DELTA ELECTRONICS, INC.

CMC-DN01

DeviceNet Slave Communication Card

Operation Manual

http://www.delta.com.tw/industrialautomation

2011-12-26-A

⚠ Warning

✓ This operation manual provides introduction on the functions, specifications, installation, basic operation and settings for CMC-DN01 and the network protocol.

✓ This is an OPEN TYPE device and therefore should be installed in an enclosure free of airborne dust, humidity, electric shock and vibration. The enclosure should prevent non-maintenance staff from operating the device (e.g. keys or specific tools are required to open the enclosure) in case danger and damage on the device may occur. DO NOT touch any terminal when the power is switched on.

✓ Please read this operation manual thoroughly and follow the instructions in case damages on the device or injuries on the operation staff occur.

🕮 Table of Content

1 Introduction to CMC-DN01

1. Thank you for choosing Delta CMC-DN01 communication card. To ensure correct installation and operation of the product, please read this operation manual carefully before using it.
2. CMC-DN01 is the DeviceNet communication card able to conduct remote setups and communications through the DeviceNet bus.
3. CMC-DN01 communication card connects Delta VFD-C2000 series AC motor drive to the DeviceNet network.

1.1 Features

- Based on the high-speed communication interface of Delta HSSP protocol, able to conduct immediate control of the AC motor drive.
- Supports Group 2 only connection and polled I/O data exchange.
- For I/O mapping, supports max. 32 words of input and 32 words of output.
- Supports EDS file configuration in DeviceNet configuration software.
- Supports all baud rates on the DeviceNet bus: 125, 250, 500 kbps and the extendable serial baud rate mode.
- Node addresses and serial baud rates can be set up directly on the AC motor drive.
- The power is automatically supplied by the AC motor drive.

1.2 Specifications

■ DeviceNet Port

Interface	5-PIN open removable connector. Of 5.08mm PIN interval
Transmission method	CAN
Transmission cable	Shielded twisted pair cable (with 2 power cables)
Baud rates	125, 250, 500 kbps and extendable serial baud rate modes
Network protocol	DeviceNet protocol

■ AC Motor Drive Port

Interface	50-PIN communication terminal
Transmission method	SPI communication
Terminal functions	1. Communication with CMC-DN01 2. Supplying power to CMC-DN01
Communication protocol	Delta HSSP protocol

■ Environment

Interference immunity	ESD (IEC 61800-5-1, IEC 6100-4-2) EFT (IEC 61800-5-1, IEC 6100-4-4) Surge Teat (IEC 61800-5-1, IEC 6100-4-5) Conducted Susceptibility Test (IEC 61800-5-1,IEC 6100-4-6)
Operation/storage	Operation: -10 to 50°C (temperature), 90% (humidity) Storage: -25 to 70°C (temperature), 95% (humidity)

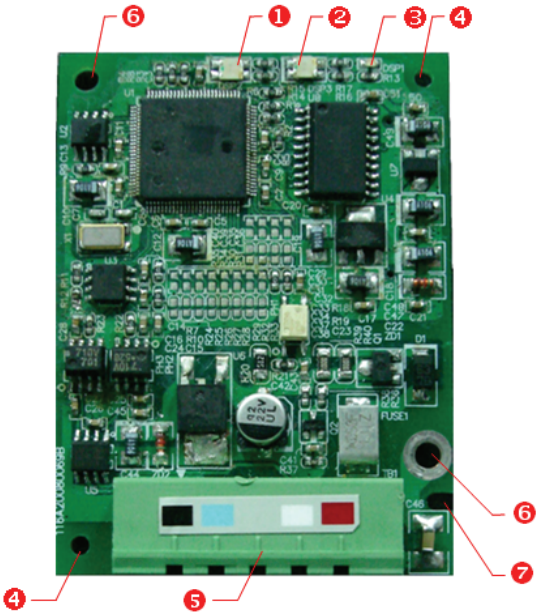
Shock/vibration resistance	International standards: IEC 61800-5-1, IEC 60068-2-6/IEC 61800-5-1, IEC 60068-2-27

■ Electrical Specifications

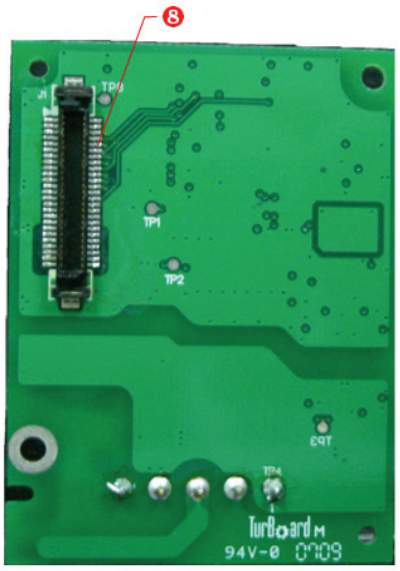
Power supply voltage	5 VDC (supplied by AC motor drive)
Insulation voltage	500 VDC
Communication wire power consumption	0.85 W
Power consumption	1 W
Weight	23g

2 Product Profile and Outline

2.1 Parts



Front View



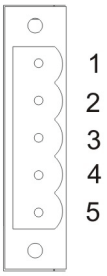
Rear View

1	NS indicator	2	MS indicator	3	POWER indicator
4	Positioning hole	5	DeviceNet port	6	Screw fixing hole
7	Fool-proof groove	8	AC motor drive port		

2.2 DeviceNet Port

The port is used to connect the card to the DeviceNet network. See below for the PIN definitions.

PIN	PIN name	Color	Definition
1	V+	Red	DC24V
2	CAN_H	White	Signal+
3	S	--	Earth
4	CAN_L	Blue	Signal-
5	V-	Black	0 V

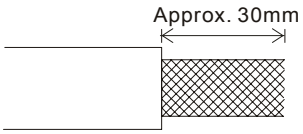


3 Basic Operation

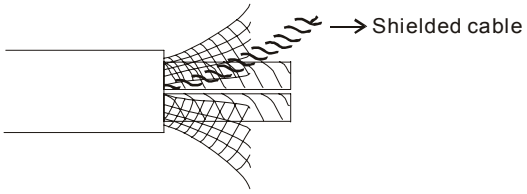
3.1 Installation

■ How to install

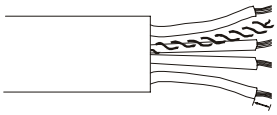
① Use an efficient tool to peel the communication cable for approx. 30mm. DO NOT damage the shielded cable during the peeling.



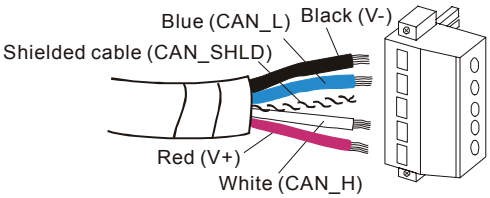
② Peel off the metallic shielded net and foil, and you will see 2 power cables (red and black), 2 signal cables (blue and white) and 1 shielded cable.



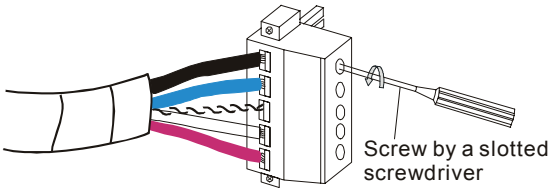
③ Peel off the exterior metallic shielded net, foil and the plastic cover of the power cable and signal cable in proper length.



④ Insert the peeled communication cables into the holes in the connector in correct order.

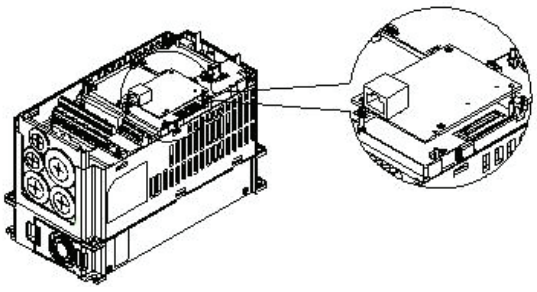


⑤ Tighten the screws on the connector by a slotted screwdriver and fix the communication cables in the holes in the connector.

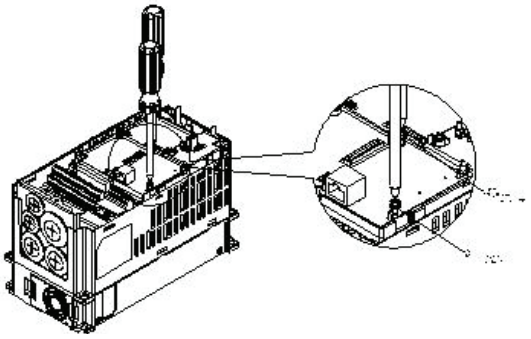


⑥ Install CMC-DN01 on the VFD-C2000 series AC motor drive:

1. Switch off the power supply of VFD-C2000.
2. Open the cover on top of VFD-C2000.
3. Place the insulation spacer into the positioning pin and aim the two holes on the PCB at the positioning pin. Press the pin to clip the holes with the PCB (Figure 1).
4. Screw up after the PCB is clipped with the holes (Figure 2).

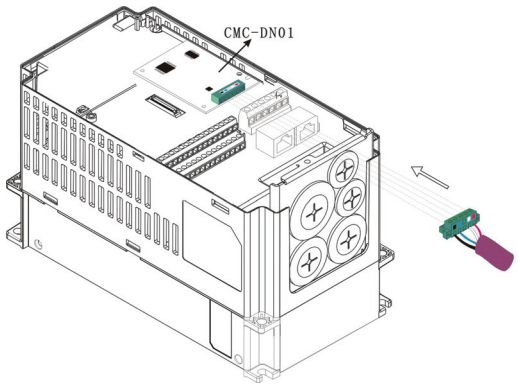


[Figure 1]



[Figure 2]

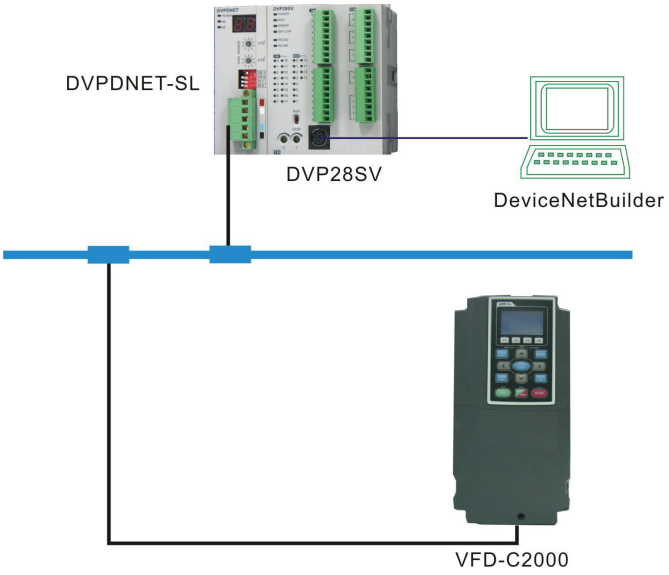
⑦ Connect to the DeviceNet port: Insert the DeviceNet connector to the DeviceNet port on CMC-DN01 (Figure 3).



[Figure 3]

■ Constructing a DeviceNet network

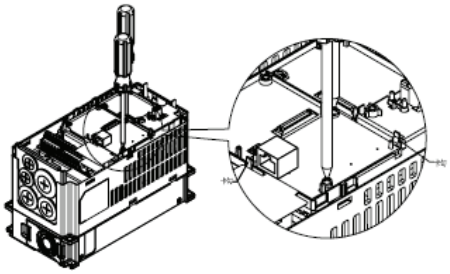
(Figure 4) The DVPDNET-SL module is the DeviceNet master. CMC-DN01 and the VFD-C2000 series AC motor drive construct the DeviceNet slave. Use the software DeviceNet Builder to configure the DeviceNet network.



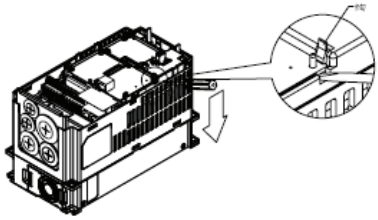
[Figure 4]

■ Disconnecting CMC-DN01 from VFD-C2000

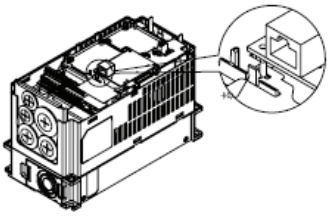
① Remove the two screws (Figure 5).
② Twist open the card clip, insert the slotted screwdriver to the hollow and prize the PCB off the card clip (Figure 6).
③ Twist open the other card clip to remove the PCB (Figure 7).



[Figure 5]



[Figure 6]



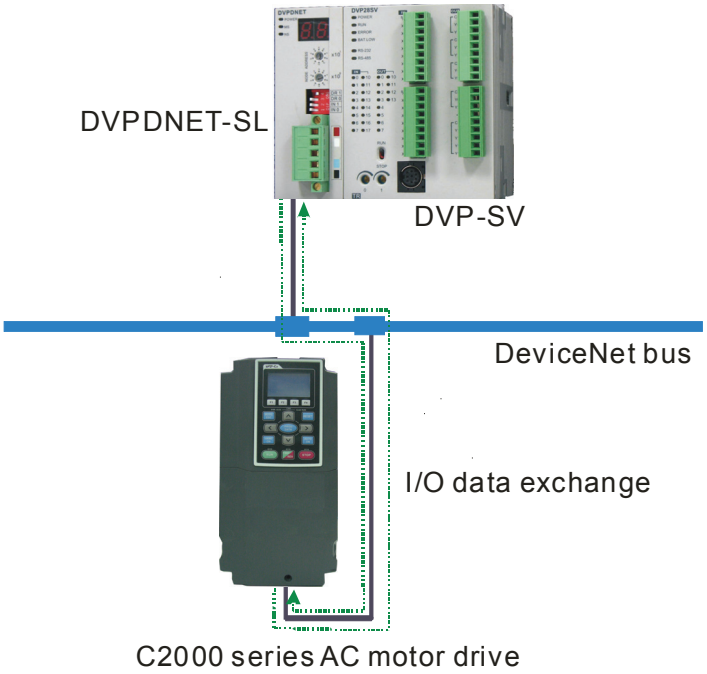
[Figure 7]

4 VFD-C2000 Series AC Motor Drive and DeviceNet Master

In this section, we will explain the relation between Delta VFD-C2000 series AC motor drive and the

DVPDNET-SL module and how they exchange data with each other. (Figure 4.1.1) The DVPDNET-SL module is the DeviceNet master, and the VFD-C2000 series AC motor drive is connected to the DeviceNet network through CMC-DN01 and further used as the DeviceNet slave.

4.1 Data Exchange in DeviceNet



[Figure 4.1.1: Data exchange in DeviceNet]

(Figure 4.1.1) The DVP-SV series PLC and DVPDNET-SL are exchanging real-time data with each other. When the data in DVP-SV are sent to DVPDNET-SL, the data in DVPDNET-SL will be sent to DVP-SV, too. According to the I/O connections established (see 4.3 for details on I/O connections), DVPDNET-SL will send the data received from DVP-SV to the VFD-C2000 series AC motor drive, and VFD-C2000 will send its data back to DVPDNET-SL.

See 4.2 for how the data received from DVP-SV are sent to the parameters in the AC motor drive.

4.2 Mapping of CMC-DN01

The VFD-C2000 series AC motor drive is connected to the DeviceNet network through CMC-DN01. Once CMC-DN01 receives the I/O data outputted from the DeviceNet master, it will next send these data to parameters in the AC motor drive. The parameters in the AC motor drive to receive these data are determined by the mapping relation set in CMC-DN01, and the setup is done by using the DeviceNet Builder software.

Open the “Parameter Edit…” page in DeviceNet Builder (Figure 4.2.1) and you can see parameters “Length of input data”, “Length of output data”, “Data_in[1]” and “Data_out[1]”. See Table 4.2.1 for the definitions of these parameters. The “Parameter Edit…” page supports only decimal values; therefore, we have to convert the value into decimal form before filling it into the value column.

Parameter	Definition
Length of input data	Number of the AC motor drive parameters sent back to

Parameter	Definition
	DeviceNet master
Length of output data	Number of the AC motor drive parameters controlled by DeviceNet master
Data_in[1]	The 1st AC motor drive parameter sent back to DeviceNet master
Data_in[2]	The 2nd AC motor drive parameter sent back to DeviceNet master
Data_in[3]	The 3rd AC motor drive parameter sent back to DeviceNet master
…	…
Data_in[32]	The 32nd AC motor drive parameter sent back to DeviceNet master
Data_out[1]	The 1st AC motor drive parameter controlled by DeviceNet master
Data_out[2]	The 2nd AC motor drive parameter controlled by DeviceNet master
Data_out[3]	The 3rd AC motor drive parameter controlled by DeviceNet master
…	…
Data_out[32]	The 32nd AC motor drive parameter controlled by DeviceNet master

Table 4.2.1

Example: Suppose the AC motor drive parameters sent back to the DeviceNet master are H2101 and H2103, and the AC motor drive parameters controlled by the DeviceNet master are H2000 and H2001, we then set the “Length of input data” parameter to “2”, “Length of output data” to “2”, “Data_in[1]” to “8449” (converted from the hex 2101), “Data_in[2]” to “8451” (converted from the hex 2103), “Data_out[1]” to “8192” (converted from the hex 2000” and “Data_out[2]” to “8193” (converted from the hex 2001).

After the setup is completed, download the new mappings to CMC-DN01.

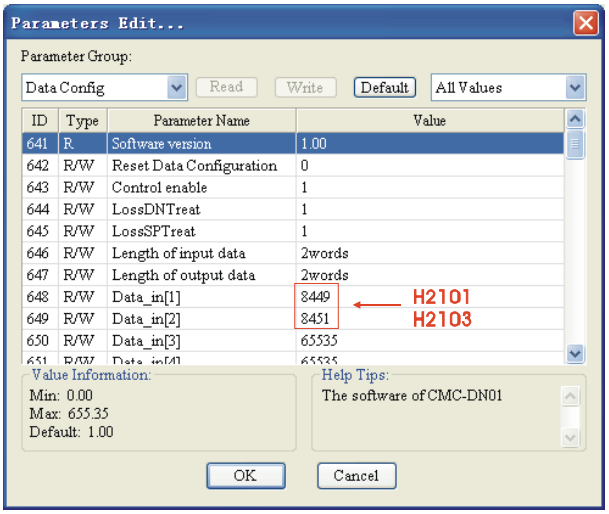


Figure 4.2.1: Input data mapping

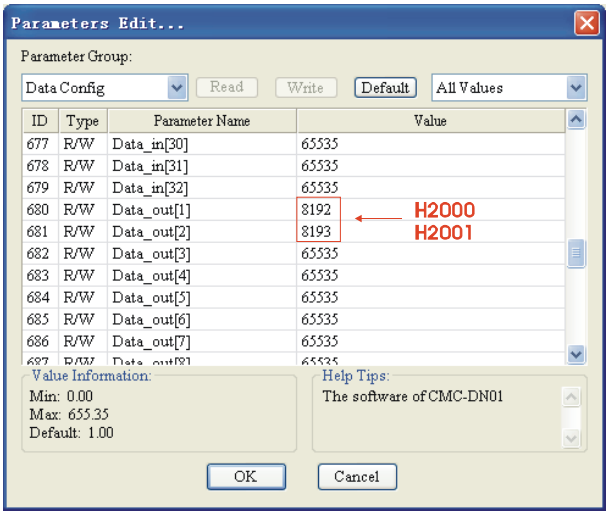


Figure 4.2.2: Output data mapping

4.3 Establishing I/O Connection

Open the “Scanner Module configuration…” page (Figure 4.3.1), and we can see that the AC motor

drive has already been configured in the DeviceNet master. The registers in the output table and input table are used for data exchange between the AC motor drive and the DeviceNet master. DVP-SV and DVPDNET-SL are exchanging data with each other. D6287, D6288, D6037 and D6039 are registers in DVP-SV. We can control and monitor parameters in the AC motor drive by controlling the registers in DVP-SV.

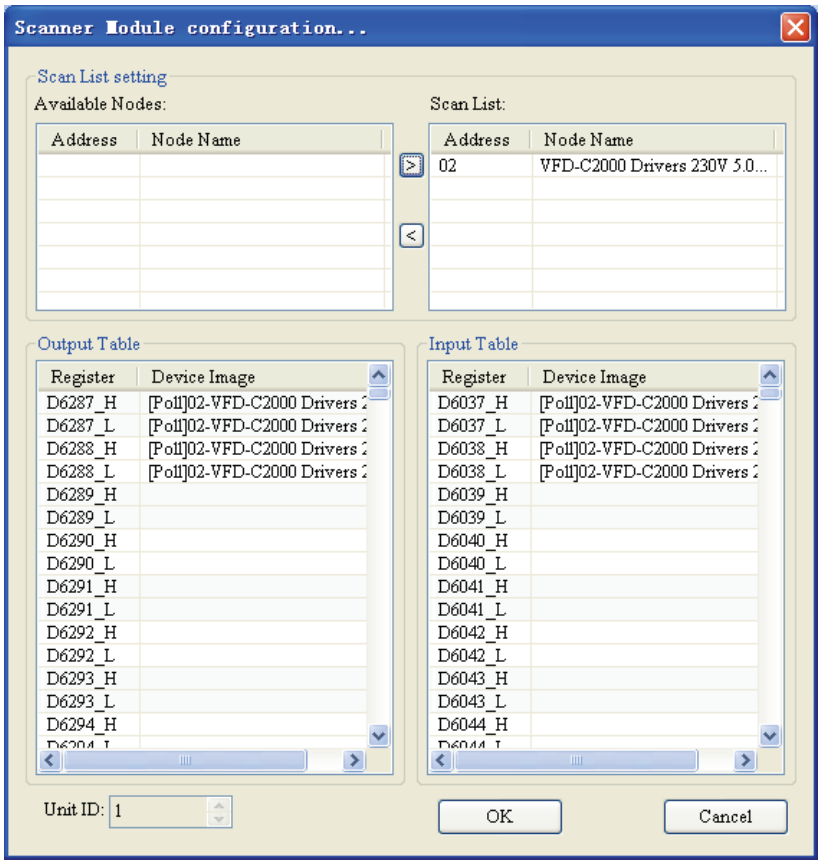


Figure 4.3.1

5 Constructing a DeviceNet Network

In this chapter, we will explain how to configure the VFD- C2000 series AC motor drive by an example.

5.1 Constructing a DeviceNet Network through CMC-DN01

1. The DeviceNet Network Structure

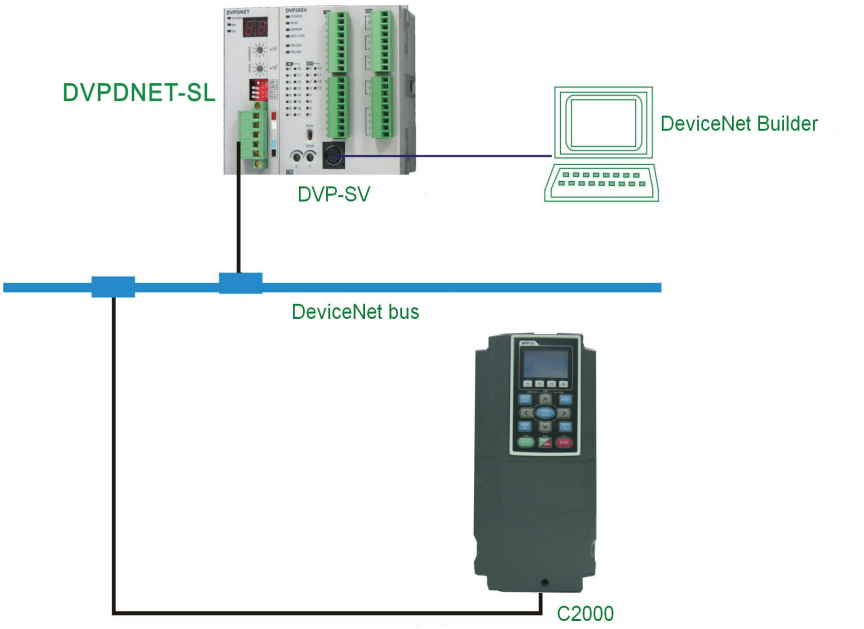


Figure 5.1.1: A constructed DeviceNet network

2. Setting DVPDNET-SL and VFD-C2000 series AC motor drive:

Module	Node address	Baud rate
DVPDNET-SL	1	500 kbps
VFD-C2000 series AC motor drive	2	500 kbps

Note: Setting the node address and baud rate of VFD-C2000 series AC motor drive in the DeviceNet network is shown below.

Parameter	Explanation	Setting range	
P00-20	The source of the frequency instruction	8	
P00-21	The source of the running instruction	5	
P09-30	Communication decoding method	0	
P9-70	The node address of the AC motor drive in the DeviceNet	DeviceNet: 0-63	
P9-71	The baud rate of the AC motor drive in the DeviceNet	Standard mode	Extended mode
		0: 125 Kbps 1: 250 Kbps 2: 500 Kbps	0: 10 Kbps 1: 20 Kbps 2: 50 Kbps 3: 125 Kbps 4: 250 Kbps 5: 500 Kbps 6: 800 Kbps 7: 1 Mbps

P9-72	The two modes of P9-71	When P9-72 is 0, P9-71 enters the standard mode; When P9-72 is 1, P9-71 enters the extended mode.

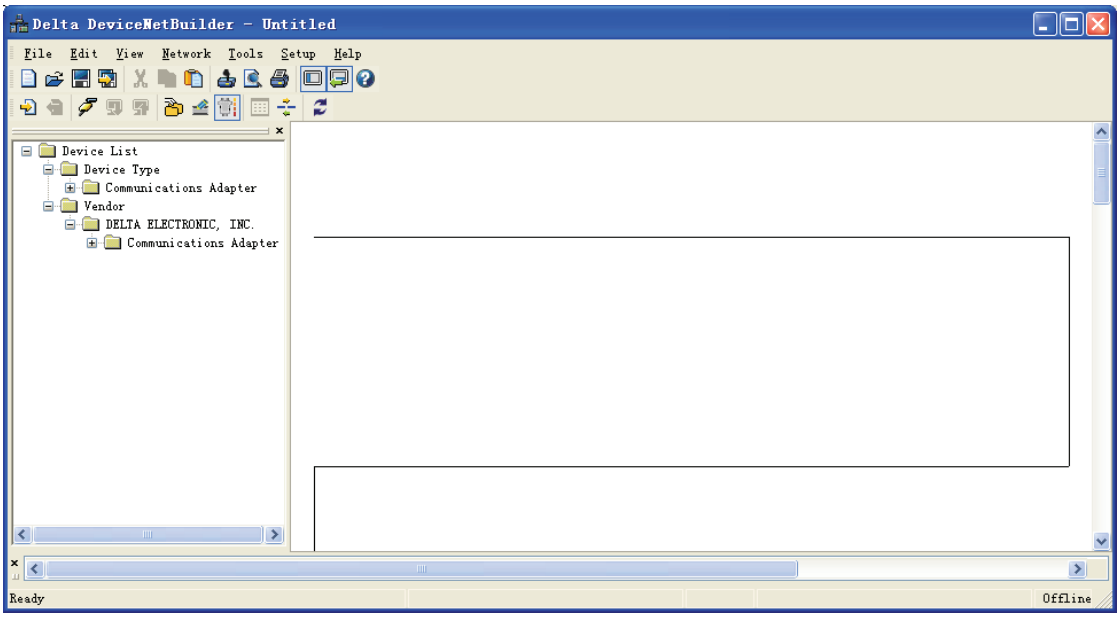
3. Please ensure that DVPDNET-SL and AC motor drive work normally; the wiring of the whole network is correct and the power supply of the DeviceNet network is normal. If online fails, please refer to the chapter of the LED indicators and trouble-shooting.

5.2 Configuring the Network by DeviceNet Builder

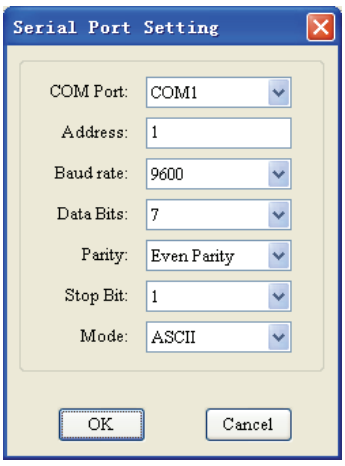
In this section, we will introduce how to configure the DeviceNet network by DeviceNet Builder.

- Configuring VFD-C2000 series AC motor drive

1. Open DeviceNet Builder.



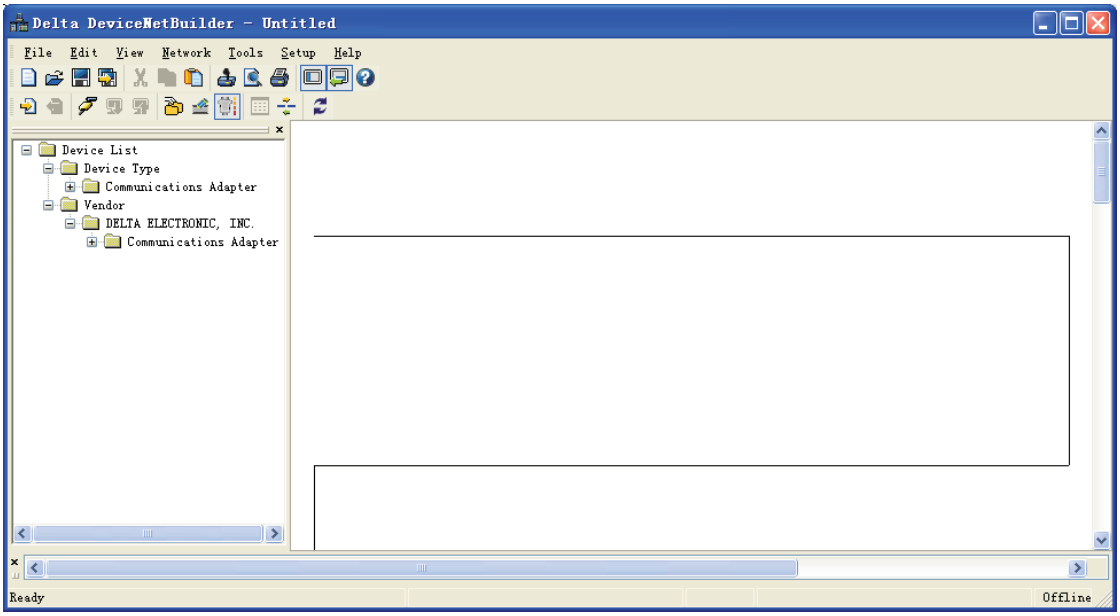
2. Select “Setup” => “Communication Setting” => “Serial Port Setting”.



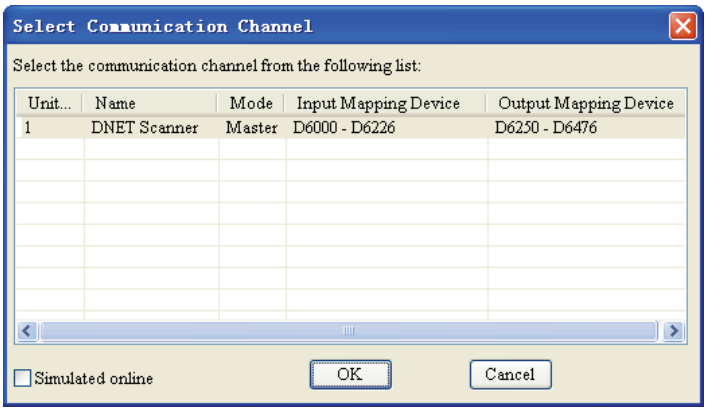
3. Set up the communication parameters.

Parameter	Definition	Default
COM port	Select the computer communication port which is to communicate with PLC	COM1
Address	Input the modbus node ID of PLC	01
Baud rate	Select the communication rate between computer and PLC	9,600 bps
Data bits	Select the communication protocol between computer and PLC	7
Parity		Even parity
Stop bit		1
Mode	Select the communication mode between computer and PLC	ASCII

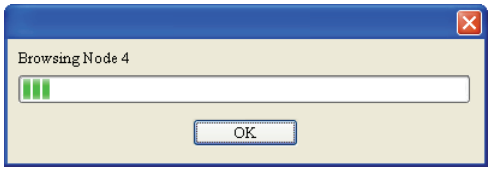
4. Click “OK” to return to the main page.



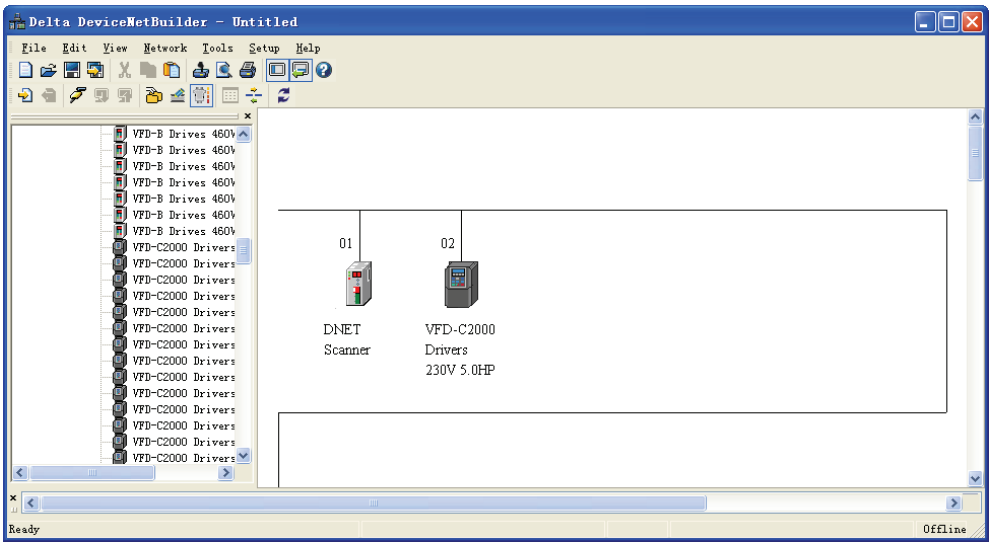
5. Select “Network” => “Online” to open the “Select Communication Channel” window.



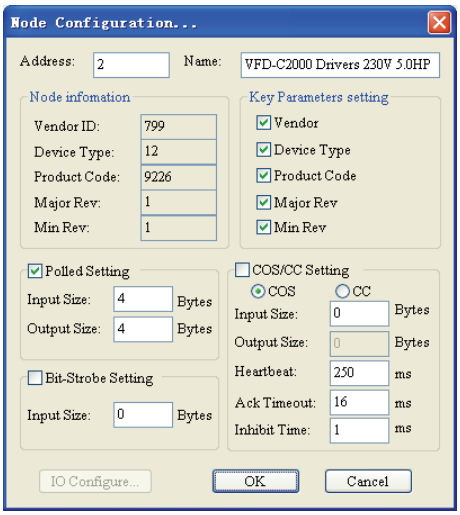
6. Click “OK” to start scanning the entire network.



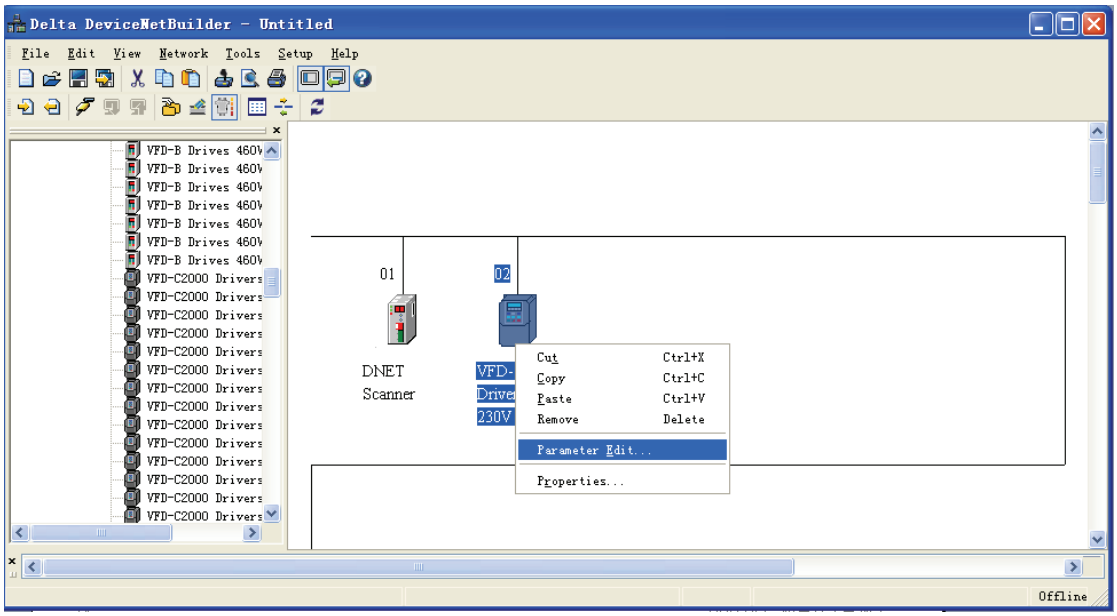
7. If the bar does not start to move forward, then it means the communication between the PC and PLC is abnormal or there are other programs also using the serial port. When the scanning is completed, a dialog box stating the scanning has been completed, and all the node icons and device names will be displayed in the software. In this example, the node address of DVPDNET-SL is “01”.



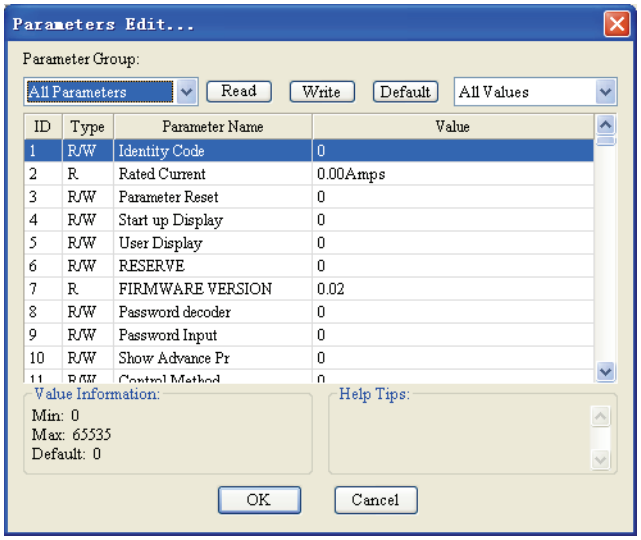
8. Double click VFD-C2000 Drivers (the AC motor drive connected to CMC-DN01) to open the “Node Configuration” dialog box. Set both the input size and output size to 4 bytes.



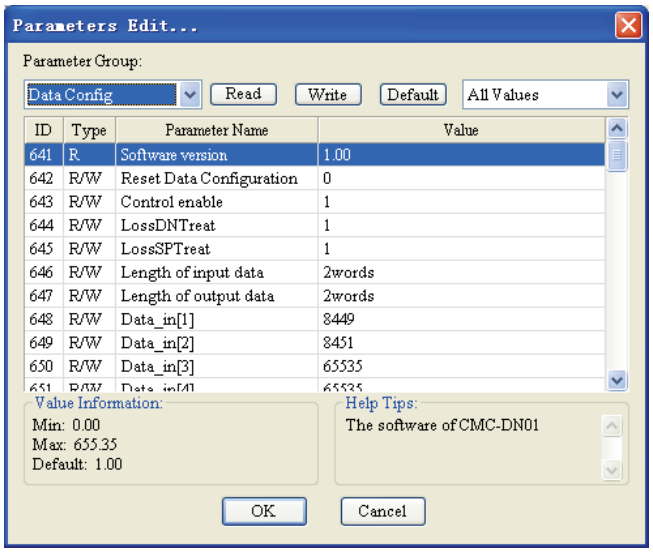
9. Right click VFD-C2000 Drivers and select “Parameter Edit…”.



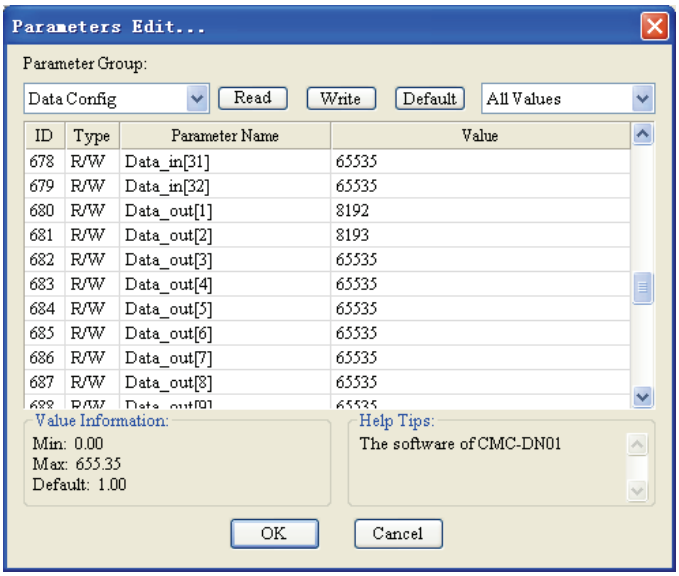
10. You will then see this dialog box.



11. Select “Data Config” in Parameter Group, and the data mapping will be displayed in the dialog box.



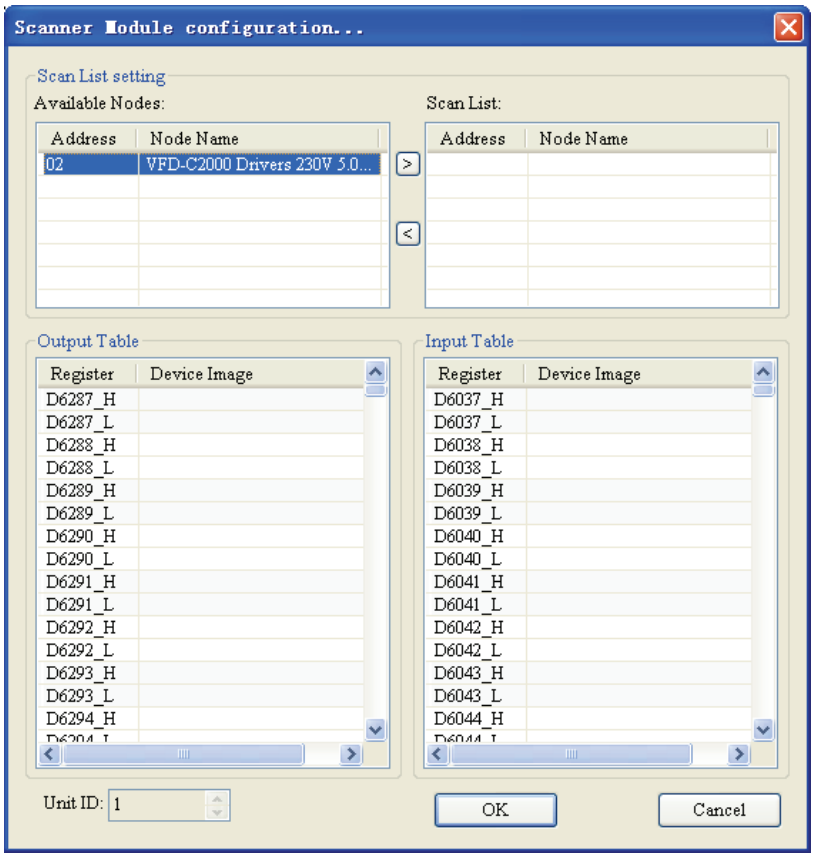
12. Set “Length of input data” parameter to “2 words”, “Length of output” to “2 words”, “Data_in[1]” to “8449” (converted from VFD-C2000 status word H2101), “Data_in[2]” to “8451” (converted from VFD-C2000 output frequency H2103), “Data_out[1]” to “8192” (converted from VFD-C2000 control word H2000” and “Data_out[2]” to “8193” (converted VFD-C2000 given frequency H2001). After all the settings are done, select “All Values” and click “Write”.



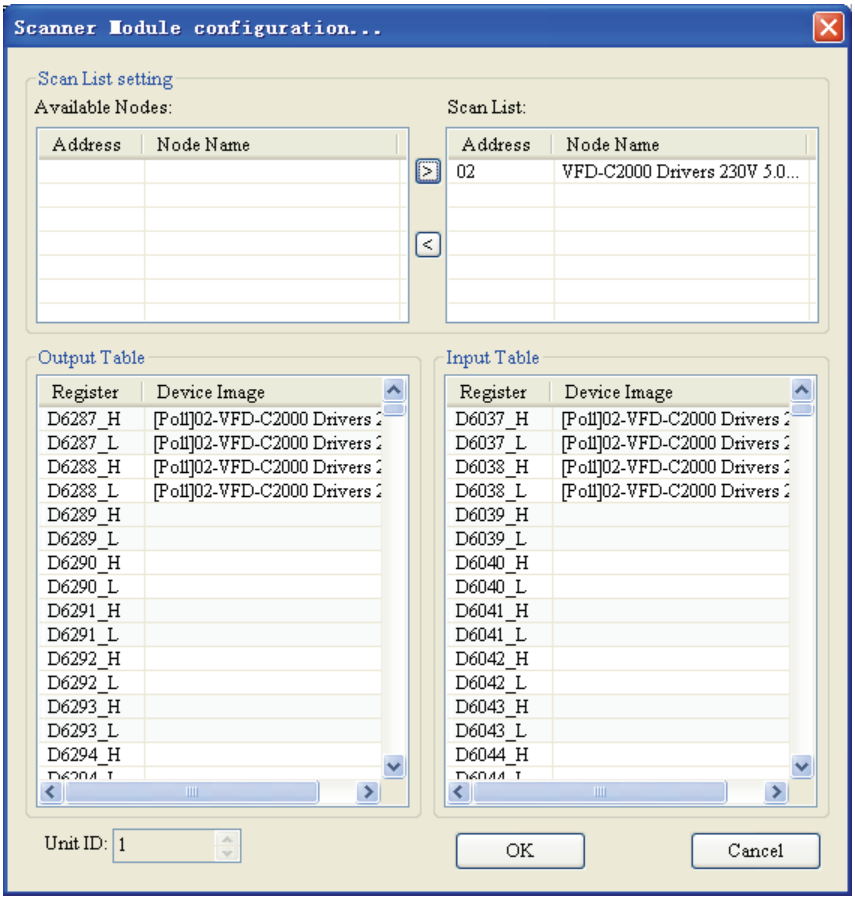
13. After the download is completed, re-power VFD-C2000.

- Configuring DVPDNET-SL scanner module

1. Double click DNET Scanner (node 01) to open the “Scanner Module configuration…” dialog box. We can now find the node VFD-C2000 Drives on the left-hand side table and an empty scan list on the right hand side.

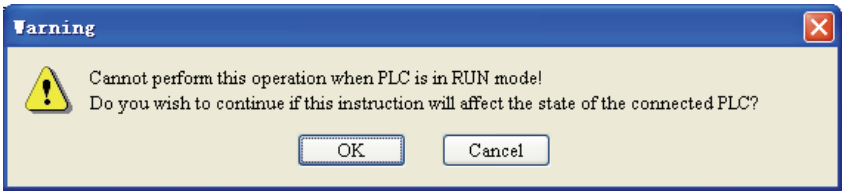


2. To add VFD-C2000 Drivers (DeviceNet slave) to the scan list, select the node and click >.



3. Make sure all the settings are correct and click “OK” to download the configurations to

DVPDNET-SL. If DVP-SV PLC is in RUN status during the download, a warning dialog box will appear.



4. Click “OK” to download the configuration to DVPDNET-SL and make sure the PLC is in RUN status. You will then see the MS LED and NS LED on CMC-DN01 are on in green color.

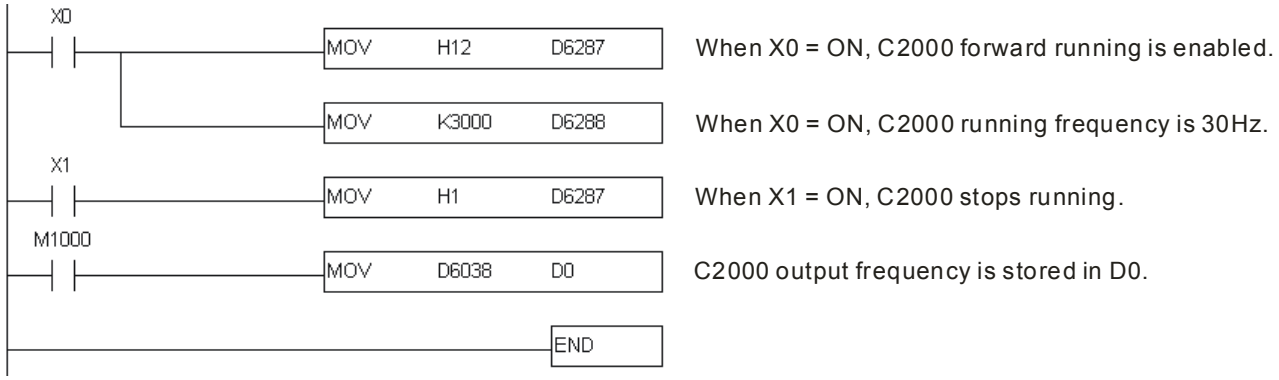
5.3 Data Mapping

After you configure the DeviceNet network following the steps above, you will then get the data mapping relations as below.

DVPDNET-SL		VFD-C2000 series AC motor drive
D6287	⇒	H2000
D6288		H2001
D6037	⇐	H2101
D6038		H2103

5.4 Editing Ladder Diagrams in PLC

I/O data contain the control word, status word, given frequency and output frequency of the AC motor drive, and therefore we manage to control the run/stop, forward running, reverse running and running speed of VFD-C2000 by a ladder diagram. See the example of a ladder diagram below.



6 Error Codes on Key Pad

When errors occur in the communication between CMC-DN01 and VFD-C2000 series AC motor drive, the error codes will be displayed on the digital key pad.

Error code	Indication	How to correct
ECid	CMC-DN01 duplicate MAC ID detection error; incorrect	Modify the value for parameter P09-70 in the AC motor drive and re-power it.

	node address for CMC-DN01	
ECLv	The 5V power supplied by the AC motor drive to CMC-DN01 is too low.	Check the power supply of the PLC.
ECtt	CMC-DN01 enters test mode.	Re-power the AC motor drive.
ECbF	DeviceNet bus-off	Re-power the AC motor drive.
ECnP	No power supply on the DeviceNet network	1) Check the wiring of CMC-DN01 and power supply in DeviceNet. 2) Reset CMC-DN01 to default settings.

7 LED Indicators and Trouble-shooting

There are 3 LED indicators on CMC-DN01. POWER LED displays the status of the power supply. MS LED and NS LED are dual-color LEDs, displaying the connection status of the communication and errors.

7.1 POWER LED

LED status	Indication	How to correct
Off	Power supply in abnormal status	Check the power supply of CMC-DN01.
On	Power supply in normal status	--

7.2 NS LED

LED status	Indication	How to correct
Off	No power supply or CMC-DN01 has not completed the MAC ID test yet.	1. Check the power of CMC-DN01 and see if the connection is normal. 2. Make sure at least one or more nodes are on the bus. ◦ 3. Check if the serial baudrate of CMC-DN01 is the same as that of other nodes.
Green light flashes	CMC-DN01 is online but has not established connection to the master.	1. Configure CMC-DN01 to the scan list of the master. 2. Re-download the configured data to the master.
Green light on	CMC-DN01 is online and is normally connected to the master.	--
Red light flashes	CMC-DN01 is online, but I/O connection is timed-out.	1. Check if the network connection is normal. 2. Check if the master operates normally.
Red light on	1. The communication is down. 2. MAC ID test failure 3. No power supply for network 4. CMC-DN01 is offline.	1. Make sure all the MAC IDs on the network are not repeated. 2. Check if the network installation is normal. 3. Check if the serial baudrate of CMC-DN01 is consistent with that of other nodes. 4. Check if the node address of CMC-DN01 is illegal.

LED status	Indication	How to correct
		5. Check if the power supply for the network is normal.

7.3 MS LED

LED status	Indication	How to correct
Off	No power supply or being offline	Check the power supply of CMC-DN01 and see if the connection is normal.
Green light flashes	Waiting for I/O data	Switch the master PLC to RUN status.
Green light on	I/O data are normal.	--
Red light flashes	Mapping error	1. Re-configure CMC-DN01. 2. Re-power the AC motor drive.
Red light on	Hardware error	1. See the error code displayed on the AC motor drive. 2. Send back to factory for repair if necessary.
Orange light flashes	CMC-DN01 is establishing connection with the AC motor drive.	If the flashing lasts for a long time, check if CMC-DN01 and the AC motor drive are correctly installed and normally connected to each other.

Appendix: DeviceNet Objects Supported

DeviceNet Objects List

Class	Object
0x01	Identity object
0x02	Message router object
0x03	DeviceNet Object
0x05	Connection object
0x0F	Parameter Object
0x95	DataConf object

Class 0x01 – Identity objects

Class attribute

Attribute ID	Access rule	Name	Data type
1	Get	Revision	UINT
2	Get	MaxInstance	UINT
3	Get	NumberofInstances	UINT
6	Get	MaxIdClass	UINT
7	Get	MaxIdInstance	UINT

Instance

Attribute ID	Access rule	Name	Data type
1	Get	VendorId	UINT
2	Get	DeviceType	UINT
3	Get	ProductCode	UINT
4	Get	Revision MaxRev MinRev	 USINT USINT
5	Get	Status	WORD
6	Get	Sn	UDINT
7	Get	ProdName StrLen ASCIIStr	 USINT STRING

Common services

Service code	Implemented for		Service name
	Class	Instance	
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attribute_Single

Class 0x02 – Message router objects

Class attribute

Attribute ID	Access rule	Name	Data type
1	Get	Revision	UINT
6	Get	MaxIdClass	UINT
7	Get	MaxIdInstance	UINT

Instance

Attribute ID	Access rule	Name	Data type
2	Get	NumAvailable	UINT
3	Get	NumActive	UINT

Common services

Service code	Implemented for		Service name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single

Class 0x03 – DeviceNet objects

Class attribute

Attribute ID	Access rule	Name	Data type
1	Get	Revision	UINT

Instance

Attribute ID	Access rule	Name	Data type
1	Get	MACID	USINT
2	Get	BaudRate	USINT
3	Get/Set	BusofInterrupt	BOOL
4	Get/Set	BusofCounter	USINT
5	Get	AllocationInfo AllocationChoice MasterNodeAddress	BYTE USINT
6	Get	MACIDSwitchChanged	BOOL
7	Get	BaudRateSwitchChanged	BOOL
8	Get	MACIDSwitchValue	USINT
9	Get	BaudRateSwitchValue	USINT

Common services

Service code	Implemented for		Service name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single
0x4B	No	Yes	Allocate_Master/Slave_Connection_Set
0x4C	No	Yes	Release_Master/Slave_Connection_Set

Class 0x05 – Connection objects

Class attribute

Attribute ID	Access rule	Name	Data type
1	Get	Revision	UINT

Instance 1: Explicit message connection

Attribute ID	Access rule	Name	Data type
1	Get	State	USINT
2	Get	InstanceType	USINT
3	Get	TransportClassTrigger	USINT
4	Get	ProducedConnectionId	UINT
5	Get	ConsumedConnectionId	UINT

Attribute ID	Access rule	Name	Data type
6	Get	InitialCommCharacterisitcs	BYTE
7	Get	ProducedConnectionSize	UINT
8	Get	ConsumedConnectionSize	UINT
9	Get/Set	ExpectedPackedRate	UINT
12	Get/Set	WatchdogTimeoutAction	USINT
13	Get	Produced Connection Path Length	USINT
14	Get	Produced Connection Path	EPATH
15	Get	Consumed Connection Patch Length	USINT
16	Get	Consumed Connection Path	EPATH

Instance 2: Polled I/O connection

Attribute ID	Access rule	Name	Data type
1	Get	State	USINT
2	Get	InstanceType	USINT
3	Get	TransportClassTrigger	USINT
4	Get	ProducedConnectionId	UINT
5	Get	ConsumedConnectionId	UINT
6	Get	InitialCommCharacteristics	BYTE
7	Get	ProducedConnectionSize	UINT
8	Get	ConsumedConnectionSize	UINT
9	Get/Set	ExpectedPackedRate	UINT
12	Get/Set	WatchdogTimeoutAction	USINT
13	Get	Produced Connection Path Length	USINT
14	Get	Produced Connection Path	EPATH
15	Get	Consumed Connection Path Length	USINT
16	Get	Consumed Connection Path	EPATH

Common services

Service code	Implemented for		Service name
	Class	Instance	
0x05	No	Yes	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Class 0x96 – Parameter objects

Class attribute

Attribute ID	Access rule	Name	Data type
1	Get	Revision	UINT

Instance 1: Parameter Instance 1 through N

Attribute ID	Access rule	Name	Data type
1	Get/Set	Parameter Value	--

Attribute ID	Access rule	Name	Data type
2	Get	Link Path Size	USINT
3	Get	Link Path	--
4	Get	Descriptor	WORD
5	Get	Data Type	USINT
6	Get	Data Size	USINT

Common services

Service Code	Implemented for		Service Name
	Class	Instance	
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Class 0x95 – DataConf

Class attribute

Attribute ID	Access rule	Name	Data type
1	Get	Revision	UINT

Instance 1 ~ N

Attribute ID	Access rule	Name	Data type
1	Get/Set	Parameter Value	--
2	Get	Link Path Size	USINT
3	Get	Link Path	--
4	Get	Descriptor	WORD
5	Get	Data Type	USINT
6	Get	Data Size	USINT

Common services

Service Code	Implemented for		Service Name
	Class	Instance	
0X05	Yes	No	Reset
0x0E	Yes	Yes	Get_Attribute_Single
0x10	No	Yes	Set_Attribute_Single

Instance list

Instance ID	Access rule	Name	Data type	Default
1	Get	Software version	UINT	####
2	Get/Set	Reset Configuration	UINT	0
3	Get/Set	Control enable	UINT	1
4	Get/Set	LossDNTreat	UINT	1
5	Get/Set	LossSPTreat	UINT	1
6	Get/Set	Output Length (master->card)	UINT	2 words
7	Get/Set	Input Length (card -> master)	UINT	2 words
10	Get/Set	Output[0] master->card	UINT	2000H
11	Get/Set	Output[1]	UINT	2001H

Instance ID	Access rule	Name	Data type	Default
12	Get/Set	Output[2]	UINT	FFFFH
13	Get/Set	Output[3]	UINT	FFFFH
14	Get/Set	Output [4]	UINT	FFFFH
15	Get/Set	Output [5]	UINT	FFFFH
16	Get/Set	Output [6]	UINT	FFFFH
17	Get/Set	Output[7]	UINT	FFFFH
18	Get/Set	Output[8]	UINT	FFFFH
19	Get/Set	Output[9]	UINT	FFFFH
20	Get/Set	Output[10]	UINT	FFFFH
21	Get/Set	Output[11]	UINT	FFFFH
22	Get/Set	Output[12]	UINT	FFFFH
23	Get/Set	Output[13]	UINT	FFFFH
24	Get/Set	Output[14]	UINT	FFFFH
25	Get/Set	Output[15]	UINT	FFFFH
26	Get/Set	Output[16]	UINT	FFFFH
27	Get/Set	Output[17]	UINT	FFFFH
28	Get/Set	Output[18]	UINT	FFFFH
29	Get/Set	Output[19]	UINT	FFFFH
30	Get/Set	Output[20]	UINT	FFFFH
31	Get/Set	Output[21]	UINT	FFFFH
32	Get/Set	Output[22]	UINT	FFFFH
33	Get/Set	Output[23]	UINT	FFFFH
34	Get/Set	Output[24]	UINT	FFFFH
35	Get/Set	Output[25]	UINT	FFFFH
36	Get/Set	Output[26]	UINT	FFFFH
37	Get/Set	Output[27]	UINT	FFFFH
38	Get/Set	Output[28]	UINT	FFFFH
39	Get/Set	Output[29]	UINT	FFFFH
40	Get/Set	Output[30]	UINT	FFFFH
41	Get/Set	Output[31]	UINT	FFFFH
42	Get/Set	Input[0] card->master	UINT	2101H
43	Get/Set	Input[1]	UINT	2103H
44	Get/Set	Input[2]	UINT	FFFFH
45	Get/Set	Input[3]	UINT	FFFFH
46	Get/Set	Input[4]	UINT	FFFFH
47	Get/Set	Input[5]	UINT	FFFFH
48	Get/Set	Input[6]	UINT	FFFFH
49	Get/Set	Input[7]	UINT	FFFFH
50	Get/Set	Input[8]	UINT	FFFFH
51	Get/Set	Input[9]	UINT	FFFFH
52	Get/Set	Input[10]	UINT	FFFFH
53	Get/Set	Input[11]	UINT	FFFFH
54	Get/Set	Input[12]	UINT	FFFFH
55	Get/Set	Input[13]	UINT	FFFFH
56	Get/Set	Input[14]	UINT	FFFFH

Instance ID	Access rule	Name	Data type	Default
57	Get/Set	Input[15]	UINT	FFFFH
58	Get/Set	Input[16]	UINT	FFFFH
59	Get/Set	Input[17]	UINT	FFFFH
60	Get/Set	Input[18]	UINT	FFFFH
61	Get/Set	Input[19]	UINT	FFFFH
62	Get/Set	Input[20]	UINT	FFFFH
63	Get/Set	Input[21]	UINT	FFFFH
64	Get/Set	Input[22]	UINT	FFFFH
65	Get/Set	Input[23]	UINT	FFFFH
66	Get/Set	Input[24]	UINT	FFFFH
67	Get/Set	Input[25]	UINT	FFFFH
68	Get/Set	Input[26]	UINT	FFFFH
69	Get/Set	Input[27]	UINT	FFFFH
70	Get/Set	Input[28]	UINT	FFFFH
71	Get/Set	Input[29]	UINT	FFFFH
72	Get/Set	Input[30]	UINT	FFFFH
73	Get/Set	Input[31]	UINT	FFFFH