

Modbus/DeviceNet Gateway

User Manual

Oct, 2007

Catalog

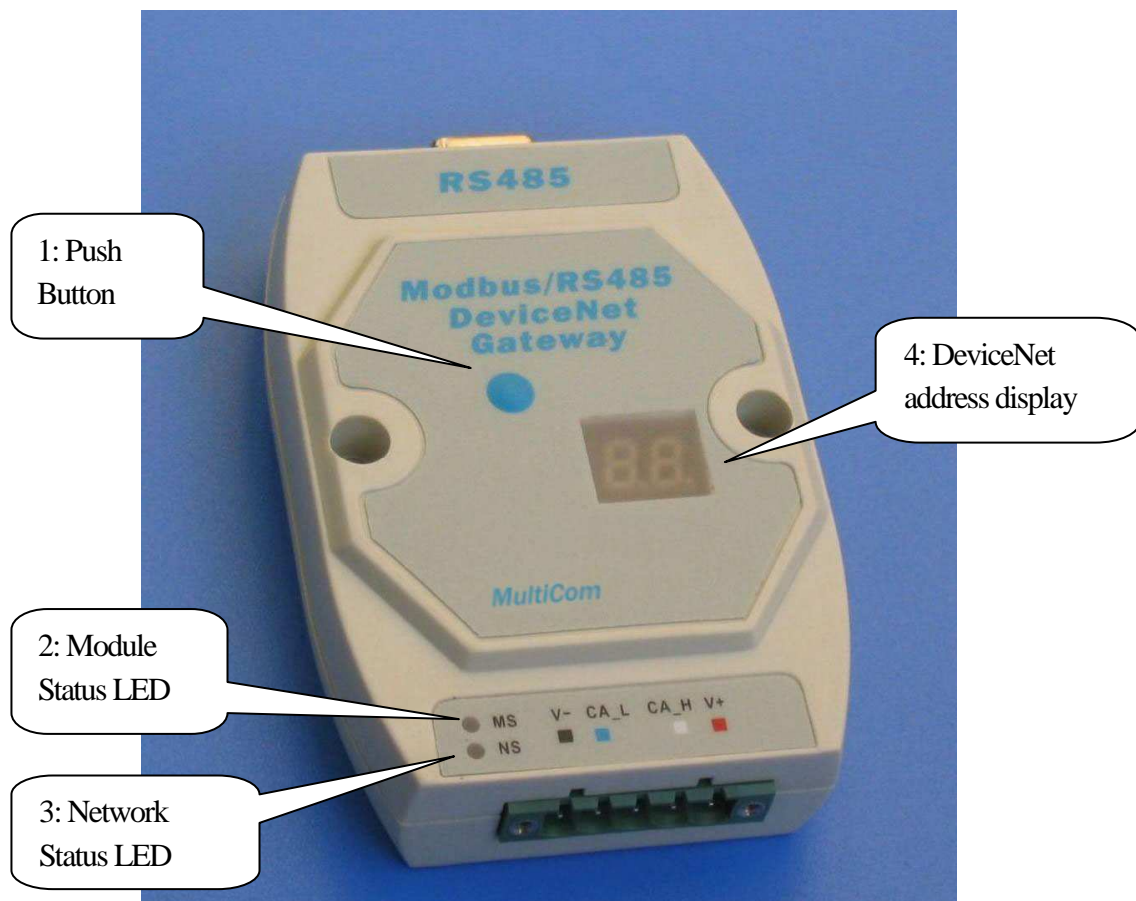
1. Overview	2
Introduction.....	2
General Features.....	2
Sub Network.....	3
2. Installation	3
2.1. DeviceNet Connector	3
2.2. Serial Communication Connector	4
2.3. DIN Rail Connection.....	5
2.4. Terminal Resistor.....	5
2.5. Setting of DeviceNet Address	5
3. Setting the Modbus Serial Port.....	5
3.1. Modbus Configuration	6
3.2. Configure the Modbus Scan List	7
3.3. DeviceNet Configuration.....	11
3.4. Exit the Program.....	12
4. Status Indicators.....	12
5. DeviceNet EDS information	13
6. Technical Specification.....	13
Mechanical.....	13
Environmental	14
EMC	14
Power Supply	14
Power Consumption.....	14
Appendix A: Modbus Communication	15
Introduction.....	15
Modbus RTU	15
Modbus commands.....	15
Appendix B: DeviceNet Profile.....	17
Appendix C: EDS file	22

1. Overview

Introduction

The Modbus/DeviceNet gateway provides a flexible connection between your Modbus enable device and DeviceNet network. Actually, the MultiCom Modbus/DeviceNet Communicator for DeviceNet acts as a gateway between almost any serial application protocol and DeviceNet network. If your device has a serial communication port, not only the simple input/output information can be transferred, but also the configuration parameter, diagnostic information built in your smart device can be read and written remotely via DeviceNet.

Following is the outlook of MD21:



General Features

- DeviceNet Group 2 Only Server communication capability (DeviceNet Slave)
- Modbus RTU mode Master
- Save/load configuration in EEPROM
- Can be customized to user's protocol
- DIN-rail mountable

Sub Network

RS485/Modbus

- Modbus RTU Master mode
- Parameters and I/O data can be network accessible

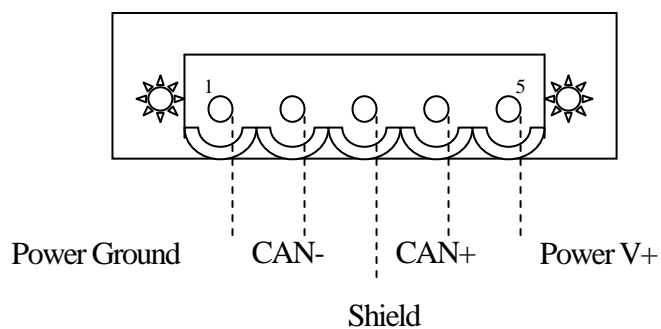
DeviceNet Interface

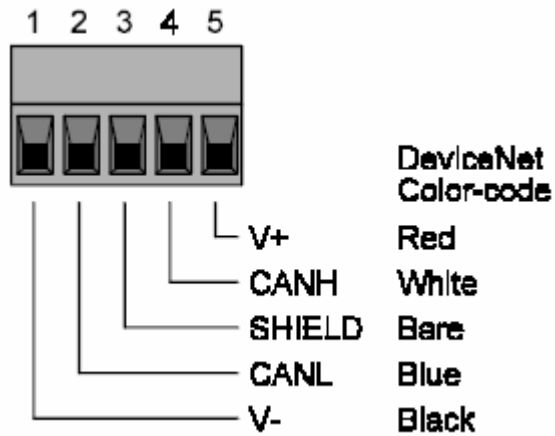
- Complete DeviceNet adapter functionality (profile no.12, group 2 only server)
- Mac ID and baudrate configuration via Network
- Polled, Change-of-state and Bit Strobe I/O

2. Installation

2.1. DeviceNet Connector

Wiring of DeviceNet

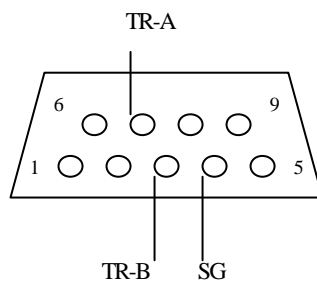




2.2. Serial Communication Connector

Modbus connection use Female DB9 connection. The Pins assignments are as follows:

MD21 utilize the DB9 female connector. Wiring of Modbus Connector:



- ✧ Pin3: TR-B
- ✧ Pin 4: Singal Groud
- ✧ Pin 7: TR-A
- ✧ Pin 8: 5V, For factory test, Don't use it!

Pin 3	RS485-A, Data+
Pin 8	RS485-B, Data -
Pin 7	VCC
Pin 2	GND

2.3. DIN Rail Connection

35mm DIN-Rail.

2.4. Terminal Resistor

You need to add two terminal resistor to DeviceNet network at the end point of the link. The type of resistor is 120ohm(metal film). Terminal resistors are also needed in Modbus. The Modbus terminal resistor is 150 ohm. There is a 150 ohm terminal resistor in MD21. You need to add another one.

2.5. Setting of DeviceNet Address

Keep pressing the blue push button longer than 2.5s, the adaptor will transfer to DeviceNet address setting mode. Then you can click the push button to add “1” to the DeviceNet address. If set, keep pressing PB longer than 2.5s again, the device will change to new address and restart with new address. The address is stored in Non volatile memory.

3. Setting the Modbus Serial Port

The Modbus-DeviceNet gateway MD21 of MultiCom can be configured by Windows Hyper-Terminal.

You can find the Hyper-Terminal in “Windows—Start—All programs—Accessories— Communications —Hyper-Terminal.”

Double Click the Push Button 1 to let the gateway go into the configuration mode.

In the following chapter, how to configure Modbus communication and some DeviceNet communication is introduced. And How to configure the Modbus Scan List and Modbus-DeviceNet map is the key point.

Notes: If the gateway has already been in normal DeviceNet communication, you can not let it go into configuration mode directly. Double click the PB quickly, then you will see the MS (Module Status) LED and NS (Network Status) LED blinking red and green alternatively. Double click the PB again in time when the led is blinking or DeviceNet Address LED is blinking “BR”.

The configuration tool uses **19200-8-N-1** Mode.

When the module has successfully gone into configuration mode, you will see this Start Menu.

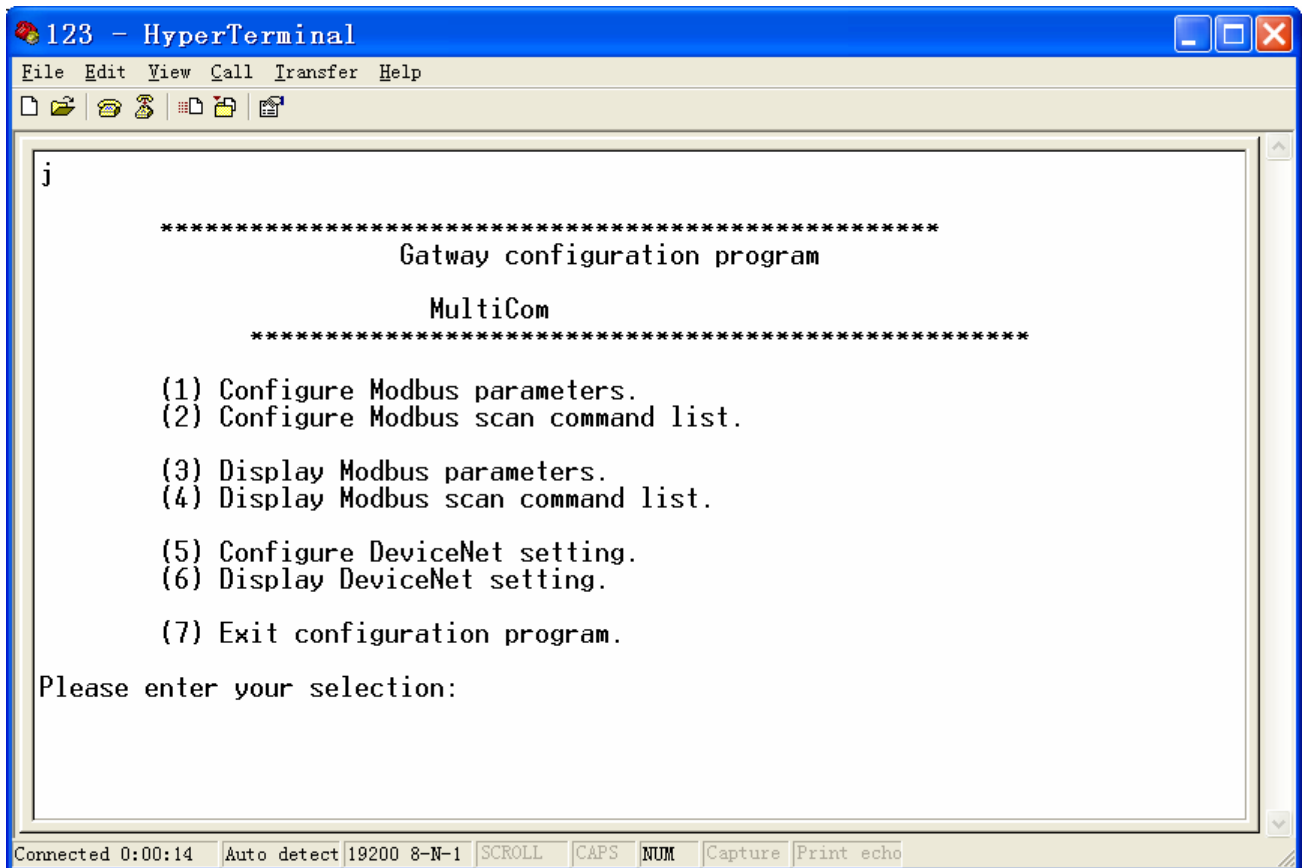


Figure 1: The Main Menu

[In following section, italic and bold words mean what you see in the Hyper-Terminal tool.]

3.1. Modbus Configuration

The first two items are used to configure Modbus.

- (1) Configure Modbus parameters.*
- (2) Configure Modbus scan command list.*

The 3rd and 4th Items are used to display your Modbus communication setting.
Enter "1", you will see:

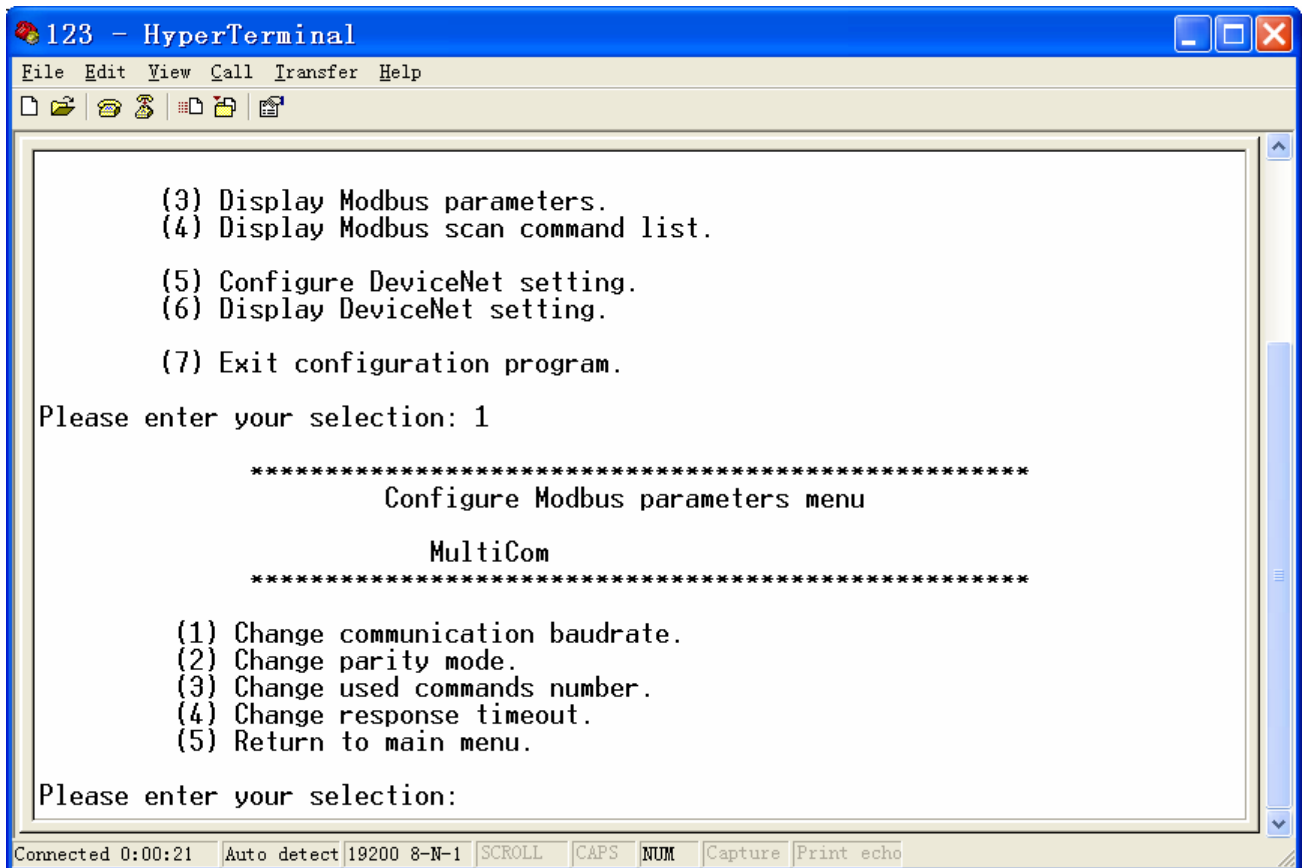


Figure 2: Modbus Communication General Configuration

The Modbus port of MD21 acts as a Modbus master.

The item “(1). *Change communication baudrate.*” is to modify the baudrate of Modbus communication. You can select it from a list.

(2) *Change parity mode.* In this item you can change your parity verification mode of Modbus communication.

(3) *Change used commands number.* You can change the number of how many commands that you have already configured to use. For instance, if you configured 30 Modbus read/write commands but you only want use the first 20 commands, set the commands number to 20 here.

(4) *Change response timeout.* The Modbus response timeout setting is input here.

(5) *Return to main menu.* Go to the main menu.

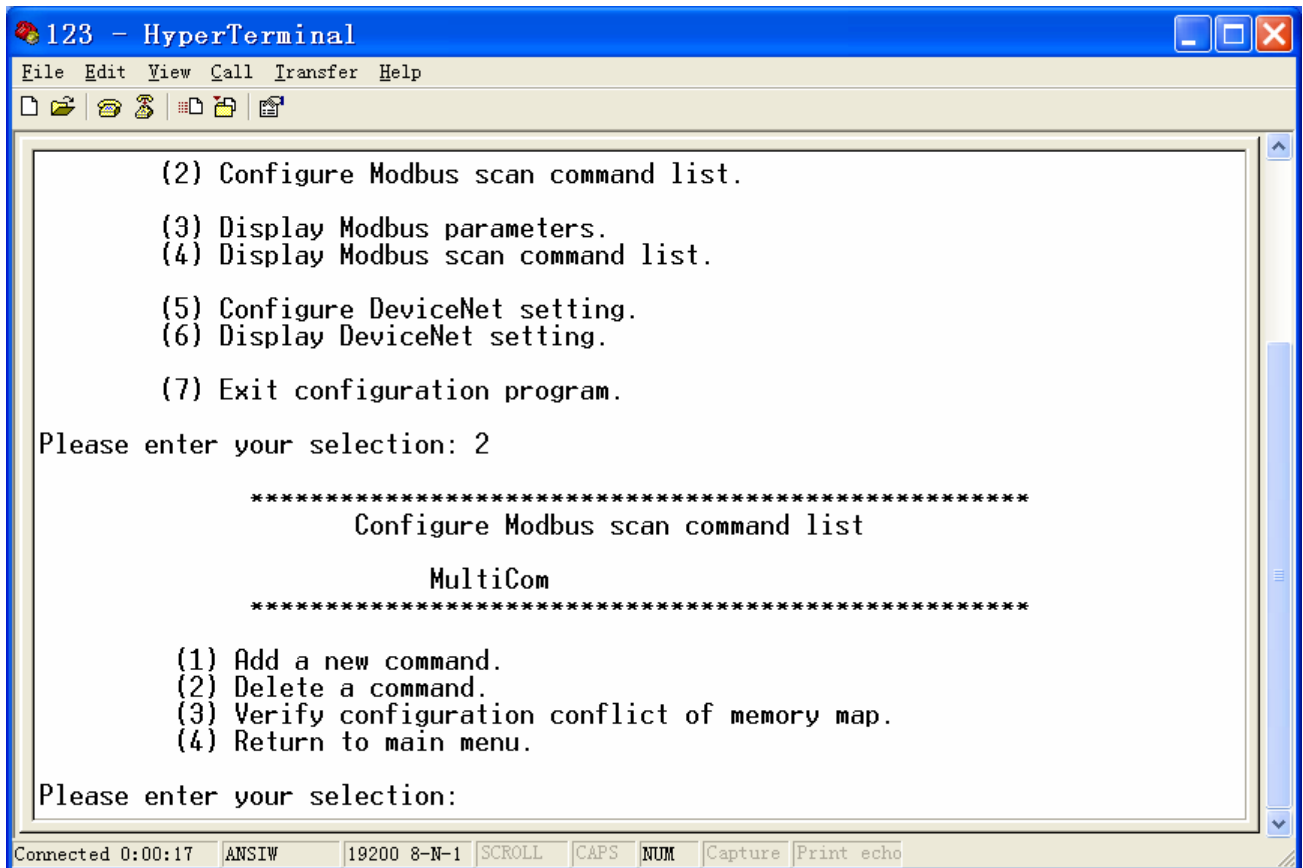
3.2. Configure the Modbus Scan List

Talking with which Modbus device, using which command, reading/writing which registers are defined here. How Modbus I/O map to DeviceNet I/O is defined in each commands. If you are not familiar to Modbus, please refer to MODBUS specification.

Configure the Modbus scan list and list map are the core configuration of Gateway.

The Modbus scan list means the gateway Modbus master **scan which devices, use which command and read/write which registers on the Modbus link.** **The list MAP** means the **data mapping** definition of how Modbus input/output map to DeviceNet I/O.

In main menu, select “(2). Configure Modbus scan command list.” You will see:



Configuration Modbus scan command list

MultiCom

- (1) Add a new command.*
- (2) Delete a command.*
- (3) Verify configuration conflict of memory map.*
- (4) Return to main menu.*

Please input your selection:

Select "1" to add a new command:

You will see:

Maxim configurable command number is 48.

New command No. is: 1

There are at most **48** commands that you can use. Each command can be defined to use one Modbus command to read/write a couple of registers of one device.

For example, you can define a command to read 10 register from 6 to 15 of Modbus node 5 as this way:

Please enter slave address (0-247): 5

Your enter is: 5

Press “3” then “Enter” mean read Modbus node address 3, you will see:

Please enter Function code (support FC is: 01;02;03;04;05;06;15;16): 3

Your enter is: 3

Select “3” means Modbus command with functional code 3, then you need to input start address of the registers:

Please enter Starting Address: 6

Your enter is: 6

Please enter Number of Regs (1-32): 10

Your enter is: 10

After that, you set the starting address of 6. And you will be asked to input number of register to read in this command immediately.

Select the number, here you input **10**,

Please enter data mapping address(1-159):

Notes: Don't Map byte 0. It's for Modbus Status (1-159)

Suggest to Use[1] 2

Then is the data mapping to DeviceNet input. Modbus command 03 is an input command, so the data will be mapped to DeviceNet input. The maxim DeviceNet input bytes number is 128. You can choose the location 1-127.

Please do NOT use the byte order zero. It is used to express the Modbus status, if 0, The Modbus is Ok, and otherwise, the Modbus data is meaningless. There is communication error occurred or even no Modbus device ongoing there. Usually your control system needs to know that.

Now is the last parameter to set.

Please select scan cycle:

(0) 50ms

(1) 200ms

(2) 1000ms

Please enter your selection:0

That means how much millisecond the gateway will send this read command once. Choose it in the range as indicated such as 50 (milliseconds).

You will see:

Create success!

Press any key to continue...

Example 2:

In the following we will give another example. In this example we will set a Modbus “BIT” operation command **15-Write Multiple Coils**

Select:

(1) Add a new command.

Then follow the wizard:

(support FC is: 01;02;03;04;05;06;15;16): 15

Your enter is: 15

Please enter Coil Address: 25

Your enter is: 25

Please enter Quantity of Coils: 10

Your enter is: 10

Please enter source address:(0-111) 10

Your enter is: 10

Here you will see the difference with the register based Modbus function code:

You gateway support **bit mapping**. You can only map the input or output coils to bits of the memory bytes.

Please enter bit offset (0-7): 2

Your enter is: 2

Please select scan cycle:

(0) 50ms

(1) 200ms

(2) 1000ms

Please enter your selection:

Now, if you want to know whether or not the right parameters are set, you can select

(4). Display Modbus scan command list.

In main menu, select 4,

Then you will see:

Display Modbus scan command list

MultiCom

(1) Display command list.

(2) Return main menu.

Please input your selection

Then select 1:

Please input your selection: 1

The following information will be shown:

<i>No.</i>	<i>SlaA</i>	<i>FC</i>	<i>StarA</i>	<i>No.P</i>	<i>Bytes</i>	<i>DA/SA</i>	<i>Bits</i>	<i>SC(ms)</i>
<i>1</i>	<i>5</i>	<i>3</i>	<i>6</i>	<i>10</i>		<i>2</i>		<i>200</i>
<i>2</i>	<i>50</i>	<i>15</i>	<i>25</i>	<i>10</i>	<i>2</i>	<i>10</i>	<i>2</i>	<i>1000</i>

Display end!

Press any key to continue...

3.3. DeviceNet Configuration

Then the next two are used to configure some DeviceNet configurations.

(5). Configure DeviceNet parameters.

(6). Display DeviceNet parameters.

Enter 5, you will see

***** Configure DeviceNet Menu *****

(1) Change INPUT bytes number of DeviceNet I/O connection.

(2) Change OUTPUT bytes number of DeviceNet I/O connection.

(3) Return main menu.

Please enter your selection:

You can change DeviceNet I/O connection input/output size of our product. As you know, if you just output 4 bytes, the gateway padded 60 bytes to 64bytes. It's not efficient for DeviceNet network capacity and DeviceNet Master/Scanner. You can reconfigure it. But please pay attention to cooperate with Modbus-DeviceNet data mapping. **Your DeviceNet input/output size must be same as the byte size configured in the DeviceNet configuration software.**

For example,

Please enter your selection: 2

Current output Connection size is 112, Please select

- (0) 8 bytes
- (1) 16 bytes
- (2) 32 bytes
- (3) 64 bytes
- (4) 96 bytes
- (5) 112bytes

As shown in next figure:

You can check the DeviceNet setting in item 6 of Main Menu:

(5) Configure DeviceNet setting.

(6) Display DeviceNet setting.

(7) Exit configuration program.

Please enter your selection: 6

Display DeviceNet setting

DeviceNet input size is: 160

DeviceNet output size is: 112

3.4. Exit the Program

In Main Menu, enter “7” to exit program. You need to restart the gateway to make all modification to take effects.

(7). Exit configuration program.

4. Status Indicators

Module status indicator (MS):

LED State	Status
Off	No power
Solid Green	Device Operational
Flashing Green	Fault, Incomplete or incorrect configuration
Flashing Red	Minor Fault
Solid Red	Unrecoverable fault

Flashing Red and Green	Self testing
------------------------	--------------

Network status indicator (NS):

LED state	Status
Off	Not powered / Not online
Flashing Green	On line, Not connected
Solid Green	Link OK, On line, Connected
Flashing Red	One or more connection timeout
Solid Red	Critical Link Failure

5. DeviceNet EDS information

DeviceNet Configuration Tool (.EDS - file)

Each device on a DeviceNet network is associated with an .EDS file, containing all necessary information about the device. This file is used by the DeviceNet configuration tool during configuration of the network. Without this file the configuration tool can not get any information on how many parameters in this device and what are they. You can also download the EDS file from web site <http://www.iMultiCom.co.kr>.

It is necessary to import the .EDS-file in the DeviceNet configuration tool in order to configure the Modbus/DeviceNet gateway Communicator on the DeviceNet network. The properties for the Communicator module itself must then be configured from the DeviceNet configuration tool.

The EDS file please refers to Appendix C.

6. Technical Specification

Modbus communication is compliant to MODICOM Modbus Protocol PI-MBUS-300 Rev.J

DeviceNet communication is compliant to Group 2 Only Server defined in “DeviceNet Protocol Release 2.0 Errata 3”.

Baudrate Support:

DeviceNet port support 125kbit/s, 250kbit/s, 500kbit/s and support baudrate auto detected.

Modbus port support 9600 kbit/s and 19200 kbit/s and set by firmware.

Mechanical

Packaging:

Plastic packaging with removable connection to DIN-rail

Protection class: IP20

Dimensions:

100 mm x 70 mm x 25 mm [Rail connector not included]

Environmental

Relative Humidity

The product is designed for a relative humidity of 5 to 95% (non-condensing)

Temperature

Operating: -20°C to +55°C

Non-Operating: -50°C to +85°C

EMC

Emission

According to EN 50081-2:1993

Immunity

According to EN 61000-6-2:1999

Power Supply

DC 11V - 30V

Power Consumption

24V 60mA [Typically] 80mA [Maxim]

Appendix A: Modbus Communication

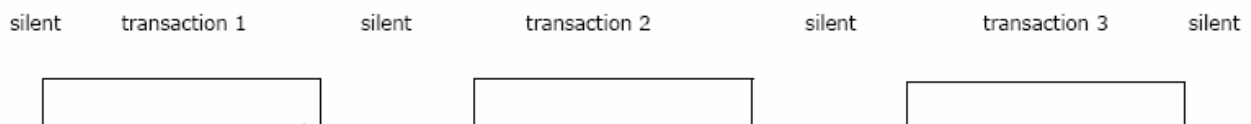
Introduction

When configured for Modbus protocols, the Modbus/DeviceNet gateway supports Modbus RTU Mode. Some basic knowledge is very helpful, as to understand how to use the gateway for configuration of the Modbus network.

The Modbus standard was developed by Modicon for communication between controllers and other devices. The transactions on the Modbus network are of master/slave type, and are named “query” and “response”. One single master sends the queries. All transactions on the network have got a frame structure where one part is common for both Modbus RTU and Modbus ASCII.

Modbus RTU

Each byte in the Modbus RTU message represents a hexadecimal value between 0 and 255. The frame looks exactly like the one mentioned earlier and there is always an interval of 3.5 silent characters between the frames. CRC (Cyclical Redundancy Check) is used for error checking. Multiple Modbus transactions would appear on the physical interface like this:



Modbus RTU Mode Transactions

Modbus commands

The following tables list all Modbus commands that are supported by the AB-C. For each command there is an explanation about what actions you as user need to take on the query and response.

01	Read Coil Status
02	Read Input Status
03	Read Holding Registers
04	Read Input Registers
05	Force Single Coil
06	Preset Single Register

07	Read Exception Status
11	Fetch Comm. Event Ctr
12	Fetch Comm. Event Log
15	Force Multiple Coils
16	Preset Multiple Registers
17	Report Slave ID
20	Read General Reference
21	Write General Reference
22	Mask Write 4X Register
23	Read/Write 4X Register
24	Read FIFO Queue

Table 1: Modbus Commands

The Modbus bus Commands in Bold are used in our gateway.

Read Holding Registers [03]		
Query	Starting Address	2 byte value where you enter the first address of the requested registers.
	No. of Points	2 byte value where you enter the number of registers to read.
Response	Byte Count	1 byte value where you enter the number of expected data bytes.
	Data	Data object where you enter the length of the received data and the destination address.

Table 2: Read Holding Registers

Preset Single register [06]		
Query	Data	Data object with 4 bytes. The fieldbus master should enter Register Address and Preset Data in these four bytes.
Response	Data	Data object with 4 bytes. The slave returns Register Address and Preset Data in these 4 bytes.

Table 3: Preset Single register

Appendix B: DeviceNet Profile

Vendor code: 632(0x278)

Product code: 9

Version: 1.1

The Modbus/DeviceNet gateway is a DeviceNet Group 2 only device.

DeviceNet Object Model

Object Interfaces:

Object Classes	Interfaces
Identify (0x01)	Message Router
DeviceNet (0x03)	Message Router
Assembly (0x04)	I/O Connection or Message router
Connection (0x05)	Message Router
Switcher (Class Id: 0x79)	I/O Connection or Message router

DeviceNet Message Types

CAN Identifier Field	Group 2 message Type
10xxxxxx111	Duplicate MAC ID check messages
10xxxxxx110	Unconnected explicit request message
10xxxxxx101	Master I/O poll command/COS message
10xxxxxx100	Master explicit request message
10xxxxxx010	Master COS acknowledge message
1111101101	Communication faulted request
10xxxxxx011	Slave explicit response message
01101xxxxxx	Slave COS message
01111xxxxxx	Slave POLL response message
1111101100	Communication faulted response

xxxxxx = node address.

Identity Object (0x01)

Class Attributes:

Attribute ID	Access Rule	Name	Data Type	Data Value
1	Get	Revision	UINT	1

Number of Instances: 1

Instance Attributes:

Attribute ID	Access Rule	Name	Data Type	Data Value
1	Get	Vendor ID	UINT	632
2	Get	Device Type	UINT	3
3	Get	Product Code	UINT	9
4	Get	Revision	STRUCT of:	
		Major Revision	UINT	1
		Minor Revision	UINT	1
5	Get	Status	WORD	0, not allocated 1, allocated
6	Get	Serial Number	UDINT	Unique 32 bit number
7	Get	Product Name	STRUCT of:	
		String length	USINT	24
		ASCII string	STRING	“Modbus/DeviceNet Gateway”

Common Services

Service Code	Implementation for		Service Name	Description of Service
	Class	Instance		
0E hex	No	Yes	Get_Attribute_Single	Returns the contents of the specified attribute.
05 hex	No	Yes	Reset	Invokes the Reset service for the device.

DeviceNet Object (0x03)

Class Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get	Revision	UINT	1

Number of Instances: 1

Instance Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
1	Get/Set	Node Address	USINT	0...63
2	Get	Baud Rate	USINT	0=125 K 1=250 K 2=500 K
3	Get	Bus-off Interrupt	BOOL	0=Keep in Error status when bus off detected 1=Reset CAN chip when bus off detected
5	Get/Special	Allocation Information Allocation Byte	Structure of: BYTE USINT	Allocation_byte* 63=master address

		Master's Node Address		255=unallocated
--	--	-----------------------	--	-----------------

*Allocation_byte

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
N/a	Ack_sup	Cyclic	COS	N/a	Strobe	Poll	Explicit

Common Services

Service Code	Implementation for		Service Name
	Class	Instance	
0E hex	Yes	Yes	Get_Attribute_Single
10 hex	No	Yes	Set_Attribute_Single
4Bhex	No	Yes	Allocate_Master/Slave_Connection_Set
4Chex	No	Yes	Release_Master/Slave_Connection_Set

Connection Object (0x05)

Class Attributes: None

Number of Instances: 2

Class Attributes:

Attribute ID	Access Rule	Name	Data Type	Data Value
1	Get	Revision	UINT	1

Instance 1 Attributes (Predefined slave explicit connection)

Attribute ID	Access Rule	Attribute Name	Data Type	Data Value
1	Get	State	USINT	0=nonexistent 1=Configuring 3=established 4=Timeout
2	Get	Instance_type	USINT	0=explicit message
3	Get	TransportClass_trigger	BYTE	83hex, server, transport class 3
4	Get	Produced_connection_id CAN header field 11 bit message	UINT	10xxxxxx011 xxxxxx=node address
5	Get	Consumed_connection_id CAN header field 11 bit message	UINT	10xxxxxx100 xxxxxx=node address
6	Get	Initial_comm_characteristics	USINT	22hex 2=slave's explicit message (produce, source) 1=master's explicit request (consume, destination)
7	Get	Produced_connection_size	UINT	
8	Get	Consumed_connection_size	UINT	
9	Get/Set	Expected_packet_rate	UINT	2500 ms (default), with timer resolution of

				10 ms
12	Get/Set	Watchdog_timeout_action	USINT	1=auto delete (other value 3=deferred)
13	Get	Produced_connection_path_length	USINT	0
14	Get	Produced_conction_path	EPATH	Null
15	Get	Consumed_connection_path_length	UINT	0
16	Get	Consumed_conection_path	EPATH	Null

Instance 2 Attributes (Predefined slave Poll message connection)

Attribute ID	Access Rule	Attribute Name	Data Type	Data Value
1	Get	State	USINT	0=nonexistent 1=configuring 3=established 4=timeout
2	Get	Instance_type	USINT	1=I/O message
3	Get	TransportClass_trigger	USINT	0x82, server, transport class 2 (if allo_choice!=POLL and Ack_sup=1, then value=0x80)
4	Get	Produced_connection_id	UINT	01111xxxxx xxxxxx=node address
5	Get	Consumed_connection_id	UINT	10xxxxxx101 xxxxxx=node address
6	Get	Initial_comm_characteristics	BYTE	21hex 2=master's poll response (produced across GP2) 1=master's poll command (consumed across GP2)
7	Get	Produced_connection_size	UINT	8
8	Get	Consumed_connection_size	UINT	8
9	Get/Set	Expected_packet_rate	UINT	0 milliseconds
12	Get	Watchdog_timeout_action	USINT	0=timeout (default)
13	Get	Produced_connection_path_length	USINT	7
14	Get/Set	Produced_connection_path Logical Segment, Class Class Number Logical Segment, Instance Instance Number Logical Segment, Attribute Attribute Number	Struct of USINT USINT USINT USINT USINT USINT	21hex 0004hex 24hex assy inst# 30hex 03hex
15	Get	Consumed_connection_path_length	USINT	7
16	Get/Set	Consumed_conection_path	Struct. of	

		Logical Segment, Class	USINT	21hex
		Class Number	USINT	0004hex
		Logical Segment, Instance	USINT	24hex
		Instance Number	USINT	Assembly instance number
		Logical Segment, Attribute	USINT	30hex
		Attribute Number	USINT	03hex

Common Services

Service Code	Implementation for		Service Name
	Class	Instance	
05hex	No	Yes	Reset
0E hex	No	Yes	Get_Attribute_Single
10 hex	No	Yes	Set_Attribute_Single

Assembly Objects (0x04)

Class Attributes

Attribute ID	Access Rule	Name	DeviceNet Data Type	Data Value
2	Get	Max Instance	UINT	190

Number of Instances: 12 input assemblies, 3 output assemblies

Instance Attributes

Attribute ID	Access Rule	Attribute Name	Data Type	Data Value
3	Get	Data	Array of Byte	Input data

Common Services

Service Code	Implementation for		Service Name
	Class	Instance	
0E hex	Yes	Yes	Get_Attribute_Single

Application Object: Switch Class (0x79)

Class Attributes: None

Number of Instances: 1

Class Attributes:

Attribute ID	Access Rule	Name	Data Type	Data Value
1	Get	Revision	UINT	1

Common Services

Service Code	Implementation for		Service Name
	Class	Instance	
0E hex	No	Yes	Get_Attribute_Single
10 hex	No	Yes	Set_Attribute_Single

Appendix C: EDS file

\$ DeviceNet Electronic Data Sheet

\$ Electronic Data Sheet

\$ Copyright (C) 2005–2007

[File]

DescText = "Modbus-DeviceNet Gateway";

CreateDate = 03-01-05;

CreateTime = 08:57:44;

ModDate = 09-08-07;

ModTime = 14:58:25;

Revision = 1.2;

[Device]

VendCode = 1016;

VendName = "MultiCom Co. Ltd.";

ProdType = 12;

ProdTypeStr = "Communication Adapter";

ProdCode = 12;

MajRev = 1;

MinRev = 2;

ProdName = "Modbus-DeviceNet Gateway";

Catalog = "MD21";

[IO_Info]

Default = 0x0001;

PollInfo = 0x0001, 7, 6;

Input1=

8, 0, 0x0001,

"Run-time measurements and State",

6, "20 04 24 74 30 03",

"8 bytes";

Input2=

16, 0, 0x0001,

"Network Input 2",

6, "20 04 24 75 30 03",

"16 Bytes";

Input3=
32, 0, 0x0001,
"Network input 3",
6, "20 04 24 76 30 03",
"32 Bytes";

Input4=
64, 0, 0x0001,
"Network input 4",
6, "20 04 24 77 30 03",
"64 Bytes";

Input5=
96, 0, 0x0001,
"Network Input 5",
6, "20 04 24 78 30 03",
"96 Bytes";

Input6=
128, 0, 0x0001,
"Network Input 6",
6, "20 04 24 79 30 03",
"128 Bytes";

Input7=
160, 0, 0x0001,
"Input7",
6, "20 04 24 7A 30 03",
"Input 7 160bytes";

Output1=
8, 0, 0x0001,
"Network Output 1",
6, "20 04 24 80 30 03",
"8 Bytes";

Output2=
16, 0, 0x0001,
"Network Output 2",
6, "20 04 24 81 30 03",
"16 Bytes";

Output3=
32, 0, 0x0001,


```
"Network Output 3",  
6, "20 04 24 82 30 03",  
"32 Bytes";
```

```
Output4=  
64, 0, 0x0001,  
"Network Output 4",  
6, "20 04 24 83 30 03",  
"64 Bytes";
```

```
Output5=  
96, 0, 0x0001,  
"Network Output 5",  
6, "20 04 24 84 30 03",  
"96 Bytes";
```

```
Output6=  
112, 0, 0x0001,  
"Network Output 6",  
6, "20 04 24 85 30 03",  
"112 Bytes";
```

```
[ParamClass]  
MaxInst      = 4;  
Descriptor   = 0x0001;  
CfgAssembly = 0;
```

```
[Params]
```

```
Param1=  
0,  
6, "20 A2 24 01 30 64",  
0x0030, 4, 1,  
"Modbus_status",  
"" ,  
"Status of Modbus, OK or The time out error Command No.",  
0, 1, 1,  
1, 1, 1, 0,  
0, 0, 0, 0,  
0;
```

```
Param2=  
0,  
6, "20 A2 24 01 30 65",  
0x0012, 8, 1,
```

```

"Input bytes",
"",
"Number of poll input connection bytes",
0, 255, 0,
1, 1, 1, 0,
0, 0, 0, 0,
0;

```

```

Param3=
0,
6, "20 A2 24 01 30 66",
0x0012, 8, 1,
"Output bytes",
"",
"Number of poll output connection bytes",
0, 255, 0,
1, 1, 1, 0,
0, 0, 0, 0,
0;

```

```

Param4=
0,
6, "20 A2 24 01 30 67",
0x0002, 8, 1,
"Mod_Output_Ctrl",
"",
"Control of Modbus Output Commands",
0, 1, 1,
1, 1, 1, 0,
0, 0, 0, 0,
0;

```

[EnumPar]

```

Param2=
"8",
"16",
"32",
"64",
"96",
"128",
"160";

```

```

Param3=
"8",

```

"16",
"32",
"64",
"96",
"112";

Param4=
"Disable Modbus Output",
"Enable Modbus Output";

[Groups]