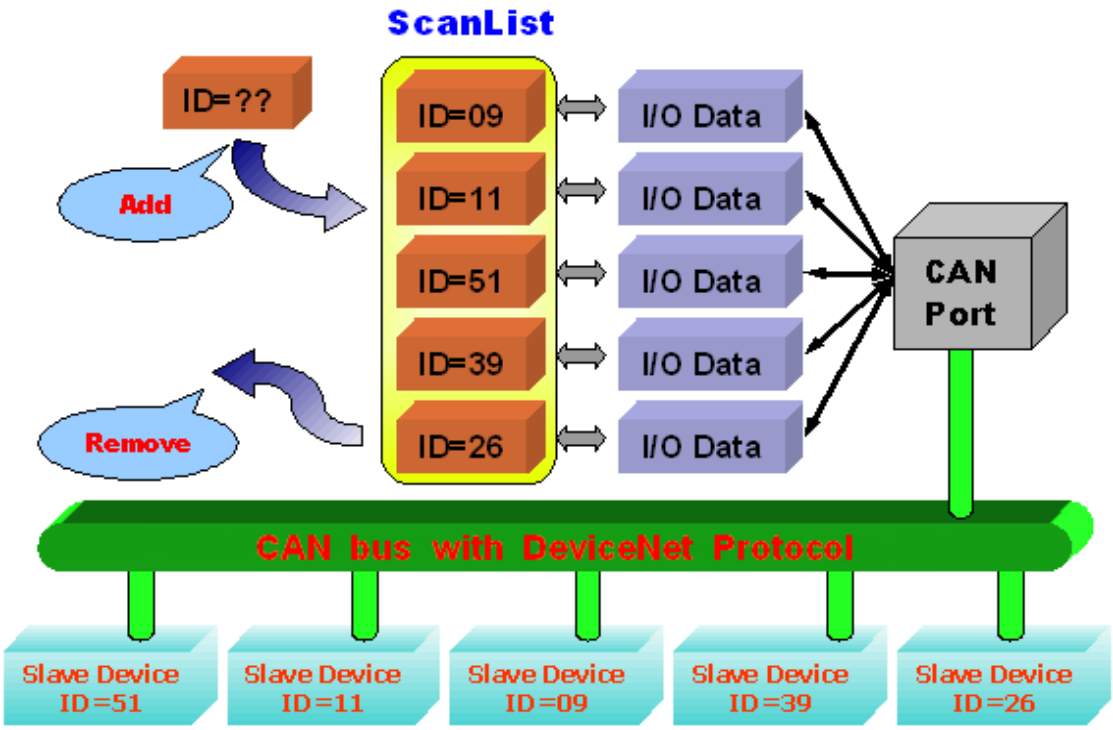



Figure 1.7 ScanList data structure

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 1.7 Features

### Hardware Features

- PISO-DNM100-D/T.
  - ∗ 33MHz 32bit 5V PCI bus (V2.1) plug and play technology.
- PISO-DNM100U-D/T
  - ∗ Universal PCI card, supports both 5V and 3.3V PCI bus.
- Driver supported for Windows XP/7/8.1/10.
- 186 compatible CPU with DeviceNet firmware inside.
- 8K bytes DPRAM inside.
- One CAN communication port.
- Compatible with CAN specification 2.0 parts A and B.
- Jumper select 120Ω terminator resistor for each port.
- 2 indicating LED (one for green and another for red).
- Direct memory mapping to the CAN controllers.
- 2500Vrms photo-isolation protection on CAN bus.
- 3000Vrms galvanic DC/DC isolation on CAN side.

### DeviceNet Firmware Features

- Programmable Master MAC ID.
- Programmable transfer-rate 125K, 250K, 500K.
- Each port support maximum nodes up to 64
- Support Group 2 Only Server functions
- Support UCMM functions
- Predefined Master-Slave 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
- Support Auto-Scan slave device function.
- Support on-line adding and removing devices.
- Support Auto-Reconnect when the connection is broken.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## *1.8 Specifications*

- CAN controller: Phillips SJA1000T.
- CAN transceiver: Phillips 82C250.
- Signal support: CAN_H, CAN_L.
- CAN controller frequency :16 MHz
- Connector: 5-pin screw terminal connector or 9-pin D-sub male connector.
- 2500Vrms photo-isolation protection on CAN bus.
- 3000Vrms galvanic DC/DC isolation on CAN side.
- 186 compactable CPU
- 8K bytes DPRAM (1K bytes for system)
- 512 K bytes Flash memory (128K bytes for system)
- 512K bytes SRAM
- RTC (real time clock) inside
- 2K EEPROM (256 bytes for system)
- 31 bytes NVRAM
- Power requirements:
  5V@400mA
- Environmental:

  Operating temp: -25~75℃
  Storage temp: -40~80℃
  Humidity: 5~95% non-condensing
  Dimensions: 127mm X 121mm

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 1.9 Block Diagram

The figure 1.8 shows the block diagram of the PISO-DNM100(U) board.

1. DPRAM (Dual Port RAM) :
   The DPRAM is the memory buffer which provides the communication channel between PC and PISO-DNM100(U).

2. EEPROM :
   The EEPROM stores the configuration information. After restarting the PC, the configuration data will be loaded form the EEPROM automatically.

3. Control CPU :
   The CPU inside implementing the DeviceNet firmware.

4. CAN Controller :
   The CAN controller is used for sending and receiving the CAN messages. There is photo isolation between CAN controller and CAN bus.

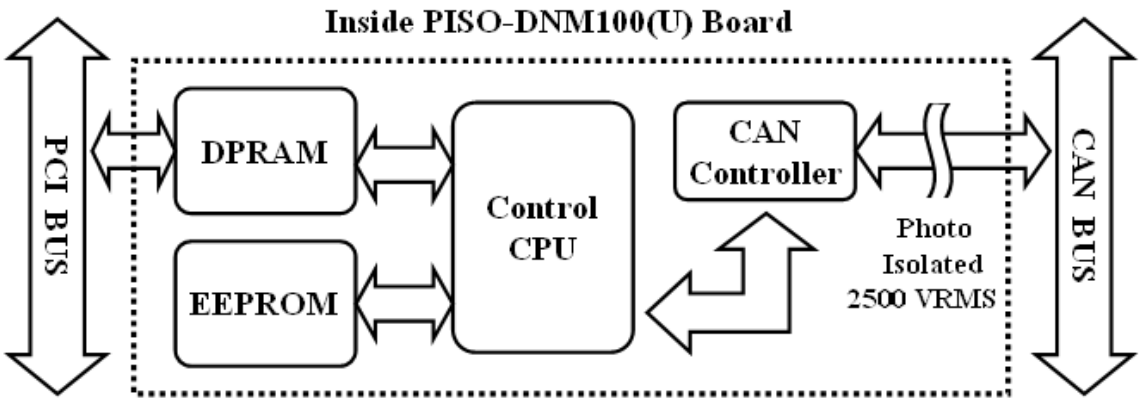



Figure 1.8 Block diagram of the PISO-DNM100(U)

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## *1.10 Product Check List*

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

- □ PISO-DNM100(U) board;
- □ Software CD ROM;
- □ Quick Start manual;
- □ Release Note

**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.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 2. Hardware Configuration

This section will describe the hardware settings of the PISO-DNM100(U). This information includes the wire connection and terminal resistance configuration for the CAN network.

## *2.1 Board Layout*

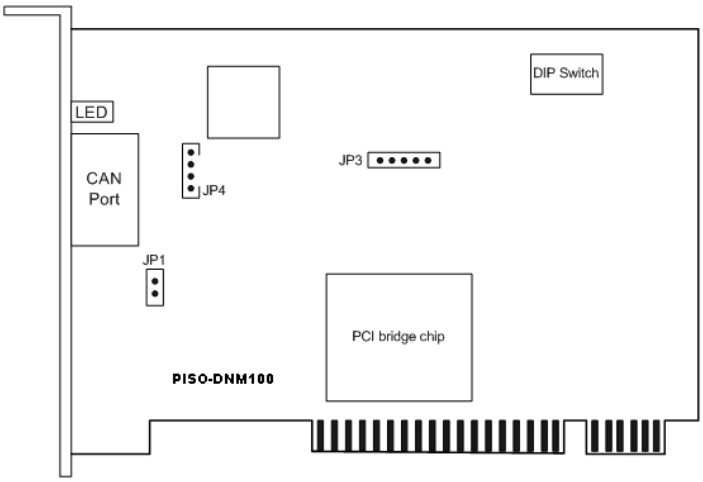



Figure2.1 PISO-DNM100 Board Layout

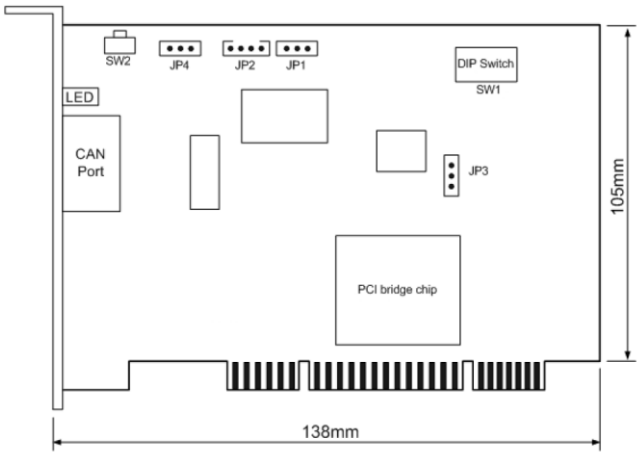



Figure 2.2 PISO-DNM100U Board Layout

Note: PISO-DNM100(U)-T layout is similar with PISO-DNM100(U)-D. The only difference is the position of CAN port connector. The positions of jumper or DIP switch are the same. Therefore, users can also refer to the PISO-DNM100(U)-D layout to configure the jumper or DIP switch if they use PISO-DNM100(U)-T.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 2.2 Jumper Selection

The following table shows the definition of jumpers or DIP switch. Users need to refer to this table to configure the PISO-DNM100(U)- D/T hardware.

PISO-DNM100 :

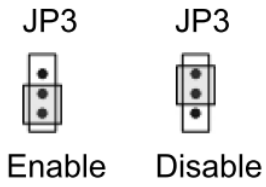
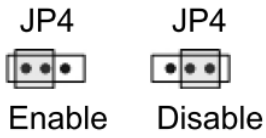
| Jumper | Description | Status |
|---|---|---|
| JP3 | The Flash protection jumper. If users would like to protect the data of the Flash. Enable this jumper. In this case, the firmware can't be updated from the utility. |  |
| JP4 | CAN Port 120Ω terminal resistance. |  |
| SW2 | The reset button for into the download mode. If users would like force the CAN board into download mode, enable this jumper to reset the firmware and let the PISO-DNM100U into the download mode. |  |
| DIP switch | DIP switch is used to set the PISO-CM100 board No. Switch1 is for bit0, switch2 is for bit1 and so forth. For example, if the left-hand-side switch (switch 1) is ON, the board No. is set to 1. The range of board No. is from 0 to 15. Be careful that the board No. for each PISO-CM100-D/T, PISO-DNM100-D/T and PISO-CPM100-D/T must be unique. | <br>This situation indicates the board No. 1. |

Table 2.1 Jumper or DIP switch selections for PISO-DNM100

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

PISO-DNM100U :

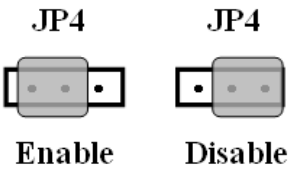
| Jumper | Description | Status |
|---|---|---|
| JP1<br>JP2<br>JP3 | None. | None. |
| SW2 | Reset button for download error. If users want to update firmware but the process is fail, users can click this button to reset the PISO-DNM100(U) into download mode. |  |
| JP4 | CAN Port 120Ω terminal resistance. |  |
| DIP switch | DIP switch is used to set the PISO-CM100(U) board No. Switch1 is for bit0, switch2 is for bit1 and so forth. For example, if the left-hand-side switch (switch 1) is ON, the board No. is set to 1. The range of board No. is from 0 to 15. Be careful that the board No. for each board<br>∗ PISO-CM100-D/T,<br>∗ PISO-CM100U-D/T,<br>∗ PISO-DNM100-D/T,<br>∗ PISO-DNM100U-D/T<br>∗ PISO-CPM100-D/T<br>∗ PISO-CPM100U-D/T<br>must be unique. | <br>This situation indicates the board No. 1. |

Table 2.2 Jumper or DIP switch selections for PISO-DNM100U

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 2.3 Connector Pin Assignment

The PISO-DNM100(U)-T is equipped with one **5-pin screw terminal connector** and the PISO-DNM100(U)-D is equipped with one **9-pin D-sub male connector** for wire connection of the CAN bus. 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.

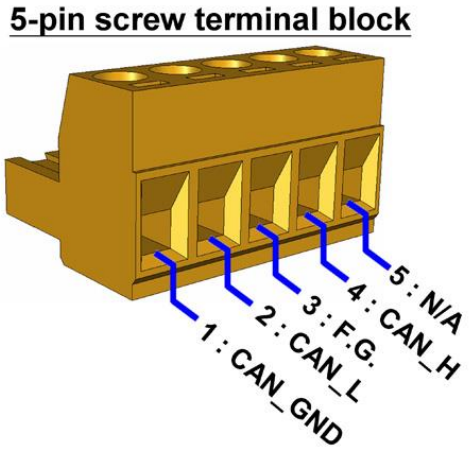



Figure2.4 5-pin screw terminal connector

| Pin No. | Signal | Description |
|---|---|---|
| 1 | CAN_GND | CAN Ground |
| 2 | CAN_L | CAN_L bus line (dominant low) |
| 3 | F.G | Optional Frame Ground |
| 4 | CAN_H | CAN_H bus line (dominant high) |
| 5 | N/A | No use |

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 2.3.2 9-pin D-sub male connector

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.

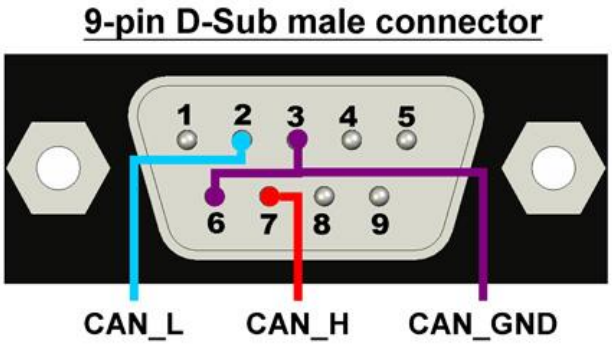



Figure2.5 9-pin D-sub male connector

| Pin No. | Signal | Description |
|---|---|---|
| 1 | N/A | No use |
| 2 | CAN_L | CAN_L bus line (dominant low) |
| 3 | CAN_GND | CAN Ground |
| 4 | N/A | No use |
| 5 | N/A | No use |
| 6 | CAN_GND | CAN Ground |
| 7 | CAN_H | CAN_H bus line (dominant high) |
| 8 | N/A | No use |
| 9 | N/A | No use |

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 2.3.3 Wire connection

In order to minimize the reflection effects on the CAN bus line, the CAN bus line has to be terminated at both ends by two terminal resistances as in the following figure. According to the ISO 11898-2 spec, each terminal resistance is 120Ω (or between 108Ω~132Ω). The length related resistance should have 70 mΩ/m. Users should check the resistances of the CAN bus, before they install a new CAN network.

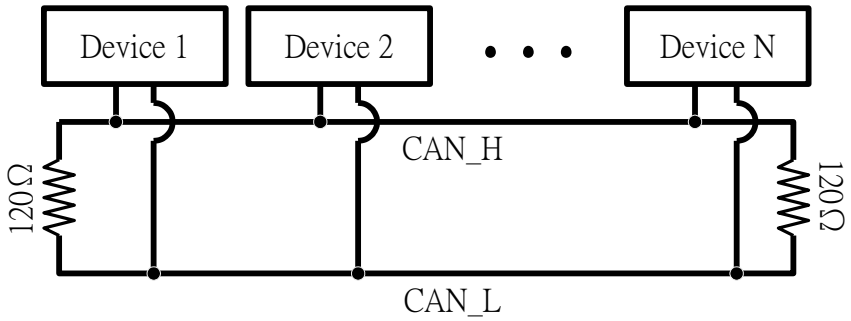



Figure 2.4 CAN bus network topology

Moreover, to minimize the voltage drop over long distances, the terminal resistance should be higher than the value defined in the ISO 11898-2. The following table can be used as a good reference.

| Bus Length (meter) | Bus Cable Parameters | | Terminal Resistance (Ω) |
|---|---|---|---|
| | Length Related Resistance (mΩ/m) | Cross Section (Type) | |
| 0~40 | 70 | 0.25(23AWG)~ 0.34mm$^2$(22AWG) | 124 (0.1%) |
| 40~300 | < 60 | 0.34(22AWG)~ 0.6mm$^2$(20AWG) | 127 (0.1%) |
| 300~600 | < 40 | 0.5~0.6mm$^2$ (20AWG) | 150~300 |
| 600~1K | < 20 | 0.75~0.8mm$^2$ (18AWG) | 150~300 |

Table 2.4 Relationship between cable characteristics and terminal resistance

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 2.4 Indicator LED

### 2.4.1 Green LED

The [Green] LED indicates the firmware status in the PISO-DNM100(U). There are 3 situations in [Green] LED.

(1). LED off:

This indicates that there are some errors on the bus or in the firmware. The DeviceNet firmware is not running.

(2). LED twinkle:

This indicates that the CAN bus works fine. But there is no any slave devices configuration in the EEPROM of the PISO-DNM100(U). The DeviceNet firmware is waiting for configuration.

(3). LED on:

This indicates that the DeviceNet firmware is running. The PISO-DNM100(U) is communicating with the slave devices.

### 2.4.2 Red LED

The [Red] LED means Network Status. It indicates that there are errors on the bus or there is any slave device's MAC ID collides with the PISO-DNM100(U)'s MAC ID. There are two situations in [Red] LED.

(1). LED off:

This indicates that there is no error on the bus and about the MAC ID.

(2). LED twinkle:

This indicates that there are errors on the bus which maybe the situations as shown bellow:

(a) The CAN connector doesn't connect to the slave devices.
(b) The power of the slave devices is off.
(c) The MAC ID collision between master and slave devices is occurring.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 2.5 Hardware Installation

When users want to use PISO-DNM100(U)-D/T, the hardware installation needs to be finished as following steps.

1. Shutdown your personal computer.
2. Configure the DIP switch and JP1 of your PISO-DNM100-D/T for board No. and terminal resistance. More detail information could be found on the figure 2.1 and table 2.1. For the PISO-DNM100U-D/T, configure the DIP switch and JP4 of your PISO-DNM100U-D/T for board No. and terminal resistance. More detail information could be found on the figure 2.2 and table 2.2.
3. Check JP3 status of PISO-DNM100-D/T. If necessary, enable it. For the PISO-DNM100U-D/T, you can visit the next step.
4. Find an empty PCI slot for your PISO-DNM100(U)-D/T on the mother board of the personal computer. Plug the configured PISO-DNM100(U)-D/T into this empty PCI slot. See figure 2.5.
5. Plug your CAN bus cable(s) into the 5-pin screw terminal connector or the 9-pin D-sub connector.

When the steps described above is completed, turn on the personal computer.

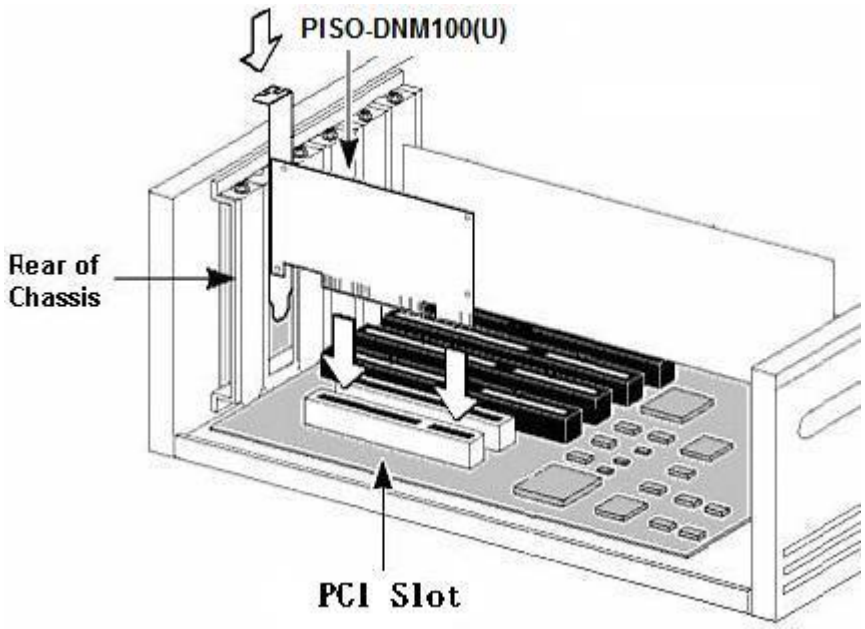



Figure 2.5 PISO-DNM100(U) installation

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 3. Driver Installation and Software Application

The DeviceNet DLL driver (DNM100.dll) collection of function calls for the PISO-DNM100(U) cards used in Windows 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 DNM100.DLL driver to implement DeviceNet network application. And then the DeviceNet DLL driver will throughout the CM100.dll into the KP_CM100.sys and windrvr6.sys to access the hardware system, as shown in the following Figure.

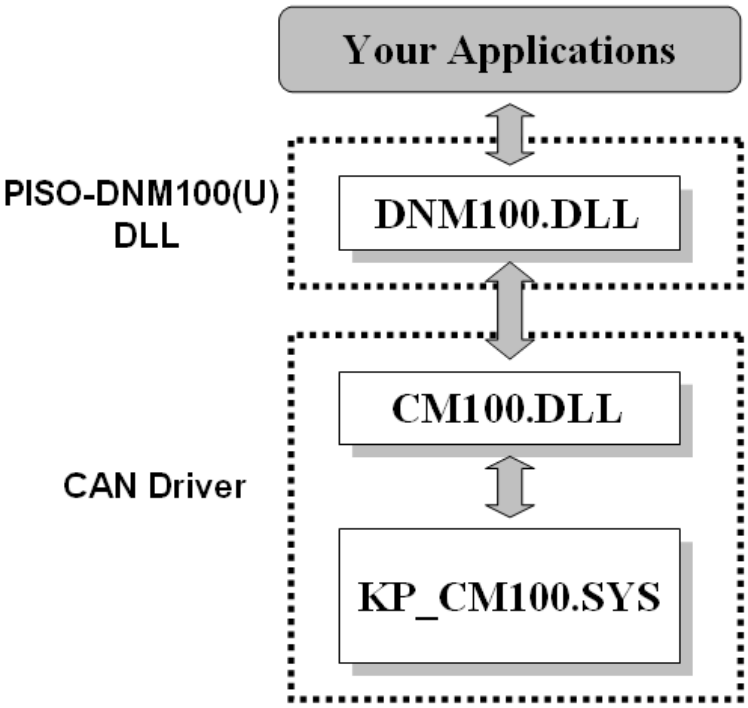



Figure 3.1 Software architecture in the Windows system

In the following sub-section, we show some flow diagrams to describe how to apply the DeviceNet protocol (DNM100.DLL) to build a master device. Section 3.2 ~ 3.10 show the flow diagram for users to understand easily. Note that users need to follow the operation principle of the DeviceNet protocol correctly and easily to communicate with the remote nodes by these connection methods.

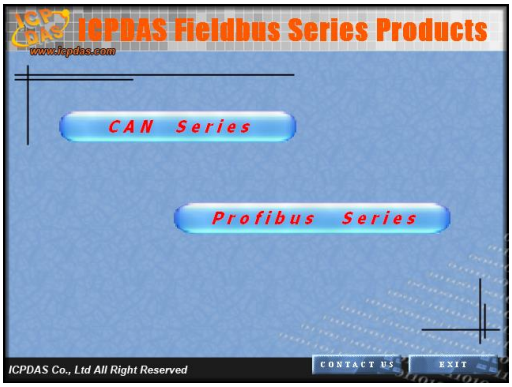
LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.1 Driver Installation of the PISO-DNM100(U)

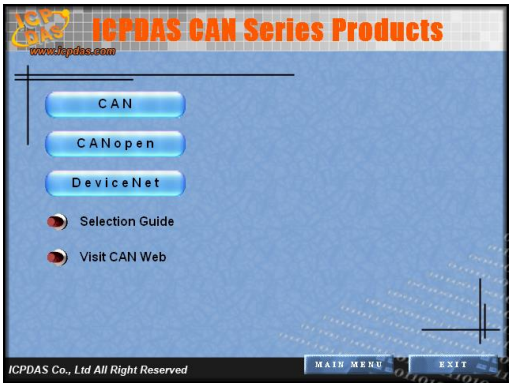
The software Installation for DeviceNet application is demonstrated as the following descriptions. After finishing the procedure, the driver, demos, manual and Utility can be in your PC. For the advance application, users can refer to the basic demo programs to develop the customized DeviceNet master application.

The driver of PISO-DNM100(U) can be used in 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 **ICPDAS.EXE** on this CD. The screenshot likes the following picture. Please click “CAN Series” item.




Step 2: After changing to the following picture, please click the “DeviceNet” item.




LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 3: When the window is changing to the picture below, please move the mouse cursor on the “Master” item.



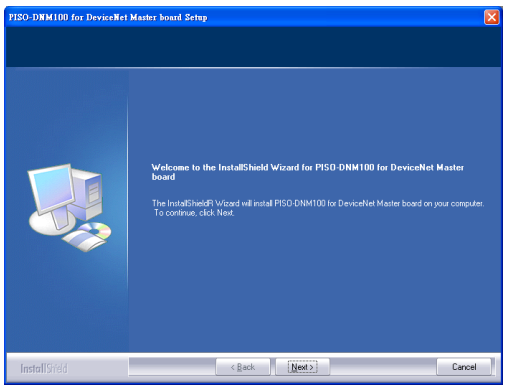

Step 4: The DeviceNet master products will be shown in the window.




Step 5: Click “install Toolkit” to install the driver of the PISO-DNM100(U) board.

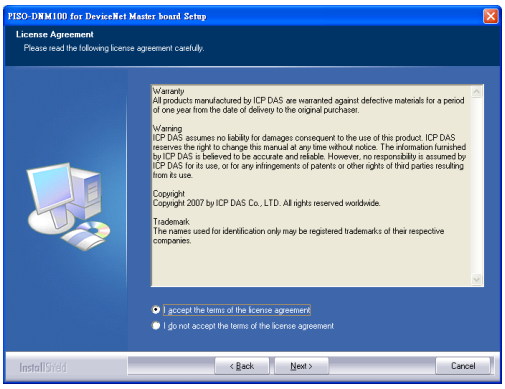
Step 6: After finishing the step 5, the users will see the setup file in the folder “PISO-DNM100_V30.exe”. Please double-click it to run the setup.

Step 7: This first setup screen shoot is shown as follows. Please press “Next” button to continue the process.

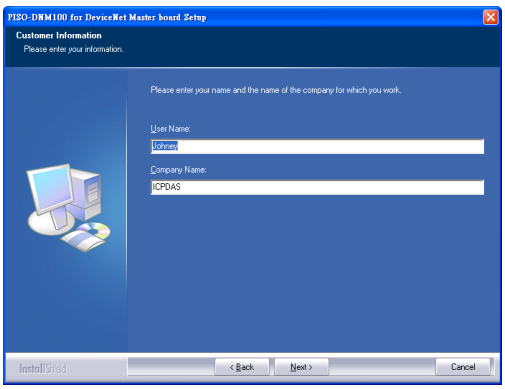



LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

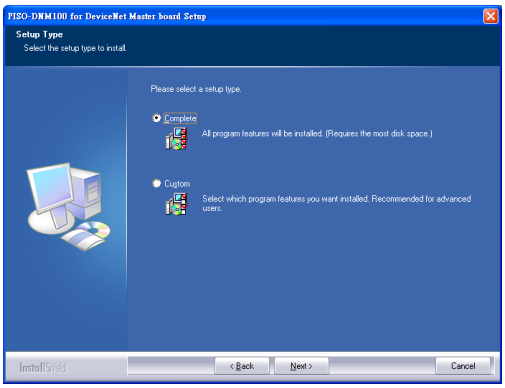
Step 8: Press “Next” button. The screen shoot is shown below. After reading the license, the users can accept it or not. If the users accept it, please select “I accept….” and press “Next” button.



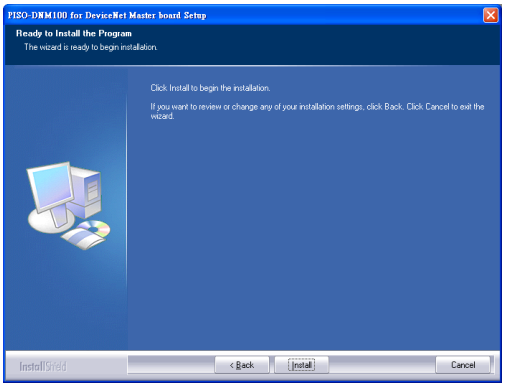

Step 9: After accepting the license, the next screen shoot is shown as follows. Users can edit your name and company name. After editing the information, please press “Next” button.



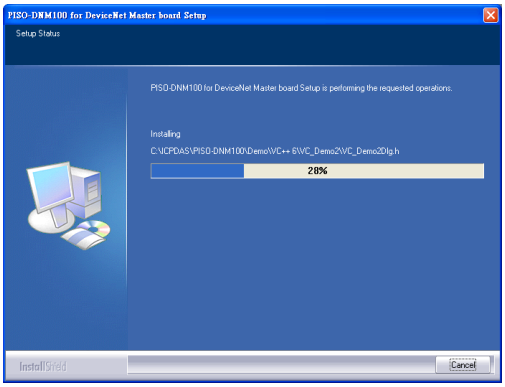

Step 10: After editing the information, the next screen shoot is shown as follows. Please select “Complete” item and press “Next” button.



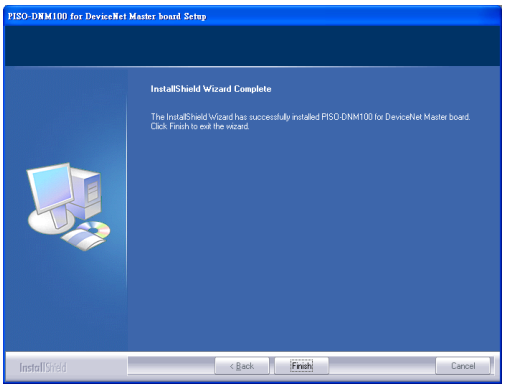
Step 11: The next screen shoot is shown as follows. Please press “Install” button. The setup process will start.




Step 12: The setup process is running. The screen shoot is shown below.

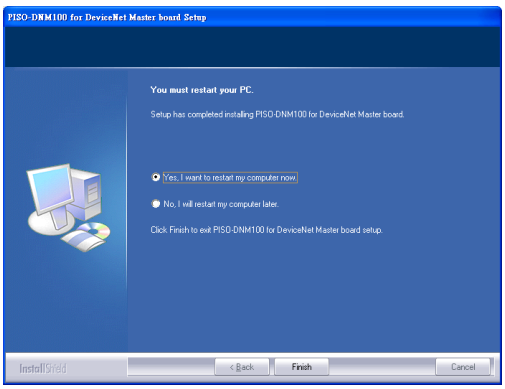



Step 13: Wait for the setup process finishing. The next screen shoot is shown below. After finishing the process, please press “Finish” button.




LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 14: The next screen shoot is shown as follows. Please restart your PC.




Then the setup software would copy the related material to the indicated directory and register the driver on your computer. The driver target directory is different according to the different systems as follows.

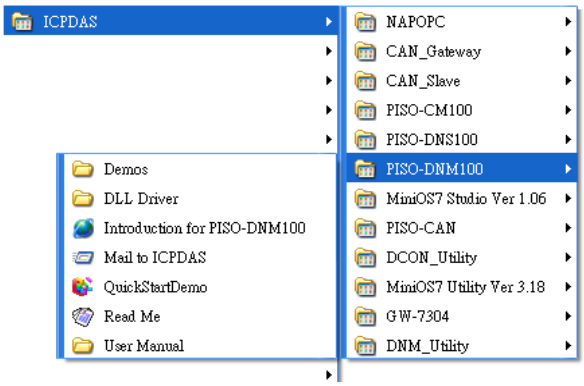
Windows NT/2000 – **WINNT\SYSTEM32\DRIVERS**

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

The other data and resource is copied to the following directory:

**C:\ICPDAS\PISO-DNM100**

**The program files picture is shown as follow.**




Note：DeviceNet Master Utility is a useful tool for users to configure and test the DeviceNet slave devices. You can find the software in the path of /devicenet/Master/DNM_Utility/ in the Fieldbus_CD or from ftp://ftp.icpdas.com/pub/cd/fieldbus_cd/devicenet/master/DNM_Utility.
After installing the software, the utility is installed in the path below.
C:\ICPDAS\DNM_Utility\DNM_Utility.exe
Please refer to the manual of utility to know the detail.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.2 Flow Diagram for Searching Devices

Before developing the DeviceNet applications, users should diagnose the connection between the slave devices. First, the users can search the slave devices in the network by using the searching functions. If the connection between the master with other slave devices is fine, the uses can find the information of the corresponding slave devices. When the users have no idea to communicate with the slave devices, users can follow these steps shown in figure 3.2. The following functions can help users to get the DeviceNet information of the slave devices. The users can find out the problem of the slave devices by using these functions. The detail information about those functions is in the next chapter.

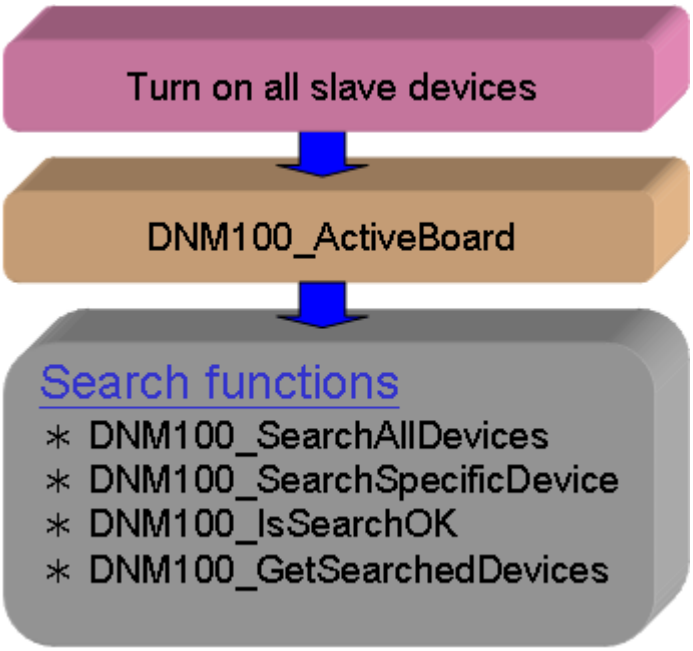



Figure 3.2 Searching Diagram

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.3 Flow Diagram for Slave Configuration

After getting the DeviceNet I/O information of the slave devices, users should save the parameters into the EEPROM in PISO-DNM100(U). The EEPROM will store the configuration data. The firmware in PISO-DNM100(U) will load the previous configuration from the EEPROM in the next boot-up. When the devices in the DeviceNet network are changed, the users must set the configuration data to fit the application. The configuration diagram is shown in Figure 3.3. There is more information about those functions in the next chapter.

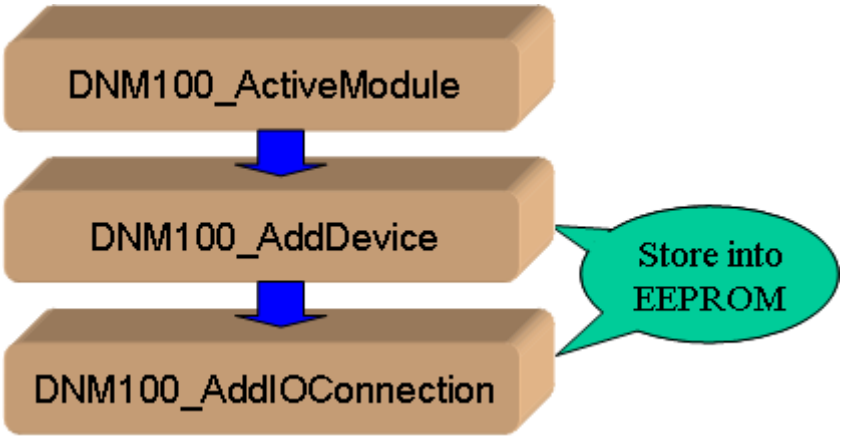



Figure 3.3 Slave Configuration Diagram

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.4 Flow Diagram for On-line Adding/Removing Device

The PISO-DNM100(U) provides the on-line adding/removing slave device functions. The users need not to break the communication between original slaves device when adding or removing the slave devices. The users can follow the steps to achieve this function. The steps are shown in Figure 3.6 and Figure 3.7.

**1. On-line Adding Devices :**

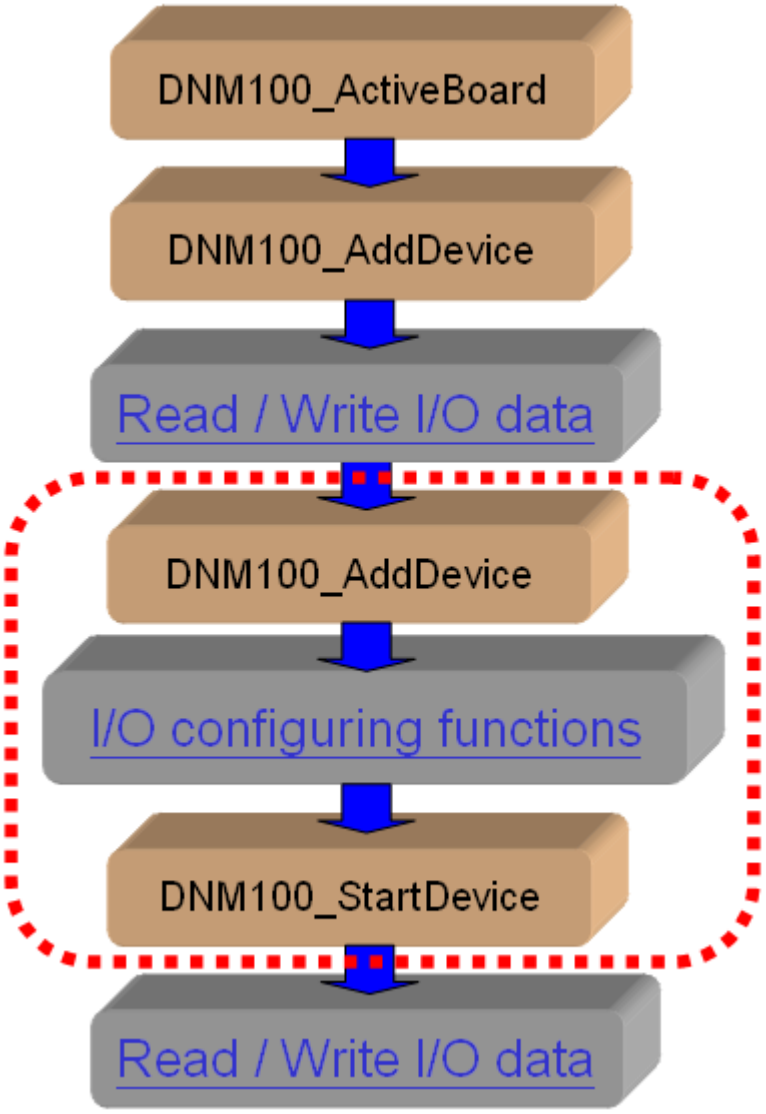



Figure 3.6 On-line Add Device Diagram

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 2. On-line Removing Devices :

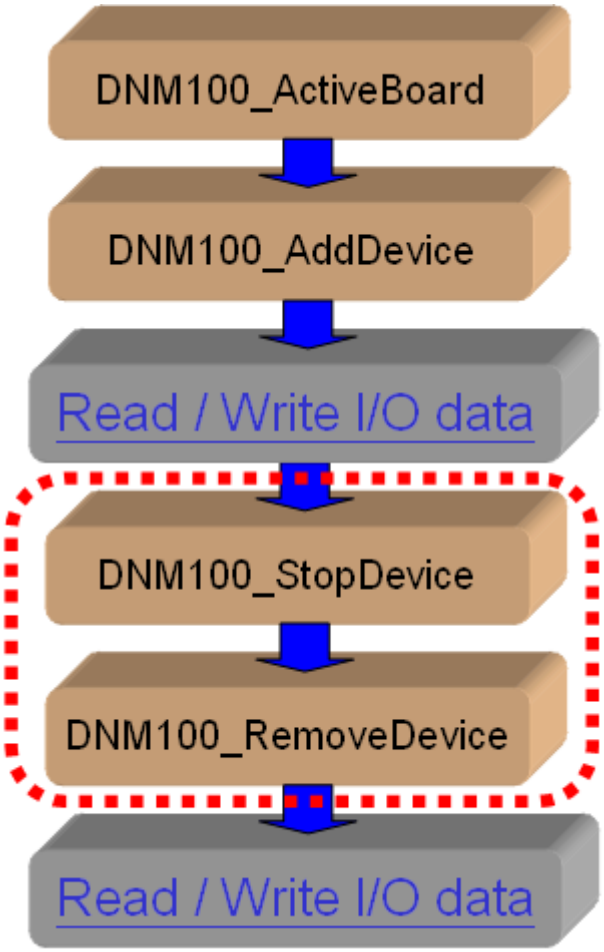



Figure 3.7 On-line Remove Device Diagram

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.5 Flow Diagram for “SetAttribute” and “GetAttribute”

The users can set or get DeviceNet device's property via DeviceNet network. The PISO-DNM100(U) provides these functions to set or get the properties of the remote devices easily. The steps are shown in Figure 3.8.

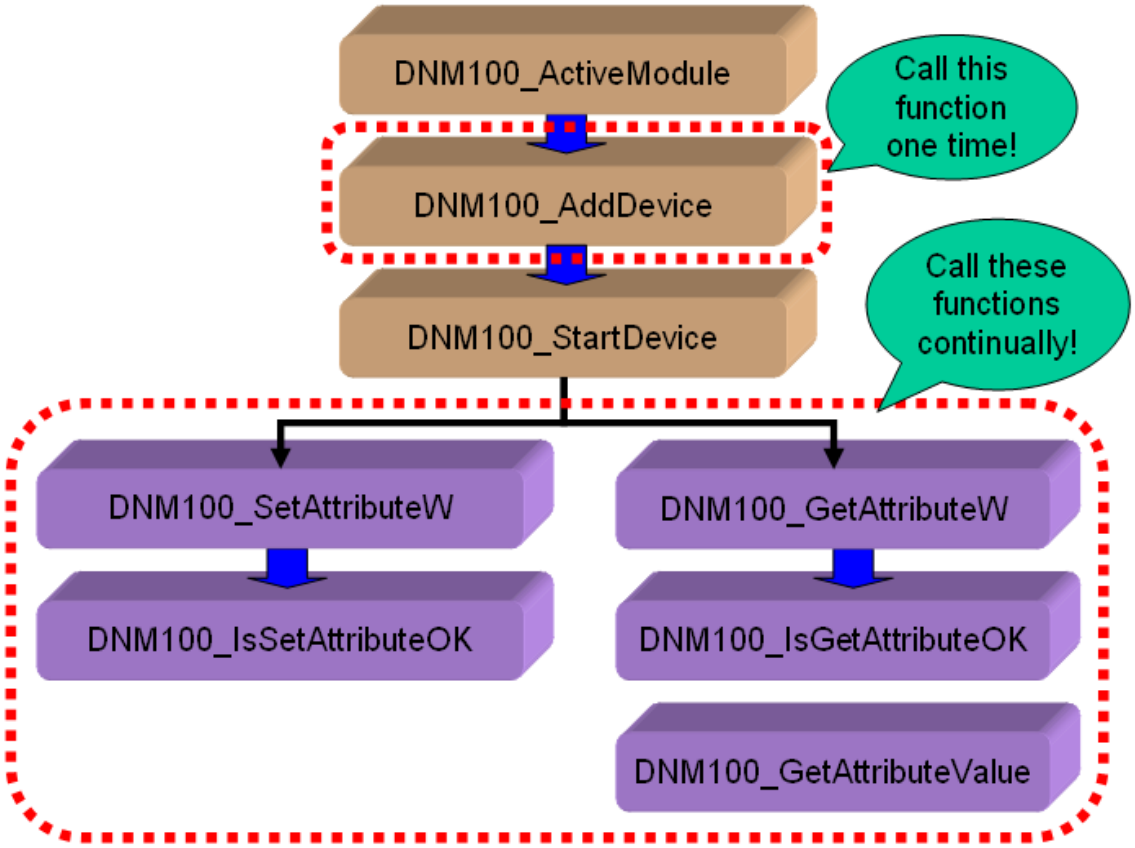



Figure 3.8 “SetAttribute” and “GetAttribute” Diagram

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.6 Flow Diagram for I/O Connection

The users can read or write device's I/O data via the DeviceNet I/O connections like Poll, Strobe, COS and Cyclic connection. There are four important steps to read and write the I/O data easily. Firstly, the users should know the device's I/O input length (in Byte) and output length (in Byte). Secondly, the users should set these two parameters by calling DNM100_AddIOConnection. Thirdly, the users can set the initial output value by calling DNM100_WriteOutputData before starting the specific slave device. If the users do not initialize the output value, the firmware default output value is 0. Fourthly, the users can start communicating with device to read or write I/O data. If the specific slave device doesn't have any output channel, the firmware will start communicating with the device automatically. The Figure 3.9 shows the main steps to achieve this function. There are more functions described in chapter 4.

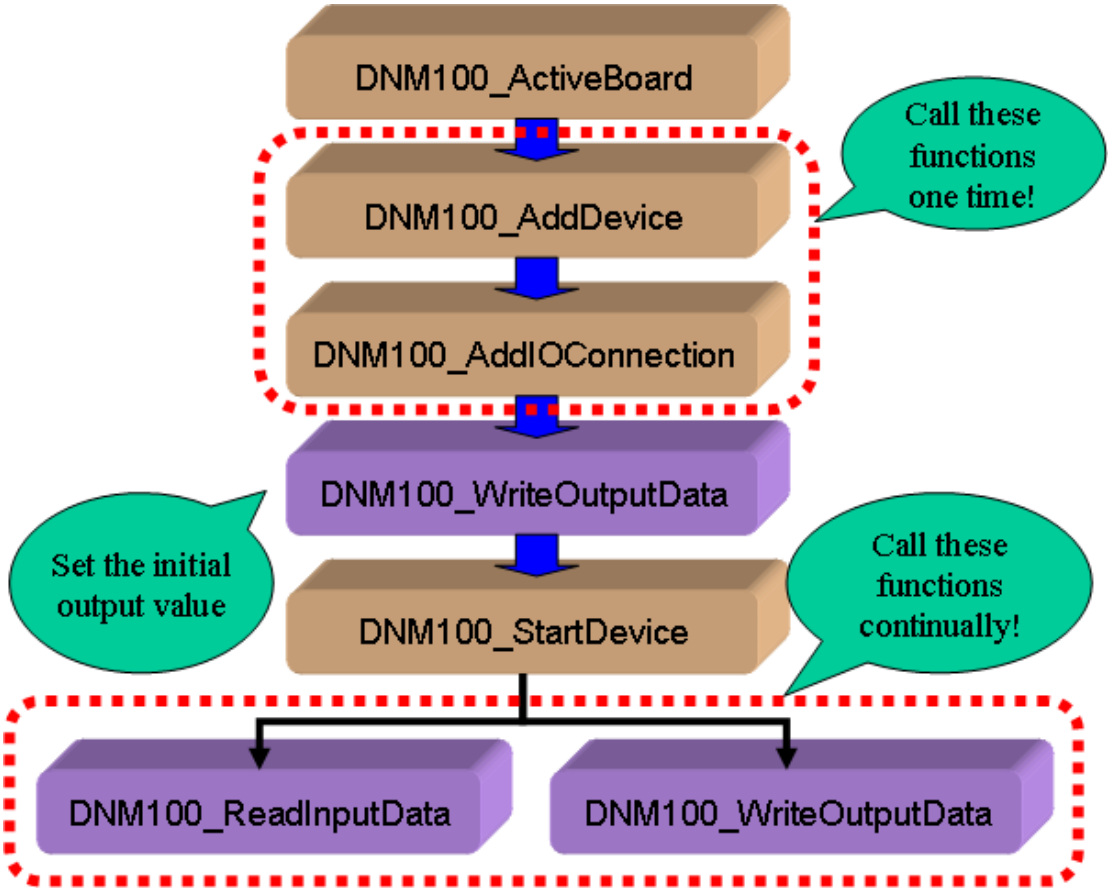



Figure 3.9 I/O Connection Diagram

Note: The Strobe connection doesn't support the output channel. The users can not use the DNM100_WriteOutputData with Strobe connection.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 3.7 Input and Output I/O Data Area (Advanced Option)

Here exist two memory areas, “Remote Input Area” and “Remote Output Area”. The input data of all DeviceNet slaves would be stored in the “Remote Input Area”, and the output data of them would be in the “Remote Output Area”. Please refer to the Figure 3.10.

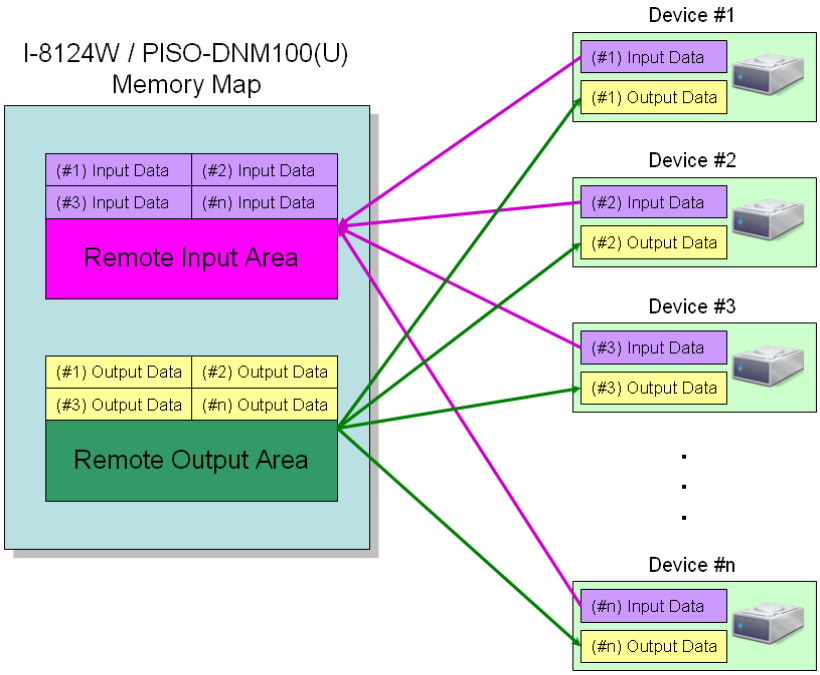



Figure 3.10 The Memory Mapping of the PISO-DNM100(U)

Users can read a bulk data from “Remote Input Area” in the PISO-DNM100(U). This bulk data contains multiple devices’ input statuses. If one of the input status of the remote DeviceNet slave changes, the corresponding data located in the “Remote Input Area” would change immediately. Oppositely, the “Remote Output Area” contains multiple devices’ output data. Users may change the output value of a certain device by changing the corresponding data located in the “Remote Output Area”.

There is another important thing. Uses need to know what the arrangement of those data. The data of the slave with the smallest DeviceNet MAC ID would be located in the most front of the Remote Input/Output Area. The data of the salve with the following MAC ID would be located in the following section, and so on. The data of the salve with the largest MAC ID would be located in the last section of the Remote Input/Output Area. Here shows three examples about the arrangement rule of the PISO-DNM100(U).

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

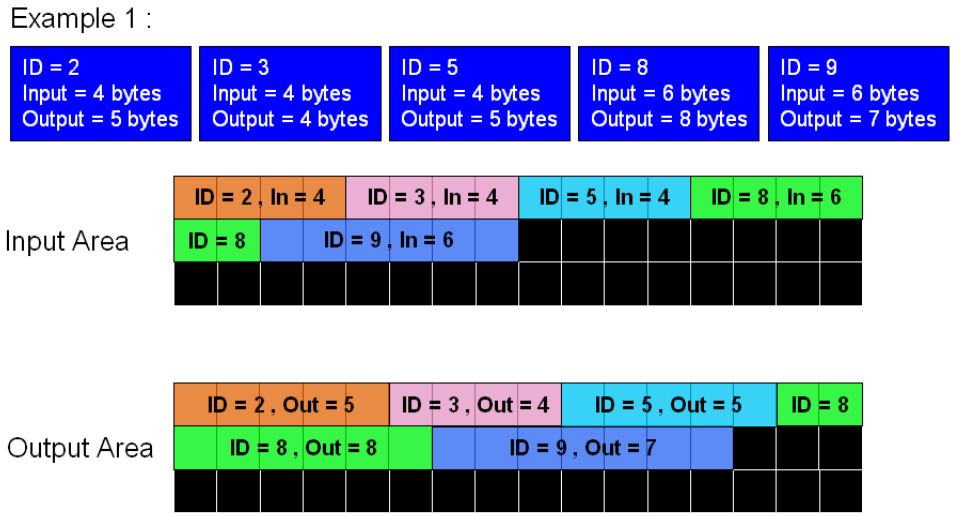



Figure 3.11 The example of the PISO-DNM100(U) memory mapping

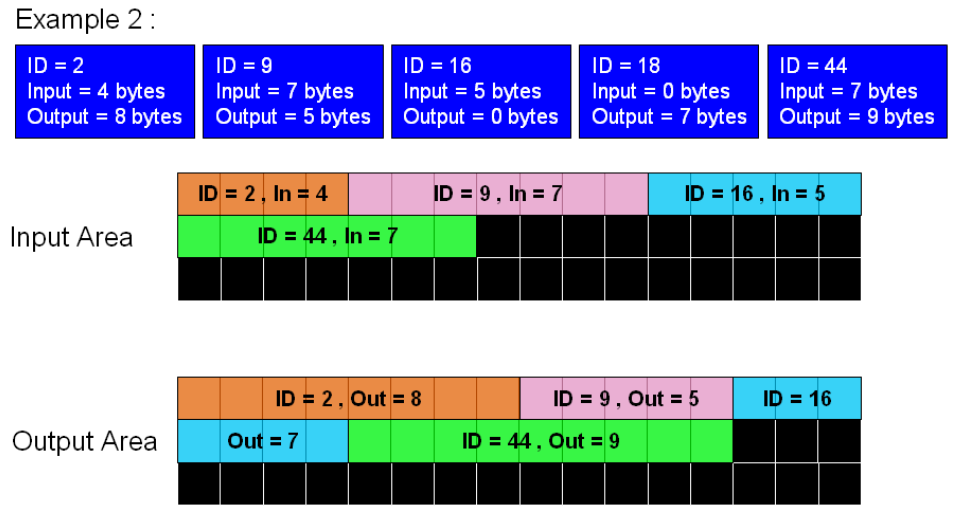



Figure 3.12 The example of the PISO-DNM100(U) memory mapping

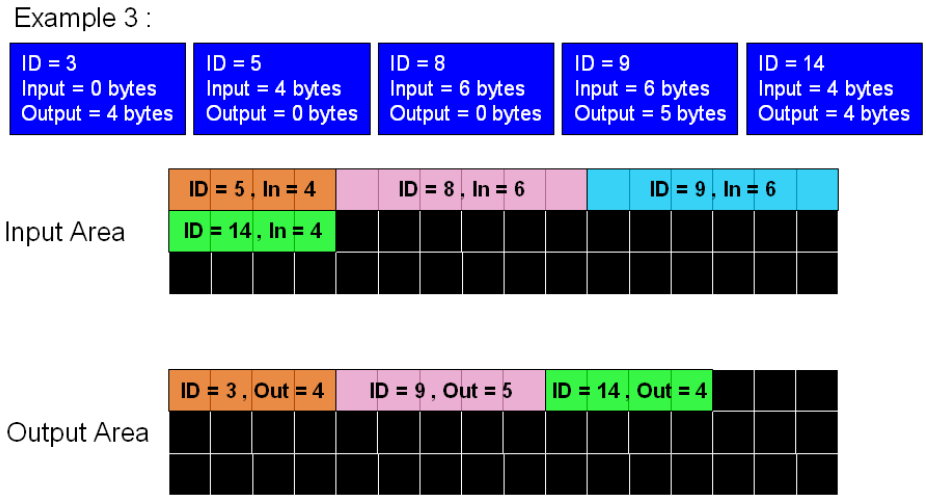



Figure 3.13 The example of the PISO-DNM100(U) memory mapping

User can read data from the “Remote Input Area” or write data to the “Remote Output Area”. There are three functions to access these two areas.

1.DNM100_ReadInputArea

Call this function to get a bulk data from the “Remote Input Area”. Please refer to the section 4.3.41 for more information.

2.DNM100_WriteOutputArea

Call this function to set a bulk data to the “Remote Output Area”. Please refer to the section 4.3.42 for more information.

3.DNM100_ReadbackOutputArea

Call this function to get a bulk data from “Remote Output Area”. Please refer to the section 4.3.43 for more information.

Note:

If users add/remove any slaves’ information into/from the PISO-DNM100(U), they need to reset firmware to update the modification of the data arrangement of the “Remote Input Area” and “Remote Output Area”. In other word, if users have called these functions below, they need to call DNM100_ResetFirmware to make the modification active.

1. DNM100_AddDevice
2. DNM100_RemoveDevice
3. DNM100_AddIOConnection
4. DNM100_RemoveIOConnection

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 4. Function description

All the functions of the PISO-DNM100(U) can be separated into five groups. The idea is shown Figure 4.1. There is more detail description in CH 4.1.

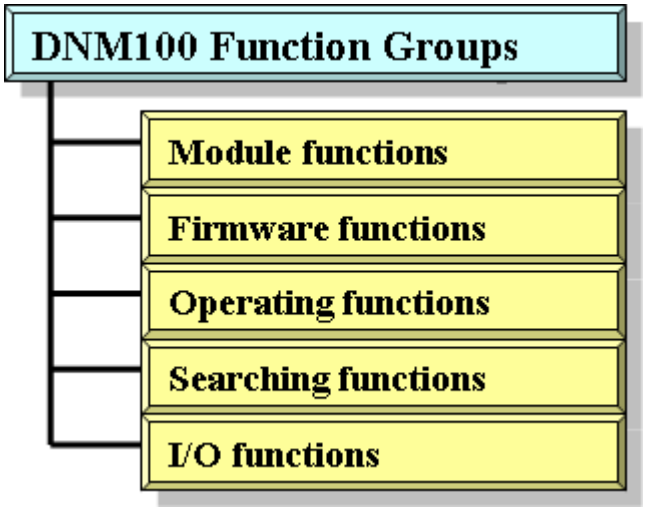



Figure 4.1 Five Function Groups

**[Board Functions]**

These functions in this group help users to find DNM100 boards or get board's information. The users can use these functions to configure or manage the boards in the PC.

**[Firmware Functions]**

These functions in this group help users to operate the firmware or get the status of the firmware inside the PISO-DNM100(U).

**[Operating Functions]**

These operating functions are the important operation of the DeviceNet master. They help users to configure the whole network.

**[Searching Functions]**

These searching functions can help user to debug the network, including the wire connection, the slave device's setting, and etc. When building the DeviceNet network, the user can use these functions to make sure that the network or the slave devices are fine.

**[I/O Functions]**

These functions help user to read or to write the I/O data from or to the

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

remote slave devices.

## *4.1 DLL Function Definition and Description*

All the functions provided in the DNM100.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 the 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 |

Table 4.1.1 Functions Table (Board Functions) 1/1

| No. | Function Name | Description |
|---|---|---|
| **1** | DNM100_GetBoardInf | Get the driver information of the PISO-DNM100(U) |
| **2** | DNM100_TotalDNM100Board | Get total PISO-DNM100(U) boards in the PC |
| **2** | DNM100_ActiveBoard | Enable the driver of the PISO-DNM100(U) |
| **3** | DNM100_CloseBoard | Close driver of the PISO-DNM100(U) |
| **4** | DNM100_GetDLLVersion | Get the DLL version of the DNM100.DLL |

Table 4.1.2 Functions Table (Firmware Functions) 1/1

| No. | Function Name | Description |
|---|---|---|
| **1** | DNM100_GetFirmwareVersion | Get the version of the firmware inside the PISO-DNM100(U) board |
| **2** | DNM100_ResetFirmware | Reset the firmware in the PISO-DNM100(U) board |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 4.1.3 Functions Table (Operating Functions) 1/2

| No. | Function Name | Description |
|---|---|---|
| **1** | DNM100_SetMasterMACID | Set the MAC ID of the PISO-DNM100(U) board (DeviceNet Master's MAC ID) |
| **2** | DNM100_GetMasterMACID | Get the MAC ID of the PISO-DNM100(U) board (DeviceNet Master's MAC ID) |
| **3** | DNM100_GetBaudRate | Get the baud rate of the CAN bus |
| **4** | DNM100_SetBaudRate | Set the baud rate of the CAN bus |
| **5** | DNM100_GetMasterStatus | Get the status of the PISO-DNM100(U) board (DeviceNet Master's status) |
| **6** | DNM100_GetSlaveStatus | Get the slave device's status. |
| **7** | DNM100_StartDevice | PISO-DNM100(U) will start to communicate with the specific slave device |
| **8** | DNM100_StopDevice | PISO-DNM100(U) will stop to communicate with the specific slave device |
| **9** | DNM100_StartAllDevice | PISO-DNM100(U) will start to communicate with all slave devices |
| **10** | DNM100_StopAllDevice | PISO-DNM100(U) will stop to communicate with all slave devices |
| **11** | DNM100_AddDevice | Add the specific slave device's information into the PISO-DNM100(U) board (DeviceNet Master) |
| **12** | DNM100_RemoveDevice | Remove the specific slave device's information from the PISO-DNM100(U) board (DeviceNet Master) |
| **13** | DNM100_AddIOConnection | Add I/O information of the specific slave device into the PISO-DNM100(U) board (DeviceNet Master) |
| **14** | DNM100_RemoveIOConnection | Remove specific slave device's I/O information from the PISO-DNM100(U) board (DeviceNet Master) |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 4.1.4 Functions Table (Operating Functions) 2/2

| No. | Function Name | Description |
|---|---|---|
| **16** | DNM100_GetAttribute | Send the get attribute command to the slave device. |
| **17** | DNM100_GetAttributeW | Send the get attribute command to the slave device. |
| **18** | DNM100_IsGetAttributeOK | Check whether the slave has replied for the getting command or not. |
| **19** | DNM100_GetAttributeValue | Get the attribute value of the DNM100_GetAttributeW |
| **20** | DNM100_SetAttribute | Send the set attribute command to the slave device. |
| **21** | DNM100_SetAttributeW | Send the set attribute command to the slave device. |
| **22** | DNM100_IsSetAttributeOK | Check whether the slave has replied for the setting command or not. |
| **23** | DNM100_GetDeviceInfoFromScanList | Get specific slave device's I/O information form the Scan List within the PISO-DNM100(U) board. |
| **24** | DNM100_GetScanList | Get the I/O information of all slave devices form the Scan List within the PISO-DNM100(U) board. |
| **25** | DNM100_ImportEEPROM | Write the I/O information of all slave devices into the EEPROM within the PISO-DNM100(U) board. |
| **26** | DNM100_ClearAllConfig | Clear all configurations in the EEPROM within the PISO-DNM100(U) board. |
| **27** | DNM100_SendExplicitMSG | Send the explicit request command. |
| **28** | DNM100_SendExplicitMSG_W | Send the explicit request command. |
| **29** | DNM100_IsExplicitMSGRespOK | Check whether the PISO-DNM100(U) has received the response message or not. |
| **30** | DNM100_GetExplicitMSGRespValue | Get the attribute value of the specific device's instance. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 4.1.5 Functions Table (Searching Functions) 1/1

| No. | Function Name | Description |
|---|---|---|
| **1** | DNM100_SearchAllDevices | PISO-DNM100(U) will search the DeviceNet network to find out the I/O information of all slave devices. |
| **2** | DNM100_SearchSpecificDevice | PISO-DNM100(U) will search the DeviceNet network to find out the I/O information of specific slave devices. |
| **3** | DNM100_IsSearchOK | Check whether the PISO-DNM100(U) has searched completely or not. |
| **4** | DNM100_GetSearchedDevices | Get the result of the searching command and retrieve the slave's I/O information. |

Table 4.1.6 Functions Table (I/O Functions) 1/1

| No. | Function Name | Description |
|---|---|---|
| **1** | DNM100_ReadInputData | Read the input data via I/O connection like Poll, Strobe, COS, Cyclic. |
| **2** | DNM100_WriteOutputData | Write the output data via I/O connection like Poll, COS, Cyclic. The Strobe doesn't support this operation. |
| **3** | DNM100_ReadbackOutputData | Read back the output data via I/O connection like Poll, COS, Cyclic. The Strobe doesn't support this operation. |
| **4** | DNM100_ReadInputArea | Read the bulk data from the input area memory. |
| **5** | DNM100_WriteOutputArea | Write the bulk data to the output area memory. |
| **6** | DNM100_ReadbackOutputArea | Read back the bulk data from the output area memory. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## *4.2 Function Return Code*

Table 4.2.1 Interpretation of the return code (Hardware Error) 1/1

| Return Code | Error ID | Comment |
|---|---|---|
| **0** | DNM100_NoError | No error |
| **10001** | DNM100_DriverError | Kernel driver is not opened. |
| **10002** | DNM100_ActiveBoardError | This board can not be activated. |
| **10003** | DNM100_BoardNumberError | The Board number exceeds the total board numbers. |
| **10004** | DNM100_PortNumberError | The Port number is not correct. |
| **10007** | DNM100_InitError | The PISO-DNM100(U) replies error. |
| **10021** | DNM100_SoftBufferIsEmpty | No CAN messages in the buffer. |
| **10022** | DNM100_SoftBufferIsFull | The software buffer is overflow. |
| **10023** | DNM100_TimeOut | The PISO-DNM100(U) has no response. |
| **10024** | DNM100_SetCyclicMsgFailure | The cyclic messages are over 5 counts. This is special function for CAN. |
| **10025** | DNM100_DpramOverRange | The command length is over 512 bytes. |
| **10026** | DNM100_NoDpramCmd | There is no command in DPRAM. |
| **10027** | DNM100_ModeError | This board can't be changed to firmware mode. |
| **10030** | DNM100_NoFileInside | There is no firmware in PISO-DNM100(U). |
| **10031** | DNM100_DownloadFailure | The download firmware process is failure. |
| **10032** | DNM100_EEPROMDamage | The EEPROM is out of order. |
| **10033** | DNM100_NotEnoughSpace | The firmware is too large to put it into the PISO-DNM100(U)-D/T |
| **10034** | DNM100_StillDownloading | The firmware is downloading. |
| **10035** | DNM100_BoardModeError | The firmware mode is error. |
| **10036** | DNM100_CardTypeError | The firmware is not for the PISO-DNM100(U). |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 4.2.2 Interpretation of the return code (General Error) 1/1

| Return Code | Error ID | Comment |
|---|---|---|
| **5000** | DNMXS_UnKnowError | The DeviceNet has some unknown errors. |
| **1000** | DNMXS_BoardNotActive | The PISO-DNM100(U) has not been activated. |
| **1001** | DNMXS_OnlineError | The master MAC ID collides with other slave device in the DeviceNet network. |
| **1002** | DNMXS_CANBusError | The CAN port can't send message. Please check the baud rate or the port of the CAN bus. |
| **1003** | DNMXS_Booting | The PISO-DNM100(U) is still booting. |
| **1050** | DNMXS_MACIDError | The MAC ID is exceed the range(0 ~ 63) |
| **1051** | DNMXS_BaudRateError | The baud rate is exceed the range(0 ~ 2) |
| **1052** | DNMXS_ConnectionTypeError | The connection type is exceed the range (0 ~ 4) |
| **1053** | DNMXS_DuplicMasterMACID | The MAC ID is the same with the master's ID. |
| **1054** | DNMXS_EEPROMError | The EEPROM is out of order. |
| **1055** | DNMXS_NowScanning | The PISO-DNM100(U) is searching the slave. |
| **1056** | DNMXS_ScanListError | The Scan List has some errors. |
| **1057** | DNMXS_DeviceExist | The information of the slave device already exists. |
| **1058** | DNMXS_DeviceNotExist | The information of the slave device doesn't exist. |
| **1059** | DNMXS_MapTableError | The MapTable has some errors. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 4.2.3 Interpretation of the return code (I/O Error) 1/1

| Return Code | MapTable Error | Comment |
|---|---|---|
| **1100** | DNMXS_ExplicitNotAllocate | The Explicit connection is not established. |
| **1101** | DNMXS_PollNotAllocate | The Poll connection is not established. |
| **1102** | DNMXS_BitStrobeNotAllocate | The Strobe connection is not established. |
| **1103** | DNMXS_COSNotAllocate | The COS connection is not established. |
| **1104** | DNMXS_CyclicNotAllocate | The Cyclic connection is not established. |
| **1105** | DNMXS_PollAlreadyExist | The Poll connection has been established. |
| **1106** | DNMXS_BitStrobeAlreadyExist | The Poll connection has been established. |
| **1107** | DNMXS_COSAlreadyExist | The COS connection has been established. |
| **1108** | DNMXS_CyclicAlreadyExist | The Cyclic connection has been established. |
| **1109** | DNMXS_CommunicationPause | The communication between PISO-DNM100(U) and all slave devices has been suspended. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 4.2.5 Interpretation of the return code (Slave Error) 1/1

| Return Code | DeviceNet Error | Comment |
|---|---|---|
| **1150** | DNMXS_SlaveNoResp | The slave has no any response. |
| **1151** | DNMXS_WaitForSlaveResp | The PISO-DNM100(U) is waiting for the response form the slave device. |
| **1152** | DNMXS_SlaveRespError | The slave replied some errors. |
| **1153** | DNMXS_OutputDataLenError | The output length of the I/O connection doesn't match the device's output length. |
| **1154** | DNMXS_InputDataLenError | The input length of the I/O connection doesn't match the device's input length. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3 Function Description

### 4.3.1 DNM100_GetBoardInfo

- **Description:**

  This function is used to obtain the driver information of PISO-DNM100(U) board.

- **Syntax:**

```
DWORD DNM100_GetBoardInf (BYTE BoardNo, DWORD *dwVID,
                          DWORD *dwDID, DWORD *dwSVID,
                          DWORD *dwSDID, DWORD *dwSAuxID,
                          DWORD *dwIrqNo)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **dwVID**: [output] The address of a variable which is used to receive the vendor ID.
  **dwDID**: [output] The address of a variable used to receive device ID.
  **dwSVID**: [output] The address of a variable applied to receive sub-vendor ID.
  **dwSDID**: [output] The address of a variable applied to receive sub-device ID.
  **dwSAuxID**: [output] The address of a variable used to receive sub-auxiliary ID.
  **dwIrqNo**: [output] The address of a variable used to receive logical interrupt number.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

### 4.3.2 DNM100_TotalDNM100Board

- **Description:**

  The function can get the count of total PISO-DNM100(U) boards in the user's PC.

- **Syntax:**

  DWORD DNM100_TotalDNM100board (BYTE *TotalBoards ,
  BYTE *BoardIDList)

- **Parameter:**

  **TotalBoards:** [output] The amount of total PISO-DNM100(U) boards.
  **BoardIDList:** [output] The list of all DIP-Switch No. in each boards.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.3 DNM100_ActiveBoard

### 4.3.3 DNM100_ActiveBoard

- **Description:**

  The function is used to activate the PISO-DNM100(U) board. It must be called once before using the other functions of PISO-DNM100(U) APIs.

- **Syntax:**

  DWORD DNM100_ActiveBoard (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.4 DNM100_CloseBoard

### 4.3.4 DNM100_CloseBoard

- **Description:**

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

- **Syntax:**

  DWORD DNM100_CloseBoard (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.5 DNM100_GetDLLVersion

- **Description:**

  The function can obtain the version information of DNM100.DLL.

- **Syntax:**

  DWORD DNM100_GetDLLVersion (void)

- **Parameter:**

  None

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.6 DNM100_GetFirmwareVersion

- **Description:**

  The function can obtain the version information of the firmware inside the PISO-DNM100(U) board.

- **Syntax:**

  DWORD DNM100_GetFirmwareVersion (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  The firmware version information. For example: If 100(hex) is return, it means firmware version is 1.00.

- **Error Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.7 DNM100_ResetFirmware

## 4.3.7 DNM100_ResetFirmware

- **Description:**

  The function is used to reset the PISO-DNM100(U) firmware. When the users have changed the baud rate of CAN bus or changed the Master's MAC ID, the function must be called to make the change enable. After calling this function, the users should wait for 1 or 2 seconds to make the firmware boot up completely.

- **Syntax:**

  DWORD DNM100_ResetFirmware (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.8 DNM100_GetMasterMACID

## 4.3.8 DNM100_GetMasterMACID

- **Description:**

  The function can get the MAC ID of the DeviceNet master (PISO-DNM100(U)).

- **Syntax:**

  DWORD DNM100_GetMasterMACID (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.9 DNM100_SetMasterMACID

### 4.3.9 DNM100_SetMasterMACID

- **Description:**

  The function can set the MAC ID of the DeviceNet master (PISO-DNM100(U)). After calling this function, the users must call DNM100_ResetFirmware to make the change enabled. It will save the information in the EEPROM in the PISO-DNM100(U).

- **Syntax:**

  DWORD DNM100_SetMasterMACID (BYTE BoardNo,
  BYTE MasterMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **MasterMACID:** [input] The new MAC ID of the master. (0 ~ 63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.10 DNM100_GetBaudRate

- **Description:**

  This function can help you to get the DeviceNet baud rate information of PISO-DNM100(U).

- **Syntax:**

  DWORD DNM100_GetBaudRate (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  The CAN bus baud rate information in the PISO-DNM100(U).
  If the value is 0, the baud rate is 125Kbps.
  If the value is 1, the baud rate is 250Kbps.
  If the value is 2, the baud rate is 500Kbps.

- **Error Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.11 DNM100_SetBaudRate

- **Description:**

  This function can set the DeviceNet baud rate of the PISO-DNM100(U). After calling this function, you must call DNM100_ResetFirmware to reset the firmware to make change enabled.

- **Syntax:**

  DWORD DNM100_SetBaudRate (BYTE BoardNo,BYTE BaudRate)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **BaudRate:** [input] The new baud rate value.
  0 : 125K bps
  1 : 250K bps
  2 : 500K bps

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.12 DNM100_GetMasterStatus

- **Description:**

  The function is used to obtain the firmware status inside the PISO-DNM100(U). The users can call this function to make sure that the DeviceNet master is online successfully.

- **Syntax:**

  DWORD DNM100_GetMasterStatus (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.13 DNM100_GetSlaveStatus

- **Description:**

  This function is to get the remote slave device's communication status.

- **Syntax:**

  DWORD DNM100_GetSlaveStatus (BYTE BoardNo,
  BYTE DesMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DesMACID:** [input] The remote slave's MAC ID. (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.14 DNM100_StartDevice

- **Description:**

  This function is used to start to communicate with the specific device that the users applying to.

- **Syntax:**

  DWORD DNM100_StartDevice (BYTE BoardNo, BYTE DesMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DesMACID:** [input] The remote slave's MAC ID. (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.15 DNM100_StopDevice

- **Description:**

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

- **Syntax:**

  DWORD DNM100_StopDevice (BYTE BoardNo, BYTE DesMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.16 DNM100_StartAllDevice

- **Description:**

  This function is used to start to communicate with all slave devices in ScanList.

- **Syntax:**

  DWORD DNM100_StartAllDevice (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.17 DNM100_StopAllDevice

- **Description:**

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

- **Syntax:**

  DWORD DNM100_StopAllDevice (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.18 DNM100_AddDevice

- **Description:**

  This function can add the slave devices into the ScanList of the PISO-DNM100(U) and save the information into the EEPROM. Before communicating with any slave devices, the users should call this function to add these devices.

- **Syntax:**

  DWORD DNM100_AddDevice (BYTE BoardNo, BYTE DesMACID,
  WORD Explicit_EPR)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **Explicit_EPR:** [input] The Expected Packet Rate. (Usually is 2500).

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.19 DNM100_RemoveDevice

## 4.3.19 DNM100_RemoveDevice

- **Description:**

  This function is used for removing the specified slave device from the ScanList in the PISO-DNM100(U). And the information of the device in EEPROM is erased at the same time.

- **Syntax:**

  DWORD DNM100_RemoveDevice (BYTE BoardNo, BYTE DesMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.20 DNM100_AddIOConnection

- **Description:**

  This method is used to configure the I/O connection of the specific MAC ID device. The PISO-DNM100(U) 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. This configuration data will be saved into EEPROM within the PISO-DNM100(U).

- **Syntax:**

```
DWORD DNM100_AddIOConnection (BYTE BoardNo, BYTE DesMACID,
                              BYTE ConType,
                              WORD DeviceInputLen,
                              WORD DeviceOutputLen,
                              WORD EPR)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ConType:** [input] The remote slave device's I/O connection type
  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type
  **DeviceInputLen:** [input] The remote slave device's input length. (Byte)
  **DeviceOutputLen:** [input] The remote slave device's output length. (Byte)
  **EPR:** [input] The expected packet rate. (mSec)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.21 DNM100_RemoveIOConnection

## 4.3.21 DNM100_RemoveIOConnection

- **Description:**

  The function is used to remove the I/O connection configuration.

- **Syntax:**

  DWORD DNM100_RemoveIOConnection (BYTE BoardNo,
  BYTE DesMACID,
  BYTE ConType)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ConType:** [input] The remote slave device's I/O connection type
  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.22 DNM100_GetAttribute

### 4.3.22 DNM100_GetAttribute

- **Description:**

  This function is used to send the request command to retrieve the attribute value of the specific device's instance. Before calling this function, you must start the device. After calling this function, you should execute the "DNM100_GetAttributeValue" to get the response message returned from remote slave device.

  This old function will be removed in the future. Please use the new function which is "DNM100_GetAttributeW".

- **Syntax:**

```
DWORD DNM100_GetAttribute (BYTE BoardNo, BYTE DesMACID,
                           BYTE ClassID, BYTE InstanceID,
                           BYTE AttributeID)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ClassID:** [input] The remote slave device's ClassID(BYTE)
  **InstanceID:** [input] The remote slave device's InstanceID(BYTE)
  **AttributeID:** [input] The remote slave device's AttributeID

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.23 DNM100_GetAttributeW

- **Description:**

  This function is used to send the request command to retrieve the attribute value of the specific device's instance. Before calling this function, you must start the device. After calling this function, you should execute the "DNM100_GetAttributeValue" to get the response message returned from remote slave device.

  This function could totally complain with the old function which has the same name without the "W". The user could use this function instead of the "DNM100_GetAttribute".

- **Syntax:**

```
DWORD DNM100_GetAttributeW (BYTE BoardNo, BYTE DesMACID,
                            WORD ClassID, WORD InstanceID,
                            BYTE AttributeID)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ClassID:** [input] The remote slave device's ClassID(WORD)
  **InstanceID:** [input] The remote slave device's InstanceID(WORD)
  **AttributeID:** [input] The remote slave device's AttributeID

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.24 DNM100_IsGetAttributeOK

- **Description:**

  This function is used to check whether the PISO-DNM100(U) has received the response message or not. After checking the response message, you should execute the “DNM100_GetAttributeValue” to get the response message returned from remote slave device.

- **Syntax:**

  DWORD DNM100_IsGetAttributeOK (BYTE BoardNo, BYTE DesMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device’s MAC ID (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.25 DNM100_GetAttributeValue

### 4.3.25 DNM100_GetAttributeValue

- **Description:**

  This function is used to get the attribute value of the specific device's instance from the remote slave device. Before calling this function, the users should call DNM100_GetAttributeW to send request command first.

- **Syntax:**

  DWORD DNM100_GetAttributeValue (BYTE BoardNo, BYTE DesMACID, WORD *DataLen, BYTE *DATA)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **DataLen:** [output] The length of the attribute value (in byte).
  **DATA:** [output] The attribute value that returned from the slave device.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.26 DNM100_SetAttribute

### 4.3.26 DNM100_SetAttribute

- **Description:**

  The method is used to set the attribute of the specific device's instance. Before calling this function, you must start the device. After calling this function, you should execute the "DNM100_IsSetAttributeOK" to check the response message returned from the remote slave device.

  This old function will be removed in the future. Please use the new function which is "DNM100_SetAttributeW".

- **Syntax:**

```
DWORD DNM100_SetAttribute (BYTE BoardNo, BYTE DesMACID,
                           BYTE ClassID, BYTE InstanceID,
                           BYTE AttributeID, WORD DataLen,
                           BYTE *DATA)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ClassID:** [input] The remote slave device's ClassID(BYTE)
  **InstanceID:** [input] The remote slave device's InstanceID(BYTE)
  **AttributeID:** [input] The remote slave device's AttributeID
  **DataLen:** [input] The length of the attribute value (in byte).
  **DATA:** [input] The attribute value that the users want to send.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.27 DNM100_SetAttributeW

- **Description:**

The method is used to set the attribute of the specific device's instance. Before calling this function, you must start the device. After calling this function, you should execute the "DNM100_IsSetAttributeOK" to check the response message returned from the remote slave device.

This function could totally complain with the old function which has the same name without the "W". The user could use this function instead of the "DNM100_SetAttribute".

- **Syntax:**

```
DWORD DNM100_SetAttributeW (BYTE BoardNo, BYTE DesMACID,
                            WORD ClassID, WORD InstanceID,
                            BYTE AttributeID, WORD DataLen,
                            BYTE *DATA)
```

- **Parameter:**

**BoardNo**: [input] PISO-DNM100(U) board number (0~15)
**DestMACID:** [input] The remote slave device's MAC ID (0~63)
**ClassID:** [input] The remote slave device's ClassID(WORD)
**InstanceID:** [input] The remote slave device's InstanceID(WORD)
**AttributeID:** [input] The remote slave device's AttributeID
**DataLen:** [input] The length of the attribute value (in byte).
**DATA:** [input] The attribute value that the users want to send.

- **Return:**

Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.28 DNM100_IsSetAttributeOK

## 4.3.28 DNM100_IsSetAttributeOK

- **Description:**

  This function is used to get the response value after executing the "DNM100_SetAttributeW" function.

- **Syntax:**

  DWORD DNM100_IsSetAttributeOK (BYTE BoardNo, BYTE DesMACID)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.29 DNM100_ClearAllConfig

### 4.3.29 DNM100_ClearAllConfig

- **Description:**

  This function will clear all configurations in the EEPROM of the PISO-DNM100(U).

- **Syntax:**

  DWORD DNM100_ClearAllConfig (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.30 DNM100_SearchAllDevices

## 4.3.30 DNM100_SearchAllDevices

- **Description:**

  This function is used to retrieve all devices in DeviceNet network. This function makes the PISO-DNM100(U) to start the searching process. The users need to check whether the process is complete or not by calling the “DNM100_IsSearchOK”. After completing the search process, the users could call the “DNM100_GetSearchedDevices“ to get the searched devices. Attention! This function will terminate all communications with remote devices. This function is usually used for developing or debugging applications.

- **Syntax:**

  DWORD DNM100_SearchAllDevices (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.31 DNM100_SearchSpecificDevice

### 4.3.31 DNM100_SearchSpecificDevice

- **Description:**

  This function is used to retrieve some devices which specified by the users. This function makes the PISO-DNM100(U) to start the searching process. The users need to check whether the process is complete or not by calling the “DNM100_IsSearchOK”. After completing the search process, the users could call the “DNM100_GetSearchedDevices“ to get the searched devices. Attention! This function will terminate all communications with remote devices. This function is usually used for developing or debugging applications.

- **Syntax:**

  DWORD DNM100_SearchSpecificDevice (BYTE BoardNo,
  WORD ListCount,
  BYTE *DesMACIDList)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **ListCount:** [input] The amount of the slave’s ID.
  **DestMACIDList:** [input] The list of all slave’s MAC ID.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.32 DNM100_IsSearchOK

### 4.3.32 DNM100_IsSearchOK

- **Description:**

  This function will check whether the searching process has finished or not.

- **Syntax:**

  DWORD DNM100_IsSearchOK (BYTE BoardNo)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.33 DNM100_GetSearchedDevices

## 4.3.33 DNM100_GetSearchedDevices

- **Description:**

  This function will get the device which have been searched in the network

- **Syntax:**

  DWORD DNM100_GetSearchedDevices (BYTE BoardNo,
  WORD *TotalDevices,
  BYTE *DesMACID,
  BYTE *Type,
  WORD *DeviceInputLen,
  WORD *DeviceOutputLen)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **TotalDevices:** [output] The amount of all slave device which are found.
  **DesMACID:** [output] The list of slave's MAC ID which are found.
  **Type:** [output] The list of slave's connection type which are found.
  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

  **DeviceInputLen:** [output] The list of slave's input length which are found.
  **DeviceOutputLen:** [output] The list of slave's output length which are found.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.34 DNM100_GetDeviceInfoFromScanList

## 4.3.34 DNM100_GetDeviceInfoFromScanList

- **Description:**

  This function will get the ScanList data of certain device in the PISO-DNM100(U).

- **Syntax:**

```
DWORD DNM100_GetDeviceInfoFromScanList
                    (BYTE BoardNo, BYTE DesMACID, WORD *ListCount,
                     BYTE *ConnectionTypeList, WORD *InputDataLenList,
                     WORD *OutputDataLenList,WORD *EPRList)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DesMACID:** [input] The MAC ID number.
  **ListCount:** [output] The amount of all information items.
  **ConnectionTypeList:** [output] The list of slave's connection type.

  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

  **InputDataLenList:** [output] The list of slave's input length.
  **OutputDataLenList:** [output] The list of slave's output length.
  **EPRList:** [output] The list of slave's expected packet rate.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.35 DNM100_GetScanList

### 4.3.35 DNM100_GetScanList

- **Description:**

  This function will get all the ScanList data in the PISO-DNM100(U).

- **Syntax:**

```
DWORD DNM100_GetScanList (BYTE BoardNo, WORD *TotalDevices,
                          BYTE *DesMACIDList,
                          BYTE *ConnectionTypeList,
                          WORD *InputDataLenList,
                          WORD *OutputDataLenList,
                          WORD *EPR_List)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **TotalDevices:** [output] The data count of all the information.
  **DestMACIDList:** [output] The MAC ID of all the slave devices in the ScanList.
  **ConnectionTypeList:** [output] The connection type of all the slave devices in the ScanList.
  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

  **InputDataLenList:** [output] The input data length of all the slave devices in the ScanList.
  **OutputDataLenList:** [output] The output data length of all the slave devices in the ScanList.
  **EPR_List:** [output] The EPR value of all the slave devices in the ScanList.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.36 DNM100_ImportEEPROM

### 4.3.36 DNM100_ImportEEPROM

- **Description:**

  This function will write all specific devices' information in the ScanList to the EEPROM.

- **Syntax:**

```
DWORD DNM100_ImportEEPROM (BYTE BoardNo,
                           WORD ListCount,
                           BYTE *ConnectionTypeList,
                           WORD *InputDataLenList,
                           WORD *OutputDataLenList,
                           WORD *EPR_List)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **ListCount:** [input] The data count of all the information.
  **ConnectionTypeList:** [input] The connection type of all slave devices.

  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

  **InputDataLenList:** [input] The input data length of all slave devices.
  **OutputDataLenList:** [input] The output data length of all slave devices.
  **EPR_List:** [input] The EPR value of all slave devices.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 4.3.37 DNM100_ReadInputData

## 4.3.37 DNM100_ReadInputData

- **Description:**

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

- **Syntax:**

```
DWORD DNM100_ReadInputData (BYTE BoardNo, BYTE DesMACID,
                            BYTE ConType, WORD *IOLen,
                            BYTE *IODATA)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ConType:** [input] The connection type of the remote slave.
  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

  **IOLen:** [output] The length of the I/O data (In byte).
  **IODATA:** [output] The remote I/O data.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.38 DNM100_WriteOutputData

### 4.3.38 DNM100_WriteOutputData

- **Description:**

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

- **Syntax:**

  DWORD DNM100_WriteOutputData (BYTE BoardNo, BYTE DesMACID,
  BYTE ConType, WORD IOLen,
  BYTE *IODATA)

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ConType:** [input] The connection type of the remote slave.
  0 : Explicit connection type
  1 : Poll connection type
  2 : Bit-Strobe connection type
  3 : COS connection type
  4 : Cyclic connection type

  **IOLen:** [Input] The length of the I/O data (In byte).
  **IODATA:** [Input] The remote I/O data.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

### 4.3.39 DNM100_SendExplicitMSG

- **Description:**

  This function is used to send the explicit request command to retrieve or configure the attribute value of the specific device's instance. Before calling this function, you must start the device. After calling this function, you should execute the "DNM100_GetExplicitMSGRespValue" to get the response message returned from remote slave device.

  This old function will be removed in the future. Please use the new function which is "DNM100_SendExplicitMSG_W".

- **Syntax:**

```
DWORD DNM100_SendExplicitMSG (BYTE BoardNo, BYTE DesMACID,
                              BYTE ServiceID, BYTE ClassID,
                              BYTE InstanceID,
                              WORD DataLen, BYTE *DATA)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ServiceID:** [input] The remote slave device's ServiceID.
  **ClassID:** [input] The remote slave device's ClassID(BYTE).
  **InstanceID:** [input] The remote slave device's InstanceID(BYTE).
  **DataLen:** [input] The length of the attribute value (in byte).
  **DATA:** [input] The attribute value that the users want to send.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.40 DNM100_SendExplicitMSG_W

- **Description:**

  This function is used to send the explicit request command to retrieve or configure the attribute value of the specific device's instance. Before calling this function, you must start the device. After calling this function, you should execute the "DNM100_GetExplicitMSGRespValue" to get the response message returned from remote slave device.

  This function could totally complain with the old function which has the same name without the "_W". The user could use this function instead of the "DNM100_SendExplicitMSG".

- **Syntax:**

```
DWORD DNM100_SendExplicitMSG_W (BYTE BoardNo, BYTE DesMACID,
                                BYTE ServiceID, WORD ClassID,
                                WORD InstanceID,
                                WORD DataLen, BYTE *DATA)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **ServiceID:** [input] The remote slave device's ServiceID.
  **ClassID:** [input] The remote slave device's ClassID(WORD).
  **InstanceID:** [input] The remote slave device's InstanceID(WORD).
  **DataLen:** [input] The length of the attribute value (in byte).
  **DATA:** [input] The attribute value that the users want to send.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.41 DNM100_IsExplicitMSGRespOK

- **Description:**

  This function is used to check whether the PISO-DNM100(U) has received the response message or not. After checking the response message, you should execute the “DNM100_GetExplicitMSGRespValue” to get the response message returned from remote slave device.

- **Syntax:**

```
DWORD DNM100_IsExplicitMSGRespOK (BYTE BoardNo,
                                  BYTE DesMACID)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device’s MAC ID (0~63)

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

### 4.3.42 DNM100_GetExplicitMSGRespValue

- **Description:**

  This function is used to get the attribute value of the specific device's instance from the remote slave device. Before calling this function, the users should call DNM100_SendExplicitMSG_W to send request command first.

- **Syntax:**

```
DWORD DNM100_GetExplicitMSGRespValue (BYTE BoardNo,
                                      BYTE DesMACID,
                                      WORD *DataLen ,
                                      BYTE *DATA)
```

- **Parameter:**

  **BoardNo**: [input] PISO-DNM100(U) board number (0~15)
  **DestMACID:** [input] The remote slave device's MAC ID (0~63)
  **DataLen:** [output] The length of the attribute value (in byte).
  **DATA:** [output] The attribute value that returned from the slave device.

- **Return:**

  Please refer to the chapter 4.2 for the function return code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.43 DNM100_ReadbackOutputData

- **Description:**

The function will read the data according with the consumed connection path of the specific MAC ID device via the I/O connection.

- **Syntax:**

```
DWORD DNM100_ReadbackOutputData (BYTE BoardNo,
                                 BYTE DesMACID,
                                 BYTE ConType,
                                 WORD *IOLen,
                                 BYTE *IODATA)
```

- **Parameter:**

**BoardNo**: [input] PISO-DNM100(U) board number (0~15)
**DestMACID:** [input] The remote slave device's MAC ID (0~63)
**ConType:** [input] The connection type of the remote slave.

0 : Explicit connection type
1 : Poll connection type
2 : Bit-Strobe connection type
3 : COS connection type
4 : Cyclic connection type

**IOLen:** [output] The length of the I/O data (In byte).
**IODATA:** [output] The remote I/O data.

- **Return:**

Please refer to the chapter 4.2 for the function return code.

### 4.3.44 DNM100_ReadInputArea (Advanced Option)

- **Description:**

  This function read the bulk data from the input area memory. Users can use this function to read all the input data quickly. It provides effective method to read the bulk data in one command. If the users need to read the input data of all slave devices, they need to call the "DNM100_ReadInputData" for each slave device in general. By calling the advanced function "DNM100_ReadInputArea", the whole input data would be read once in this function call. Please refer to the section 3.7 for more description.

- **Syntax:**

```
DWORD DNM100_ReadInputArea (BYTE BoardNo,
                            WORD Offset,
                            WORD DataLen,
                            BYTE *DataArray)
```

- **Parameter:**

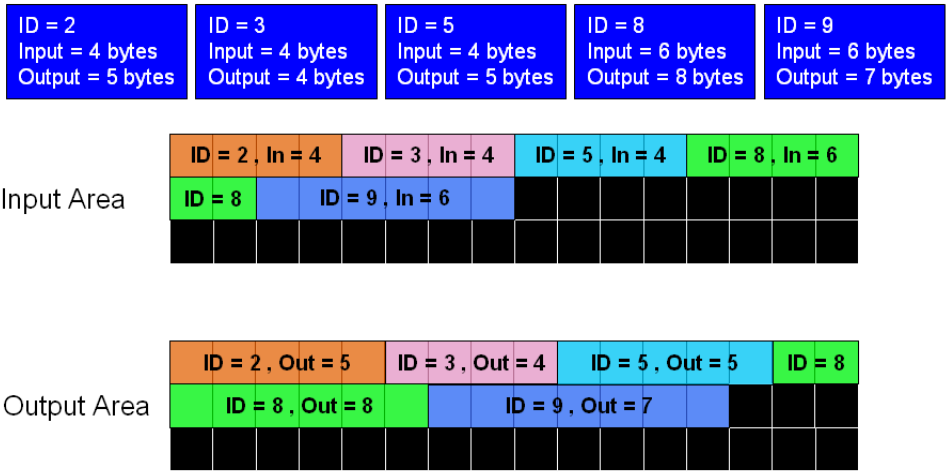
  **BoardNo:** [input] PISO-DNM100(U) board number (0~15).
  **Offset:** [input] The offset of the input area memory.
  **DataLen:** [input] The length in byte which the users want to read.
  **DataArray:** [output] The data pointer of the obtained data from the input area memory.

- **Return:**

  Please refer to the section 4.2 for the return code.

- **Example:**

Example 1 :




BYTE BoardNo = 0; //Assume that the PISO-DNM100's board No. is 0.
WORD Offset = 0;
WORD DataLen = 0;
BYTE DataArray[512] = {0};

//Read whole input data of all slave devices.
Offset = 0; //Read the data from the beginning of the input area.
DataLen = 4 + 4 + 4 + 6 + 6; //The sum of all slave's input length.
DNM100_ReadInputArea (BoardNo, Offset, DataLen, DataArray);
DataArray = [The whole input data of all slave devices];

//Read input data of the slave device which ID = 5.
Offset = 4 + 4; //Read the data from the beginning of the ID = 5.
DataLen = 4; //The input length of the ID = 5.
DNM100_ReadInputArea (BoardNo, Offset, DataLen, DataArray);
DataArray = [The input data of the slave device which ID = 5];

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.45 DNM100_WriteOutputArea (Advanced Option)

- **Description:**

This function can write the bulk data to the output area memory. The DeviceNet slave will change the output status of the DeviceNet slaves according to the data located in the output area memory. Users can use this function to write a large amount of the output data. It provides effective method to change bulk data in one command. If the users need to write the output data of all slave devices, they need to call the "DNM100_WriteOutputData" for each slave device in general. By calling the advanced function "DNM100_WriteOutputArea", the whole output data would be wrote once in this function call. Please refer to the section 3.7 for more description.

- **Syntax:**

```
DWORD DNM100_WriteOutputArea (BYTE BoardNo,
                              WORD Offset,
                              WORD DataLen,
                              BYTE *DataArray)
```

- **Parameter:**

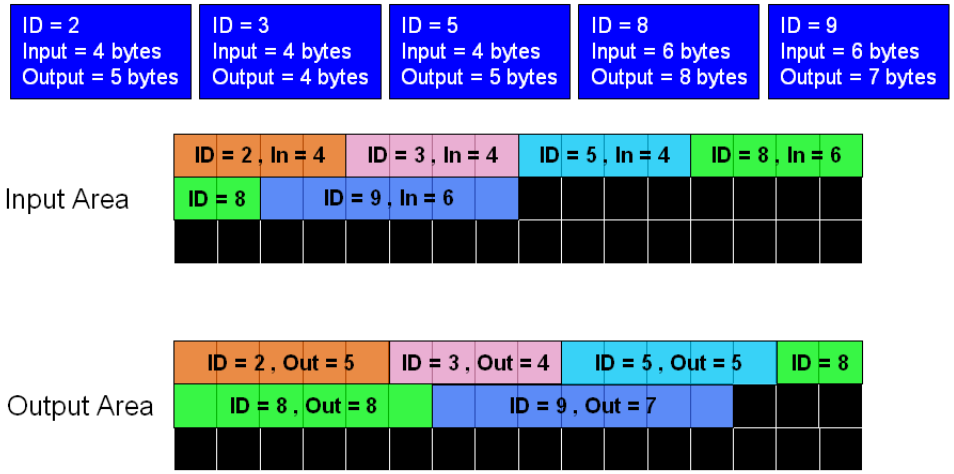
**BoardNo:** [input] PISO-DNM100(U) board number (0~15).
**Offset:** [input] The offset of the input area memory.
**DataLen:** [input] The length in byte which the users want to write.
**DataArray:** [input] The data pointer pointed the data written to the output data area.

- **Return:**

Please refer to the section 4.2 for the return code.

- **Example:**

Example 1 :




BYTE BoardNo = 0; //Assume that the PISO-DNM100's board No. is 0.
WORD Offset = 0;
WORD DataLen = 0;
BYTE DataArray[512] = {0};

//Write the whole output data of all slave devices.
Offset = 0; //Write the data from the beginning of the output area.
DataLen = 5 + 4 + 5 + 8 + 7; //The sum of all slave's output length.
DataArray = [The whole output data];
DNM100_WriteOutputArea (BoardNo, Offset, DataLen, DataArray);

//Write the output data of the slave device which ID = 5.
Offset = 5 + 4; //Write the data from the beginning of the ID = 5.
DataLen = 5; //The output length of the ID = 5.
DataArray = [The output data of the ID = 5];
DNM100_WriteOutputArea (BoardNo, Offset, DataLen, DataArray);

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

### 4.3.46 DNM100_ReadbackOutputArea (Advanced Option)

- **Description:**

  This function reads the bulk data from the output area memory. It dose not change the output data located in the output area memory. Users can use this function to read back a large amount of the output data, but the data may be different with the real output statuses of the DeviceNet slaves. It provides effective method to read bulk data in one command. If the users need to read back the output data of all slave devices, they need to call the “DNM100_ReadbackOutputData” for each slave device in general. By calling the advanced function “DNM100_ReadbackOutputArea”, the whole output data would be wrote once in this function call. Please refer to the section 3.7 for more description.

- **Syntax:**

```
DWORD DNM100_ReadbackOutputArea (BYTE BoardNo,
                                 WORD Offset,
                                 WORD DataLen,
                                 BYTE *DataArray)
```

- **Parameter:**

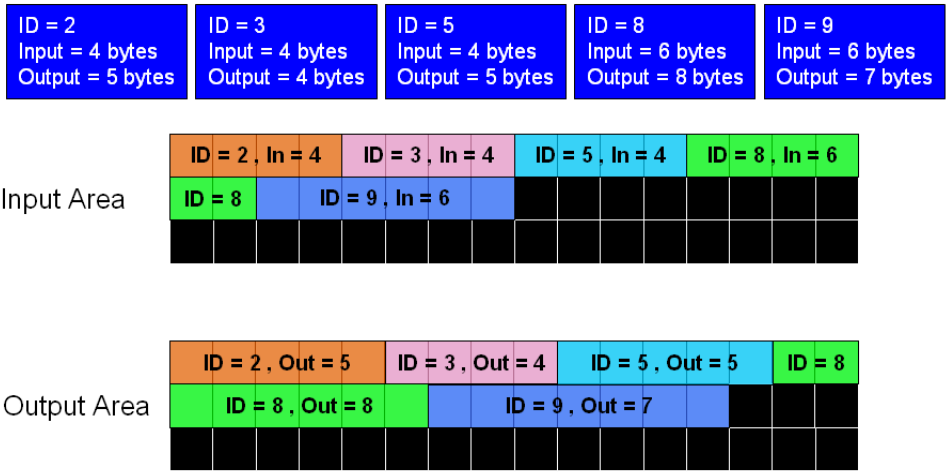
  **BoardNo:** [input] PISO-DNM100(U) board number (0~15).
  **Offset:** [input] The offset of the output area memory.
  **DataLen:** [input] The length in byte which the users want to read.
  **DataArray:** [output] The data pointer pointed the observed data form the output area memory.

- **Return:**

  Please refer to the section 4.2 for the return code.

- **Example:**

Example 1 :




BYTE BoardNo = 0; //Assume that the PISO-DNM100's board No. is 0.
WORD Offset = 0;
WORD DataLen = 0;
BYTE DataArray[512] = {0};

//Read back the whole output data of all slave devices.
Offset = 0; //Read back the data from the beginning of the output area.
DataLen = 5 + 4 + 5 + 8 + 7; //The sum of all slave's output length.
DNM100_ReadbackOutputArea (cSlot, Offset, DataLen, DataArray);
DataArray = [The whole output data];

//Read back the output data of the slave device which ID = 5.
Offset = 5 + 4; //Read back the data from the beginning of the ID = 5.
DataLen = 5; //The output length of the ID = 5.
DNM100_WriteOutputArea (cSlot, Offset, DataLen, DataArray);
DataArray = [The output data of the ID = 5];

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 5. Demo Programs for Windows

All of demo programs will not work normally if PISO-DNM100(U) 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. After completing the driver installation, the related demo programs, development library and declaration header files for different development environments are installed in the system as follows.

The PISO-DNM100(U)'s root directory is C:\ICPDAS\PISO-DNM100

| | |
|---|---|
| \|--\DLL | The DLL driver for the user's application |
| \|--\Driver | The window driver of the PISO-DNM100(U) |
| \|--\Manual | The user manual of the PISO-DNM100(U) |
| \|--\Demo | Demo program |
| \|--\Demo\BCB 6 | Demos for Borland C++ Builder 6 |
| \|--\Demo\VC++ 6 | Demos for Visual C++ 6 |

## 5.1 A brief introduction to the demo programs

VC_Demo1 : Demonstrate the basic functions to communicate with the remote slave device. The demo program will lead you step by step to complete the setting and communication.

VC_Demo2 : Demonstrate the scan function to scan all the remote slave devices in the same DeviceNet network. The demo program will show you all the slave devices and their I/O connection type.

BCB_Demo1 : Demonstrate the scan function and add/remove function to configure the information of the remote slave device.

BCB_Demo2 : Demonstrate the I/O functions to access the I/O data of the remote slave device. The demo program will show you the input value of the remote device and let you send out the data to the output pins of the remote slave devices.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.2 Wire Connection of the CAN bus

Before starting the demos, the users should have at least one slave device. Here show the users how to connect the master and slave devices by CAN bus. The slave devices should be connected to form the serial type which is shown as Figure 5.1

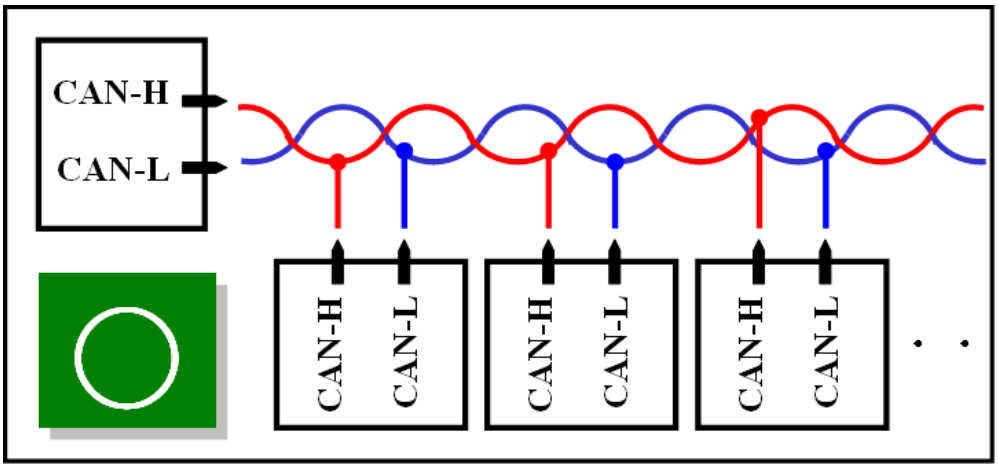



Figure 5.1 Correct wire connection

The following wire connection is **wrong** which is shown as Figure 5.2

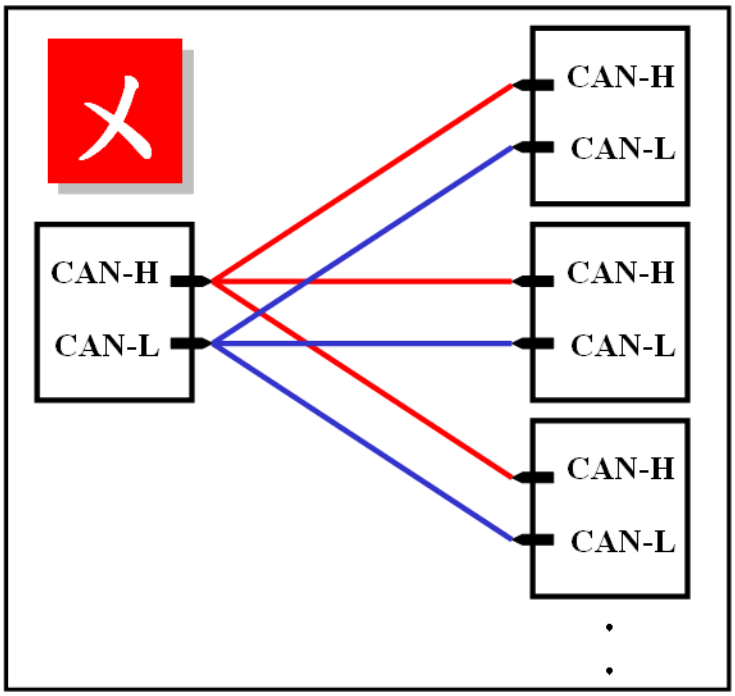



Figure 5.2 Wrong wire connection

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.3 VC_Demo1 Introduction

VC_Demo1 is the example used for starting the DeviceNet communication. The screen shoot is shown as Figure 5.3. This demo program is designed to communicate with slave devices step by step. This program will read the input value of the slave device when the POLL connection has been established. Before exercising this demo, the users should have at least one DeviceNet slave device which has input channels (AI or DI) and finish the wire connection between the Master and slave device. (See Figure 5.1)

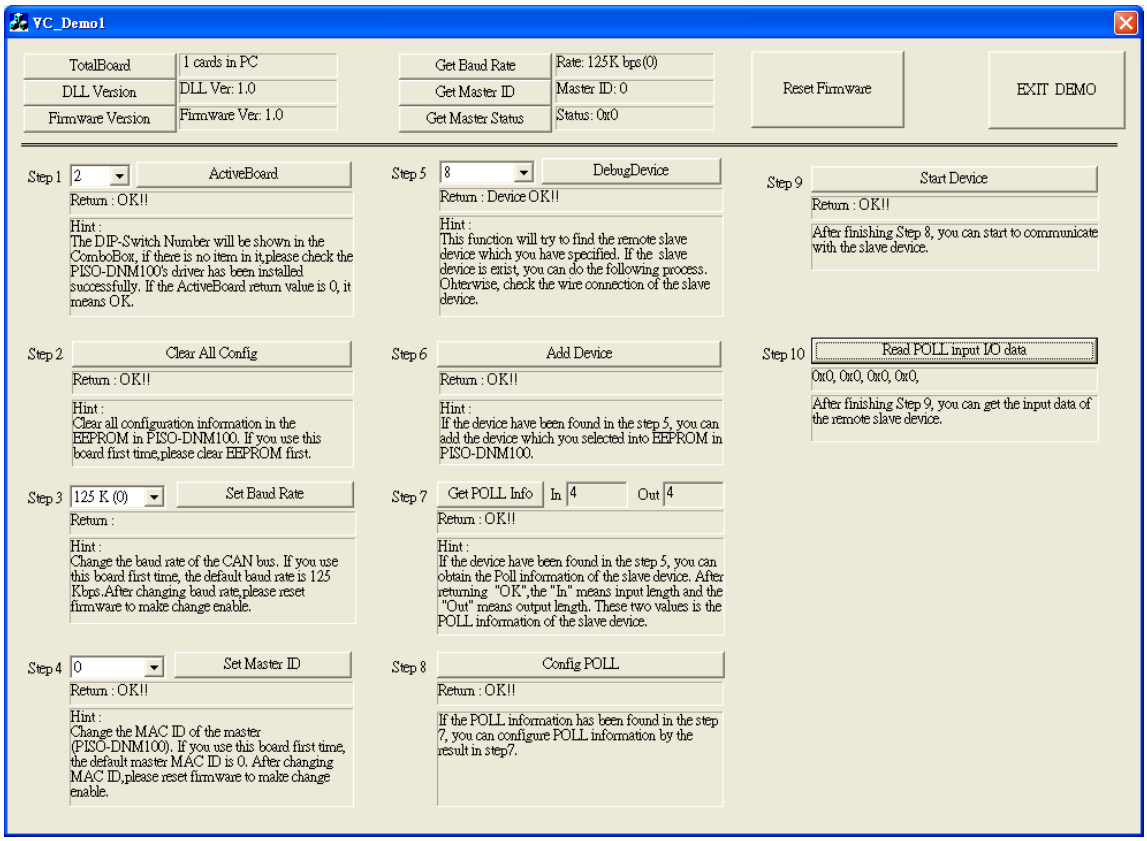



Figure 5.3 the screen shoot of VC_Demo1

After running the program, the users will see the “TotalBoards” information on the left and up corner of the screen. This function determinates how many PISO-DNM100(U) in your PC automatically. If the program doesn’t find any board, the users should check that the windows driver has been installed successfully. Otherwise, if it has found at lease one board, the users can continue exercising the demo program.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 1 : ActiveBoard

Before performing other buttons, the “ActiveBoard” button should be clicked firstly. The DIP-Switch on the board means the ID of this board. The users should make sure that every board’s ID in your PC is unique. The dropdown list will show the board’s ID which the users have selected. After clicking the button, the return code will be 0. Otherwise, please check the windows driver has been installed successfully.

Step 2 : Clear All Config

To avoid unknown configuration in the PISO-DNM100(U), the users can click this button to clear all configuration in the board.

Step 3 : Set Baud Rate

The default baud rate of DeviceNet is 125Kbps. If the users want to change the value, they can select the correct value then click the button. After changing the baud rate, the users should reset the firmware in the PISO-DNM100(U) by clicking the “ResetFirmware” button. It needs to wait for 1 or 2 seconds to the next step.

Step 4 : Set Master ID

The default Master’s MAC ID is 0. If the users want to change the value, you can select the correct value then click the button. After changing the Master’s ID, the uses should reset the firmware in the PISO-DNM100(U) by clicking the “ResetFirmware” button. It needs to wait for 1 or 2 seconds to the next step.

Step 5 : Debug Device

Before performing this function, the users should set the MAC ID of the slave device and turn on it. In the demo, the users can select the MAC ID of the slave device then click “Debug Device” button. This function will try to find the appointed remote slave device waiting for 5 seconds, if the slave device exists in the network, the return value will be 0. Otherwise, the device doesn’t exist without response. The users can go to next step until the problem of the slave device has been solved.

Step 6 : Add Device

Before performing this function, the users should use step 5 to find an exist device. This function can add the device’s information found at step 5 into

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

EEPROM of the PISO-DNM100(U). If it is successful, the return value will be 0.

Step 7 : Get POLL Info

This function is to obtain the device's POLL information. After the slave device responses the data, it would show in the "In" and "Out". "In" means the input length of the slave device. "Out" means the output length of the slave device. If the slave device responses successfully, the return value will be 0.

Step 8 : Config POLL

If the step 7 is successful, the users can perform "Config POLL" button. This function is to add the device's POLL information into EEPROM in the PISO-DNM100(U). If it is successful, the return value will be 0.

Step 9 : Start Device

If the step 8 is successful, the users can perform "Start Device" button. This function would communicate with the slave device which the users have configured in the previous steps. If it is successful, the return value will be 0.

Step 10 : Read POLL Input I/O Data

If the master is communicating with the slave device successfully, the users can read the input I/O data from the slave device in this step. This function would obtain the input I/O data and show them in byte (8-bits).

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.4 VC_Demo2 Introduction

VC_Demo2 is the example used for scanning the DeviceNet slave devices in the network. The screen shoot is shown as Figure 5.4. This demo program is designed to operate the master step by step. This program will show the information of all the slave devices in the network. Before exercising this demo, the users should have at least one DeviceNet slave device and finish the wire connection between the Master and slave device. (See Figure 5.1)

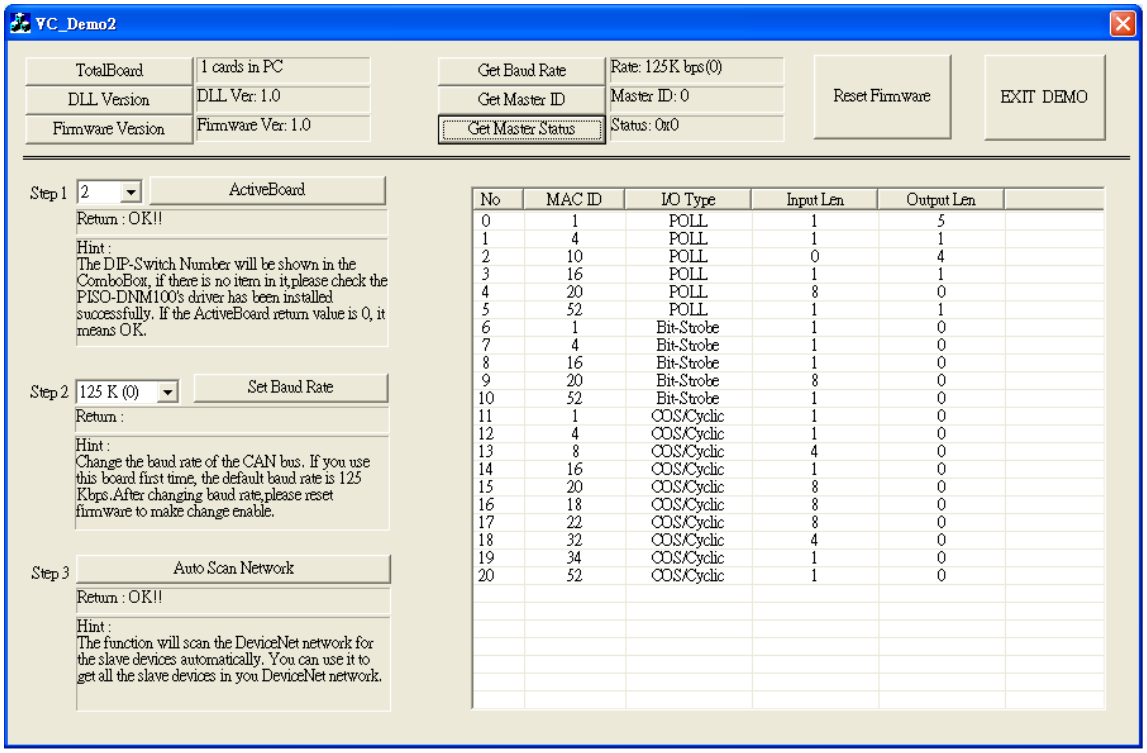



Figure 5.4 The screen shoot of VC_Demo2

After running the program, the users will see the “TotalBoards” information on the left and up corner of the screen. This function determinates how many PISO-DNM100(U) in your PC automatically. If the program doesn’t find any board, the users should check that the windows driver has been installed successfully. Otherwise, if it has found at lease one board, the users can continue exercising the demo program.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Step 1 : ActiveModule

Before performing other buttons, the “ActiveBoard” button should be clicked firstly. The DIP-Switch on the board means the ID of this board. The users should make sure that every board’s ID in your PC is unique. The dropdown list will show the board’s ID which the users have selected. After clicking the button, the return code will be 0. Otherwise, please check the windows driver has been installed successfully.

Step 2 : Set Baud Rate

The default baud rate of DeviceNet is 125Kbps. If the users want to change the value, they can select the correct value then click the button. After changing the baud rate, the users should reset the firmware in the PISO-DNM100(U) by clicking the “ResetFirmware” button. It needs to wait for 1 or 2 seconds to the next step.

Step 3 : Auto Scan Network

Before performing this function, the users should set the MAC ID and the baud rate of the slave device and turn on it. In the demo, the user can click “Auto Scan Network” button to obtain all the I/O information of all slave devices in the network. The scan procedure needs about 30 seconds. The users will see the entire slave device in the network and their I/O information in the list table.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.5 BCB_Demo1 Introduction

BCB_Demo1 is the example used for scanning the DeviceNet slave devices in the network. The screen shoot is shown as Figure 5.5. This program will show the information of all the slave devices. This demo is similar to VC_Demo2. Before exercising this demo, the users should have at least one DeviceNet slave device and finish the wire connection between the Master and slave device. (See Figure 5.1) The users can configure the slave device by the scanning information. This demo can be a tool to add or remove the configuration of the slave device.

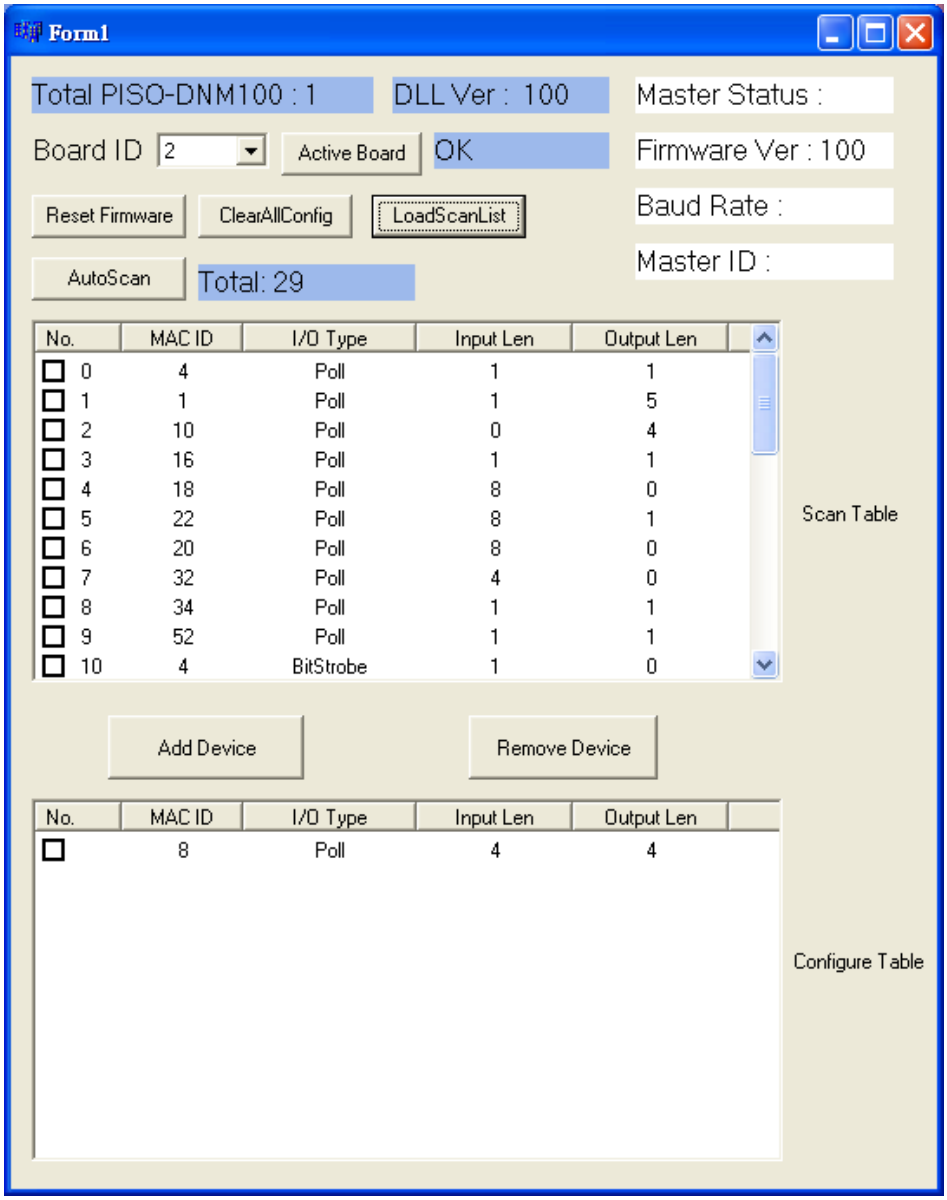



Figure 5.5 The screen shoot of BCB_Demo1

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

After running the program, the users would see the “Total PISO-DNM100(U) : x” information on the left and up corner of the screen. This function determinates how many PISO-DNM100(U) in your PC automatically. If the program doesn’t find any board, the users should check that the windows driver has been installed successfully. Otherwise, if it has found at lease one module, the users can continue exercising the demo program.

Step 1 : Active Module

Before performing other buttons, the “ActiveBoard” button should be clicked firstly. The DIP-Switch on the board means the ID of this board. The users should make sure that every board’s ID in your PC is unique. The dropdown list will show the board’s ID which the users have selected. After clicking the button, the return code will be 0. Otherwise, please check the windows driver has been installed successfully.

Step 2 : Auto Scan

Before performing this function, the users should set the MAC ID and the baud rate of the slave device and turn on it. In the demo, the user can click “Auto Scan Network” button to obtain all the I/O information of all slave devices in the network. The scan procedure needs about 30 seconds. The users will see the entire slave device in the network and their I/O information in the “Scan Table”.

Step 3 : Add Device

This function is to add the device’s information into EEPROM in the PISO-DNM100(U). The users can check the item which you want to add. After checking what you want, push “Add Device” button to add the information into the EEPROM. If it is successful, the items which you selected will be shown in the “Configure Table”.

Step 4 : LoadScanList

This function is to obtain the device’s information from EEPROM in the PISO-DNM100(U). The users can check the information in the EEPROM. After performing the function, the information will be shown in the “Configure Table”.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 5.6 BCB_Demo2 Introduction

BCB_Demo2 is the extension of the BCB_Demo1. The screen shoot is shown as Figure 5.6. This program can read the input data and write the output data in every second. This demo is similar to BCB_Demo1. We just introduce the extension part. Before exercising this demo, the users should have at least one DeviceNet slave device and finish the wire connection between the Master and slave device. (See Figure 5.1) The users can configure the slave device by the scanning information. This demo can be a tool to add or remove the configuration of the slave device. Additionally, the users can operate the I/O data form the remote slave devices.

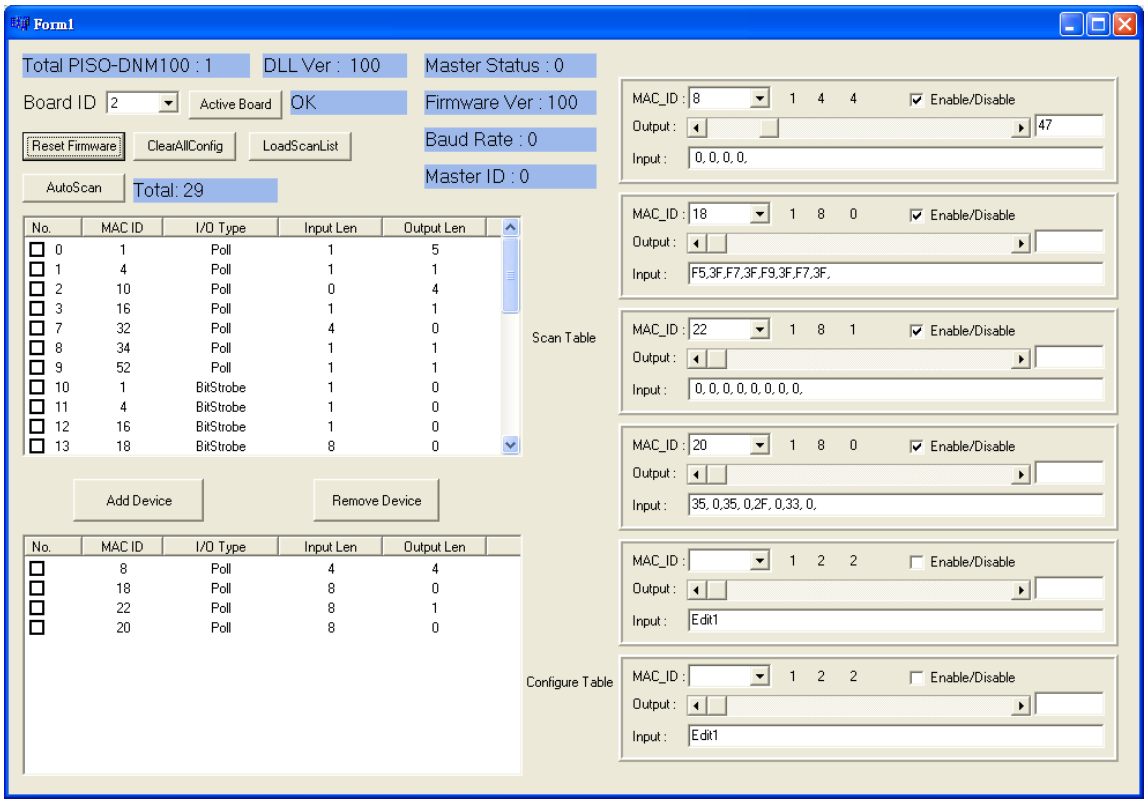



Figure 5.6 The screen shoot of BCB_Demo2

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

If the users want to know how to configure the slave device information into EEPROM in the PISO-DNM100(U), please refer to section 5.5.

If “Active Board” is OK, the users can click “LoadScanList” button. The configuration information will be shown in “Configure Table”. At the same time, the MAC IDs also are shown on the right side of the screen. The users can check the “Enable/Disable” box to enable or disable the read and write the I/O data. The “Output” scroll bar presents the output value which will be written to the output channel of the slave device. If the scroll value is 0x23, every byte of the slave device’s output is 0x23. The users can find out the change of the output channel easily. The “Input” field presents the input value from the input channel of the slave device.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

# 6. LabVIEW Driver Introduction

## 6.1 Software Installation

The PISO-DNM100(U) LabVIEW 8.x driver is the PISO-DNM100(U) function reference for the DeviceNet master device used on LabVIEW 8.x environment on Windows. Before users use this driver to develop the machine control system, the PISO-DNM100(U) driver must be installed first, because the PISO-DNM100(U) LabVIEW 8.x driver needs to call the function of it. The driver architecture is shown as Figure 6.1.

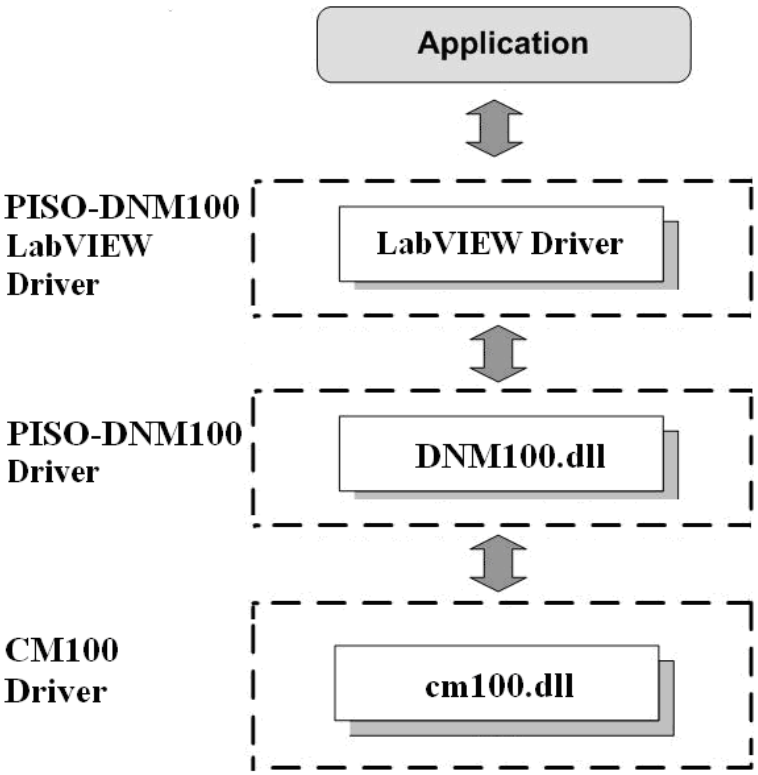



Figure 6.1 Driver concept of PISO-DNM100(U) LabVIEW driver

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

After completing the LabVIEW driver installation, the directory of PISO-DNM100(U) LabVIEW Driver is C:\ICPDAS\PISO-DNM100\LabVIEW.

|--\Driver LabVIEW driver

|--\Demo LabVIEW demo program

PISO-DNM100(U) LabVIEW Function palette is showed as below.

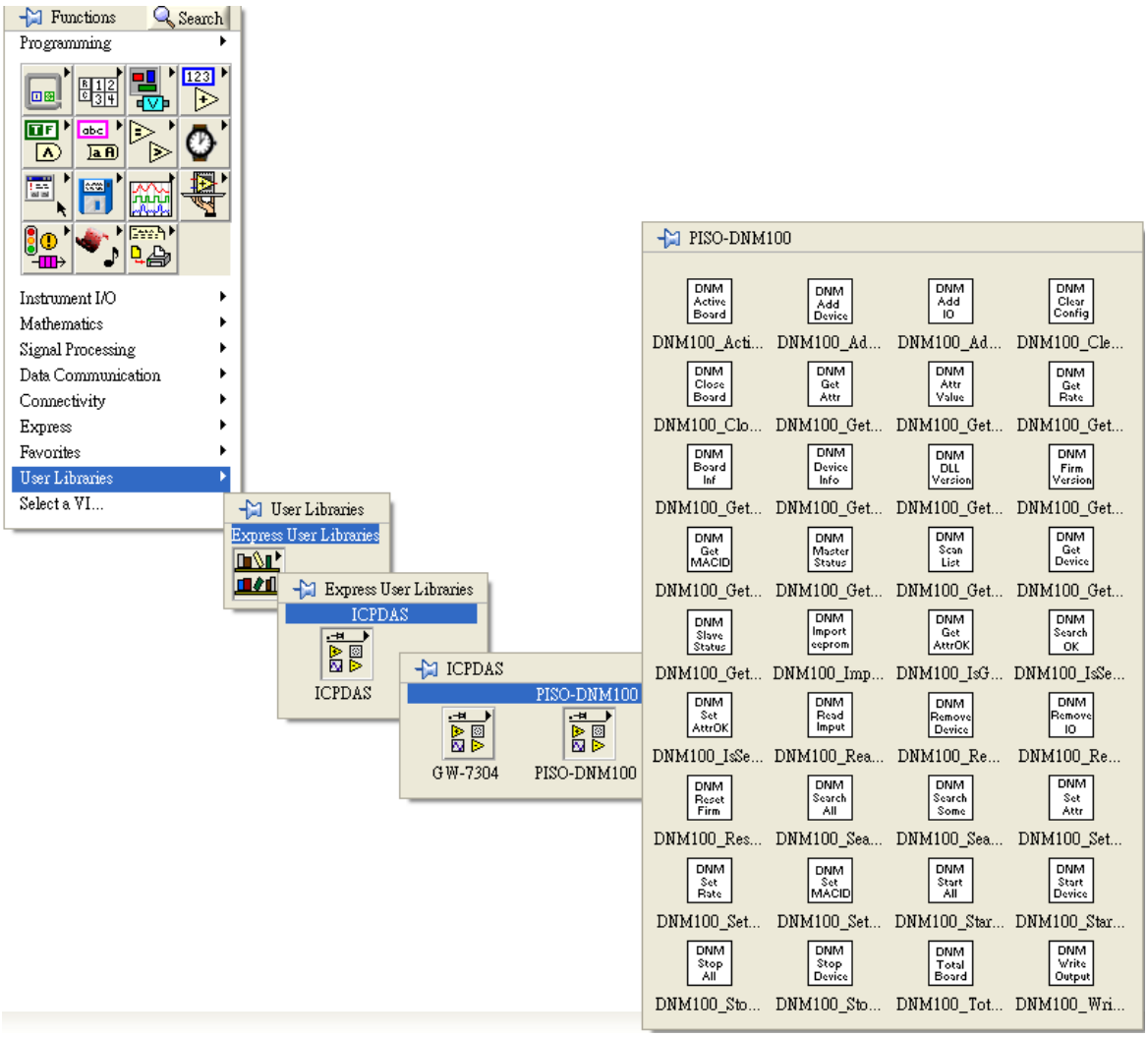



Figure 6.2 PISO-DNM100(U) LabVIEW Function palette

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 6.2 Function description

Every function provides each VI for user to use in the LabVIEW environment. All the functions provided by the DNM100.dll are listed in the following table.

Table 6.2.1 Board Functions

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| 1 |  | DNM100_GetBoardInf | Get the driver information of the PISO-DNM100(U) |
| 2 |  | DNM100_TotalDNM100Board | Get total PISO-DNM100(U) boards in the PC |
| 3 |  | DNM100_ActiveBoard | Enable the driver of the PISO-DNM100(U) |
| 4 |  | DNM100_CloseBoard | Close driver of the PISO-DNM100(U) |
| 5 |  | DNM100_GetDLLVersion | Get the DLL version of the DNM100.DLL |

Table 6.2.2 Firmware Functions

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| 1 |  | DNM100_GetFirmwareVersion | Get the version of the firmware inside the PISO-DNM100(U) board |
| 2 |  | DNM100_ResetFirmware | Reset the firmware in the PISO-DNM100(U) board |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 6.2.3 Operating Functions 1/3

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| 1 |  | DNM100_SetMasterMACID | Set the MAC ID of the PISO-DNM100(U) board (DeviceNet Master's MAC ID) |
| 2 |  | DNM100_GetMasterMACID | Get the MAC ID of the PISO-DNM100(U) board (DeviceNet Master's MAC ID) |
| 3 |  | DNM100_GetBaudRate | Get the baud rate of the CAN bus |

Table 6.2.4 Operating Functions 2/3

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| 17 |  | DNM100_GetAttributeValue | Get the attribute value of the DNM100_GetAttribute |
| 18 |  | DNM100_SetAttribute | Send the set attribute command to the slave device. |
| 19 |  | DNM100_IsSetAttributeOK | Check whether the slave has replied for the setting command or not. |
| 20 |  | DNM100_GetDeviceInfoFromScanList | Get specific slave device's I/O information form the Scan List within the PISO-DNM100(U) board. |
| 21 |  | DNM100_GetScanList | Get the I/O information of all slave devices form the Scan List within the PISO-DNM100(U) board. |
| 22 |  | DNM100_ImportEEPROM | Write the I/O information of all slave devices into the EEPROM within the PISO-DNM100(U) board. |
| 23 |  | DNM100_ClearAllConfig | Clear all configurations in the EEPROM within the PISO-DNM100(U) board. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 6.2.5 Operating Functions 3/3

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| **4** |  | DNM100_SetBaudRate | Set the baud rate of the CAN bus |
| **5** |  | DNM100_GetMasterStatus | Get the status of the PISO-DNM100(U) board (DeviceNet Master's status) at present |
| **6** |  | DNM100_GetSlaveStatus | Get the slave device's status. |
| **7** |  | DNM100_StartDevice | PISO-DNM100(U) will start to communicate with the specific slave device |
| **8** |  | DNM100_StopDevice | PISO-DNM100(U) will stop to communicate with the specific slave device |
| **9** |  | DNM100_StartAllDevice | PISO-DNM100(U) will start to communicate with all slave devices |
| **10** |  | DNM100_StopAllDevice | PISO-DNM100(U) will stop to communicate with all slave devices |
| **11** |  | DNM100_AddDevice | Add the specific slave device's information into the PISO-DNM100(U) board (DeviceNet Master) |
| **12** |  | DNM100_RemoveDevice | Remove the specific slave device's information from the PISO-DNM100(U) board (DeviceNet Master) |
| **13** |  | DNM100_AddIOConnection | Add I/O information of the specific slave device into the PISO-DNM100(U) board (DeviceNet Master) |
| **14** |  | DNM100_RemoveIOConnection | Remove specific slave device's I/O information from the PISO-DNM100(U) board (DeviceNet Master) |
| **15** |  | DNM100_GetAttribute | Send the get attribute command to the slave device. |
| **16** |  | DNM100_IsGetAttributeOK | Check whether the slave has replied for the getting command or not. |

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

Table 6.2.6 Searching Functions

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| 1 |  | DNM100_SearchAllDevices | PISO-DNM100(U) will search the DeviceNet network to find out the I/O information of all slave devices. |
| 2 |  | DNM100_SearchSpecificDevice | PISO-DNM100(U) will search the DeviceNet network to find out the I/O information of specific slave devices. |
| 3 |  | DNM100_IsSearchOK | Check whether the PISO-DNM100(U) has searched completely or not. |
| 4 |  | DNM100_GetSearchedDevices | Get the result of the searching command and retrieve the slave's I/O information. |

Table 6.2.7 I/O Functions

| No. | VI ICON | Function Name | Description |
|---|---|---|---|
| 1 |  | DNM100_ReadInputData | Read the input data via I/O connection like Poll, Strobe, COS, Cyclic. |
| 2 |  | DNM100_WriteOutputData | Write the output data via I/O connection like Poll, COS, Cyclic. The Strobe doesn't support this operation. |

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

## 6.3 LabVIEW Demo Introduction

The LabVIEW Demo is similar to the VC_Demo1. The screen shoot is shown as Figure 6.3.1. This demo program is designed to communicate with slave device step by step. All operating steps and detail descriptions can see in the section 5.3.




Figure 6.3.1 The screen shoot of LabVIEW Demo