DVPSCM Series

DVPSCM12-SL Serial Communication Module
DVPSCM52-SL BACnet MS/TP Slave Communication Module

Operation Manual

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

DVP-0186520-02

⚠ Warning

- ✓ This operation manual provides the introduction of specifications, installation, basic operation and setting, and contents related to communication protocols.
- ✓ The module is an open-type device. It has to be installed in the distribution box which is dust-proof, moisture-proof, and free from shock and vibration. To prevent people who are not technicians from operating the module or to prevent accidents from damaging the module, additional protection measures are necessary (eg the distribution box has to be opened with a special tool or with a key). Besides, do not touch any terminal when the power supply is on.
- ✓ Be sure to read this manual carefully, and operate the module according to the instructions lest the product should be damaged or the staff should be hurt.

🕮 Contents

1. Introduction 1

1.1 Functions 1

1.2 Specifications 1

2. Product Appearance and Product Profile 3

2.1 Dimensions 3

2.2 Product Profile 3

2.3 LED Indicators 4

2.4 Definitions of Pins on RS-485/RS-482 Communication Ports 4

3. Installation and Wiring 5

3.1 Installation 5

4. Control Registers (CR) 7

4.1 Table of Control Registers 7

4.2 Contents of Control Registers 8

4.3 Right-side Module Numbers 13

5. Rapid Start 15

6. Introduction of SCMSoft 23

6.1 SCM Project 23

6.2 COM PORT Setting 23

6.3 UD Link (User-defined Link) 24

6.3.1 TX Packets and RX Packets 25

6.3.2 Command 27

6.3.3 Sequence 28

6.4 MODBUS Advance 29

6.5 BACnet MS/TP Slave Function (Supported by DVPSCM52-SL) 29

6.5.1 BACnet Parameters 29
6.5.2 BACnet Object 30
7. Application 31
7.1 MODBUS 31
7.1.1 Connection between the MODBUS Slave and the Delta Product 32
7.1.2 Connection between the MODBUS Master and the Delta Product 33
7.2 Connecting to WPLSoft 41
7.3 RS-485 42
7.3.1 Connecting to the Electricity Meter 42
7.4 BACnet MS/TP Slave Function (Supported by DVPSCM52-SL) 61
8. Error Flags 65

1. Introduction

Thanks for using Delta communication module DVPSCM12/52-SL. In order to ensure the correct installation and operation of this product, please read the manual before you use the module.

DVPSCM12/52-SL is a serial communication module. It supports MODBUS, and the UD Link (user-defined format of RS-485). Besides, it can be used as a RS-422 communication port or RS-485 communication port through which a program is uploaded or downloaded. SCMSoft, the setting software of DVPSCM12/52-SL, is built in Delta communication software DCISoft. Please download DCISoft_v1.08 from Delta website.

DVPSCM52-SL is a slave communication module using a building automation control network communication protocol. It is equipped with all the functions of DVPSCM12-SL, and supports the BACnet MS/TP slave communication protocol. It can read/write the BV values or AV values from/into a BACnet MS/TP master. SCMSoft, the setting software of DVPSCM52-SL, is built in Delta communication software DCISoft. Please download DCISoft_v1.08 from Delta website.

1.1 Functions

- It provides RS-422 and RS-485 communication ports (COM1 & COM2).
- RS-422/RS-485 communication and the power supply are isolated from each other.
- There are two built-in 120Ω terminal resistors and switches.
- Each communication port can connect to at most 32 devices.
- It has the MODBUS data exchange functions (MODBUS Advance).
- It has the user-defined communication protocol, and the process planning function (UD Link).
- DVPSCM52-SL supports the BACnet MS/TP slave functions, and can connect to a superior device.
- The MPUs it supports: DVPSA2 (V1.0), DVPSX2 (V1.2), DVPSV (V2.2), DVPSE (V1.0)，EH2-L (V2.20), and EH3-L (V1.00) series.

1.2 Specifications

- The RS-485/RS-422 interface

Item	Specifications
Terminal	European terminal blocks with spring plugs
Transmission speed	1,200, 2,400, 4,800, 9,600, 19,200, 38,400, 57,600, 76,800, 115,200, 230,400, and 460,800 bps
Communication format	Stop bit: 1, 2; Parity bit: None, Odd, Even; Data bit: 7, 8
Communication protocol	MODBUS ASCII/RTU, UD Link, and BACnet MS/TP slave (supported by DVPSCM52-SL)

- Environmental specifications

Item	Specifications
Noise immunity	ESD (IEC 61131-2, IEC 61000-4-2): 8 kV Air Discharge EFT (IEC 61131-2, IEC 61000-4-4): ±1 KV (Communication I/O) CS (IEC 61131-2, IEC 6100-4-6): 0.15 ~ 80 MHz, 3 Vrms RS (IEC 61131-2, IEC 61000-4-3): 80 ~ 100 MHz, 10 V/m, 1.4 ~ 2.0 GHz
Operating/Storage temperature	Operation: 0 ~ 55°C (temperature); 50 ~ 95% (humidity); pollution degree 2 Storage: -25 ~ 70°C (temperature), 5 ~ 95% (humidity)
Shock/Vibration resistance	International standard norms IEC61131-2, IEC68-2-6 (TEST Fc) / IEC61131-2 & IEC 68-2-27 (TEST Ea)
Standard	CE cULus

- Electrical Specifications

Item	Specifications
Supply voltage	24 V DC (supplied by the internal bus through the MPU)
Power consumption	1.5 W
Insulation voltage	2500 VDC
Weight	Approximately 95 g

- BACnet Protocol Implementation Statement

◆ Introduction of the standard BACnet device

Model	Introduction
DVPSCM52-SL	BACnet Application Specific Controller (B-ASC)

◆ BIBBs which are supported

Model	BIBBs	BIBB name
DVPSCM52-SL	DS-RP-B	Data Sharing-ReadProperty-B
	DS-WP-B	Data Sharing-WriteProperty-B
	DM-DDB-B	Device Management-DynamicDeviceBinding-B
	DM-DOB-B	Device Management-DynamicObjectBinding-B
	DM-DCC-B	Device Management-DeviceCommunicationControl-B
	DS-RPM-B	Data Sharing-ReadPropertyMultiple-B
	DS-WPM-B	Data Sharing-WritePropertyMultiple-B

◆ Objects which are supported

Model	Object	Creation	Deletion
DVPSCM52-SL	Analog value	Not supported	Not supported
	Binary value	Not supported	Not supported
	Device	Not supported	Not supported

◆ Data link layer options

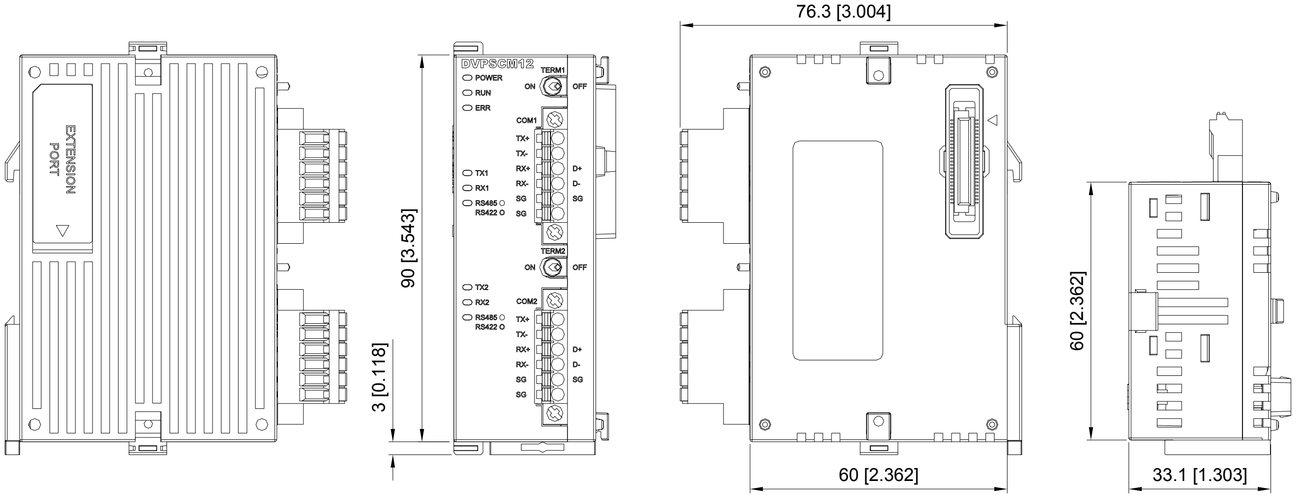
Model	Data link	Baud rates which are supported
DVPSCM52-SL	MS/TP Slave	9600, 19200, 38400, 76800

◆ Character set which is supported

Model	Character set
DVPSCM52-SL	ANSI X3.4

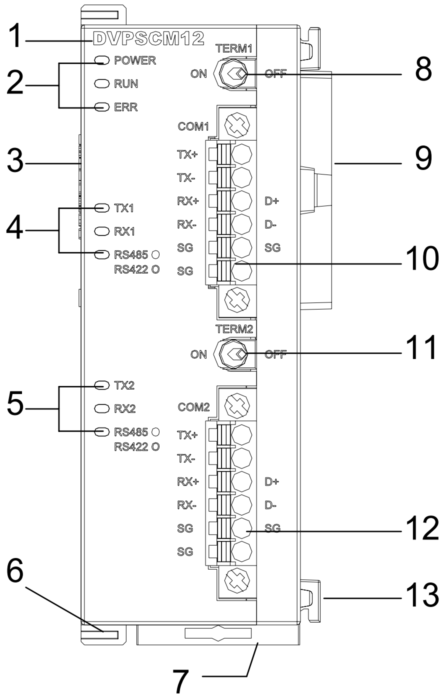
2. Product Appearance and Product Profile

2.1 Dimensions



Unit: mm [inches]

2.2 Product Profile

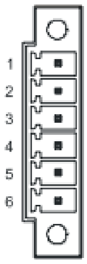


1. Model name	8. Switch for terminal resistor 1
2. POWER, RUN, ERR LED indicators	9. Extension port for the MPU/left-side module
3. Extension port for the left-side module	10. RS-485⁄RS-422 communication port 1
4. TX1, RX1, RS-485⁄RS-422 LED indicators	11. Switch for terminal resistor 2
5. TX2, RX2, RS-485⁄RS-422 LED indicators	12. RS-485⁄RS-422 communication port 2
6. Fixing clip for the left-side module	13. Positioning hole for the I/O module
7. DIN rail clip	

2.3 LED Indicators

LED indicator	Status		Indication	Disposal
POWER	Green light	On	Normal power supply	No action is required.
		Off	No power supply	Check whether the power supply is on.
RUN	Green light	On	The status of the SCM module is RUN.	No action is required.
		Off	The status of the SCM module is STOP.	No action is required.
ERR	Red light	On	Hardware error	Contact the original factory.
		Flash	There is an error in system settings or communication.	Restore it to the factory default.
		Off	No error	No action is required.
TX1∕TX2	Orange light	Flash	Data is being transmitted through the RS-485∕RS-422 port.	No action is required.
		Off	No data is being transmitted through the RS-485∕RS-422 port.	No action is required.
RX1∕RX2	Orange light	Flash	Data is being received through the RS-485∕RS-422 port.	No action is required.
		Off	No data is being received through the RS-485∕RS-422 port.	No action is required.
RS-485∕RS-422	Green light	On	RS-485 mode	No action is required.
		Off	RS-422 mode	No action is required.

2.4 Definitions of Pins on RS-485/RS-482 Communication Ports

Terminal block	Terminal no.	RS-485	RS-422
	1		TX+
	2		TX-
	3	D+	RX+
	4	D-	RX-
	5	SG	SG
	6		SG

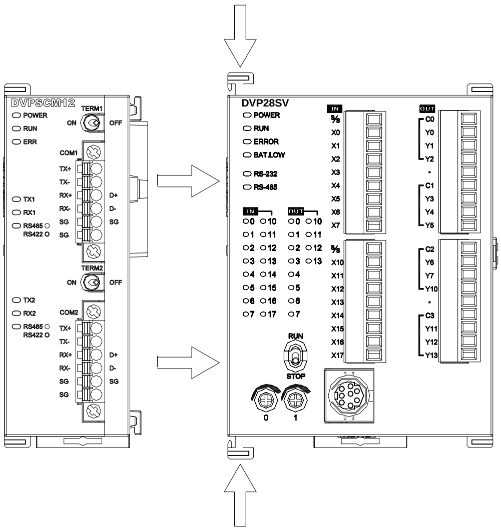
3. Installation and Wiring

This chapter introduces how an SCM module connects to an MPU.

3.1 Installation

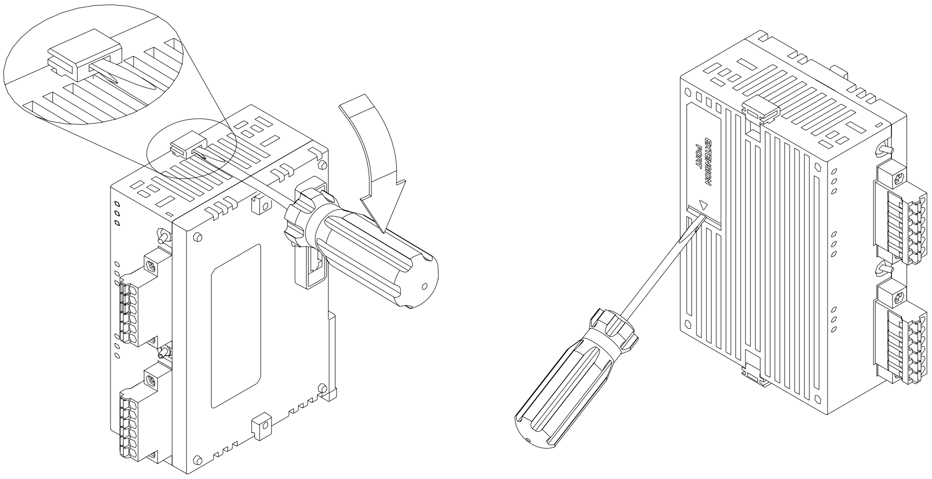
The MPU of the PLC connects to the SCM module.

- Adjust the clips connecting to the left-side module on the MPU.
- Direct the I/O module to the interface on the MPU, and combine the I/O module with the MPU as shown in the figure below.
- Tighten the clips connecting to the left-side module on the MPU.



The SCM module connects to other I/O modules.

- Before the SCM module connects to the inferior I/O module, the fixing clip for the I/O module has to be loosened by the screwdriver, and the side cover has to be opened.



MEMO

4. Control Registers (CR)

4.1 Table of Control Registers

CR#	Attribute	Name of the register	Description
0	R	Model code	The code is set up by the system. Model code of DVPSCM12-SL=H’4041 Model code of DVPSCM52-SL=H’4042
1	R	Firmware version	The firmware version is displayed in a hexadecimal value. For example, H’0100 indicates that the firmware version is V1.00.
2	Reserved		
3	R/W	Group number triggered by COM1 UD Link	The group number triggered by COM1 UD Link
4	R/W	Reference address of the data senT through COM1 in UD Link	It is used when COM1 UD Link chooses “Base+Offset”. “Reference data register+Offset” defines the actual source device for the data sending.
5	R/W	Reference address of the data received through COM1 in UD Link	It is used when COM1 UD Link chooses “Base+Offset”. “Reference data register+Offset” defines the actual source device for the data receiving.
6	Reserved		
7	R/W	Group number triggered by COM2 UD Link	The Group number triggered by COM2 UD Link
8	R/W	Reference address of the data sent through COM2 in UD Link	It is used when COM2 UD Link chooses “Base+Offset”. “Reference data register+Offset” defines the actual source device for the data sending.
9	R/W	Reference address of the data received through COM2 in UD Link	It is used when COM2 UD Link chooses “Base+Offset”. “Reference data register+Offset” defines the actual source device for the data receiving.
10	R	Module status	RUN or STOP
11~19	R	Error Flag	The flag for an error in the module
20~27	R	UD Link status	The execution status of UD Link
28~29	Reserved		
30	R/W	Triggering the UD Link sequence	0: Not triggered, 1~254: Number of times the UD Link sequence is triggered 255: Always triggered
31	R/W	Triggering the data exchange through COM1 to read bits or words	High byte: bit; Low byte: word 0: Not triggered; 1: Triggered once; 2: Always triggered
32	R/W	Triggering the data exchange through COM2 to read bits or words.	High byte: bit; Low byte: word 0: Not triggered; 1: Triggered once; 2: Always triggered
33	R/W	Triggering the data exchange through COM1 to write bits or words.	High byte: bit; Low byte: word 0: Not triggered; 1: Triggered once; 2: Always triggered
34	R/W	Triggering the data exchange through COM2 to write bits or words.	High byte: bit; Low byte: word 0: Not triggered; 1: Triggered once; 2: Always triggered
35~36	R/W	Selecting the “reading bits through COM1” checkbox	Bit = 0: Disabling the function of reading bits through COM1. Bit = 1: Enabling the function of reading bits through COM1.
37~38	R/W	Selecting the “reading words through COM1” checkbox	Bit = 0: Disabling the function of reading words through COM1. Bit = 1: Enabling the function of reading words through COM1.

CR#	Attribute	Name of the register	Description
39~40	R/W	Selecting the "reading bits through COM2" checkbox	Bit = 0: Disabling the function of reading bits through COM2. Bit = 1: Enabling the function of reading bits through COM2.
41~42	R/W	Selecting the "reading words through COM2" checkbox	Bit = 0: Disabling the function of reading words through COM2. Bit = 1: Enabling the function of reading words through COM1.
43~44	R/W	Selecting the "writing bits through COM1" checkbox	Bit = 0: Disabling the function of writing bits through COM1. Bit = 1: Enabling the function of writing bits through COM1.
45 ~ 46	R/W	Selecting the "writing words through COM1" checkbox	Bit = 0: Disabling the function of writing words through COM1. Bit = 1: Enabling the function of writing words through COM1.
47~48	R/W	Selecting the "writing bits through COM2" checkbox	Bit = 0: Disabling the function of writing bits through COM2. Bit = 1: Enabling the function of writing bits through COM2.
49~50	R/W	Selecting the "writing words through COM2" checkbox	Bit = 0: Disabling the function of writing words through COM2. Bit = 1: Enabling the function of writing words through COM2.
51~115	Reserved		
116	R/W	Sending the MODBUS command	1: Enabling the sending After the sending of the MODBUS command is complete, CR#116 is reset to 0.
117	R/W	Processing status of the MODBUS command	0: Not yet received; 1: Processing; 2: Received; 3: Reception failure
118	R/W	Destination of the MODBUS command	1: COM1, 2: COM2
119	R/W	Length of the MODBUS command	Setting the length of the MODBUS command
120~249	R/W	Contents of the MODBUS command	The space for storing the MODBUS command which is sent/received

4.2 Contents of Control Registers

CR#0: Model code

[Description]

1. Model code of DVPSCM12-SL=H'4041
2. Model code of DVPSCM52-SL=H'4042
3. The model code can be read out in the program to judge whether the I/O module exists.

CR#1: Firmware version

[Description]

The firmware version is displayed in a hexadecimal value, for example, H'0100 indicates that the firmware version is V1.00.

CR#3：Group number triggered by COM1 UD Link

[Description]

Enter the Group number edited in UD Link. The data is transmitted through COM1.

When the register is set to 1, it indicates that the content of Group ID#1 is triggered and executed. The register is reset to 0 after the execution is complete, and CR#26 is set to 1.

Default = 0, no Group is triggered.

CR#4: Reference address of the data sent through COM1 in UD Link

[Description]

This control register is used when COM1 Protocol chooses “UD Link”, and “Base+Offset” is chosen in the variable editing message.

The input value is the data register number, and the packet editor defines the actual source device for the data sending.

If “Base+Offset” is chosen in the packet editor, “Base+Offset” defines the actual source device for the data sending.

Example: Enter 1 in CR#4 → D1,

Choose “Base+Offset” in the packet editor, and enter 10 for the offset and 2 for the length → (R (Base+Offse [10], 2)).

D (1+10) indicates reading two bytes in D11.

This control register is used when “Base+Offset” is set in the packet editor.

CR#5: Reference address of the data received through COM1 in UD Link

[Description]

This control register is used when COM1 Protocol chooses “UD Link”, and “Base+Offset” is chosen in the variable editing message.

The input value is the data register number, and the packet editor defines the actual source device for the data receiving.

If “Base+Offset” is chosen in the packet editor, “Base+Offset” defines the actual source device for the data receiving.

CR#7: Group number triggered by COM2 UD Link

[Description]

Please refer to the description of CR#3.

CR#8: Reference address of the data sent through COM2 in UD Link

[Description]

Please refer to the description of CR#4.

CR#9: Reference address of the data received through COM2 in UD Link

[Description]

Please refer to the description of CR#5.

CR#10: Module status

[Description]

The PLC controls RUN/STOP status of the SCM module.

CR#11~19: Error flag

[Description]

With regard to the error flag in the SCM module, please refer to chapter 8.

CR#20~27：UD Link status

[Description]

The execution status of UD Link

CR#30: Triggering the UD Link sequence

[Description]

High byte: COM1; Low byte: COM2

Enter directly the number of times the UD Link sequence is triggered.

0: Not triggered; 1~254: Number of times he UD Link sequence is triggered; 255 (H' FF): Always triggered

CR#31: Triggering the data exchange through COM1 to read bits or words

[Description]

High byte: COM1 Bit; Low byte: COM1 Word

0: Not triggered; 1: Triggered once; 2: Always triggered

		COM1 Word		
		Not triggered	**Triggered once**	**Always triggered**
COM1 Bit	**Not triggered**	H' 0000	H' 0001	H' 0002
	Trigger once	H' 0100	H' 0101	H' 0102
	Always triggered	H' 0200	H' 0201	H' 0202

CR#32: Triggering the data exchange through COM2 to read bits or words

[Description]

High byte: COM2 Bit; Low byte: COM2 Word

0: Not triggered; 1: Triggered once; 2: Always triggered

Please refer to the table in the description of CR#31 for hexadecimal values.

CR#33: Triggering the data exchange through COM1 to write bits or words

[Description]

High byte: COM1 Bit; Low byte: COM1 Word

0: Not triggered; 1: Triggered once; 2: Always triggered

Please refer to the table in the description of CR#31 for hexadecimal values.

CR#34: Triggering the data exchange through COM2 to write bits or words

[Description]

High byte: COM2 Bit; Low byte: COM2 Word

0: Not triggered; 1: Triggered once; 2: Always triggered

Please refer to the table in the description of CR#31 for hexadecimal values.

CR#35~36: Selecting the “reading bits through COM1” checkbox

[Description]

Select the “reading bits through COM1” checkbox. The SCM module can read at most 32 groups of data (No.1~No.32).

CR#	CR35															
Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
No.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

CR#35: No.16~No.1; CR#36: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#37~38: Selecting the “reading words through COM1” checkbox

[Description]

Select the “reading words through COM1” checkbox. The SCM module can read at most 32 groups of data (No.1~No.32).

CR#37: No.16~No.1; CR#38: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#39~40: Selcting the “reading bits through COM2” checkbox

[Description]

Select the “reading bits through COM2” checkbox. The SCM module can read at most 32 groups of data (No.1~No.32).

CR#39: No.16~No.1; CR#40: No.32~No.17

0: Enable the function; 1: Disable the function

CR#41~42: Selecting the “reading words through COM2” checkbox

[Description]

Select the “reading words through COM2” checkbox. The SCM module can read at most 32 groups of data (No.1~No.32).

CR#41: No.16~No.1; CR#42: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#43~44: Selecting the “writing bits through COM1” checkbox

[Description]

Select the “writing bits through COM1” checkbox. The SCM module can write at most 32 groups of data (No.1~No.32).

CR#	CR43															
Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
No.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

CR#43: No.16~No.1; CR#44: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#45~46: Selecting the “writing words through COM1” checkbox

[Description]

Select the “writing words through COM1” checkbox. The SCM module can write at most 32 groups of data (No.1~No.32).

CR#45: No.16~No.1; CR#46: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#47~48: Selecting the “writing bits through COM2” checkbox

[Description]

Select the “writing bits through COM2” checkbox. The SCM module can write at most 32 groups of data (No.1~No.32).

CR#47: No.16~No.1; CR#48: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#49~50: Selecting the “writing words through COM2” checkbox

[Description]

Select the “writing words through COM1” checkbox. The SCM module can write at most 32 groups of data (No.1~No.32).

CR#49: No.16~No.1; CR#50: No.32~No.17

0: Enabling the function; 1: Disabling the function

CR#116: Sending the MODBUS command

[Description]

The SCM module sends the MODBUS command.

1: Enabling the sending.

After the sending of the MODBUS command is complete, CR#116 is reset to 0.

CR#117: Processing status of the MODBUS command

[Description]

0: Not yet received; 1: Processing; 2: Received; 3: Reception failure

CR#118: Destination of the MODBUS command

[Description]

Designate the sending port of the SCM module. 1: COM1; 2: COM2

CR#119：Length of the MODBUS command

[Description]

The length of the MODBUS command which is sent (in a hexadecimal value) depends on the start content of CR120.

CR#120~249：Contents of the MODBUS command

[Description]

The contents of the MODBUS command which is sent (in a hexadecimal value)

4.3 Right-side Module Numbers

After the SCM module is installed, the related functions of the I/O module are controlled by the PLC program. The PLC provides two instructions (FROM and TO) to read/write the data from/into the control register of the special module.

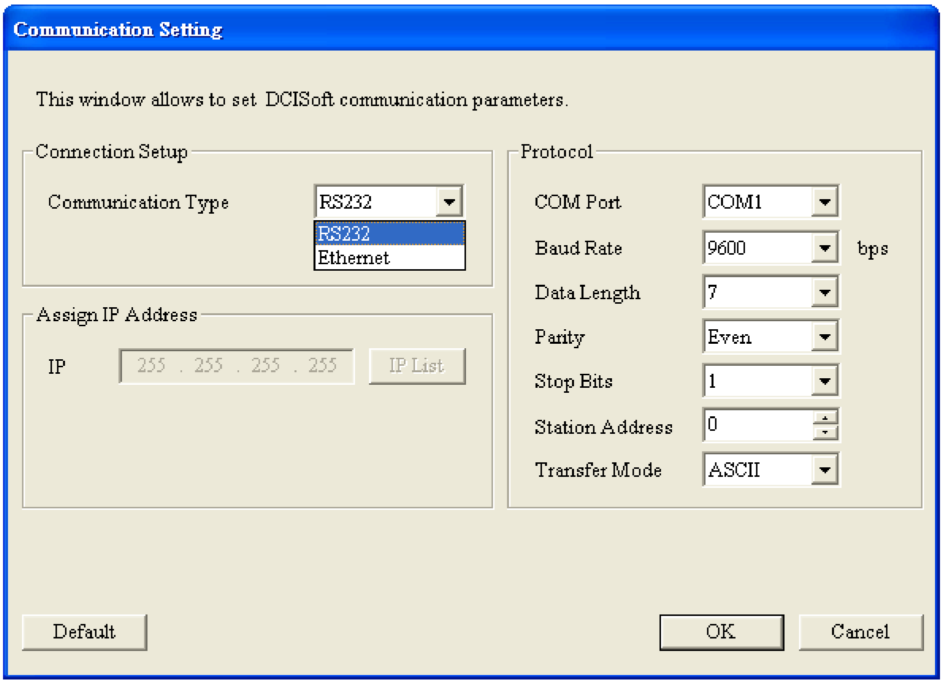
Left-side module numbers: Each left-side/right-side module connecting to the MPU of the PLC has a number in order that the module can be recognized when the user writes the PLC program. For the left-side module, the first I/O module connecting to the left side of the MPU of the PLC is numbered K100, the second module is numbered K101, the third module is numbered K102, and the others are numbered by analogy. At most 8 modules can connect to the MPU of the PLC.

5. Rapid Start

This chapter introduces how to execute MODBUS RS-485/RS-422 communication through the communications ports on the SCM module.

【Communication setting】

Open DCISoft, click “Tools” and choose “Communication Setting”. The user can choose the communication port, and set the information related to RS-232. If an Ethernet module (DVPEN01-SL) is used with the SCM module, the user can select “Ethernet” in “Communication Type” box to upload/download the program.



Communication Setting

This window allows to set DCISoft communication parameters.

Connection Setup

Communication Type: RS232 (RS232, Ethernet)

Assign IP Address

IP: 255 . 255 . 255 . 255

Protocol

COM Port: COM1

Baud Rate: 9600 bps

Data Length: 7

Parity: Even

Stop Bits: 1

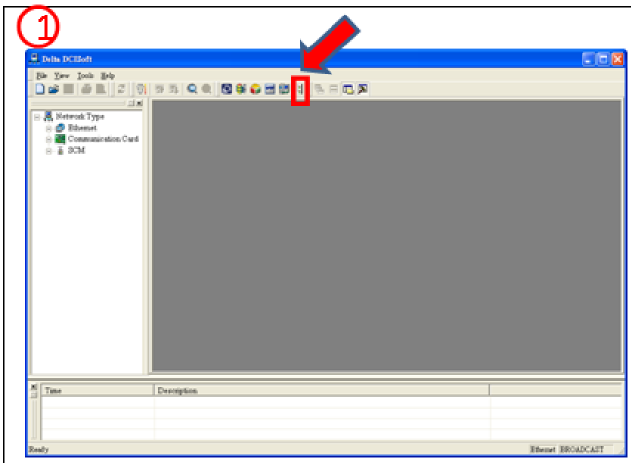
Station Address: 0

Transfer Mode: ASCII

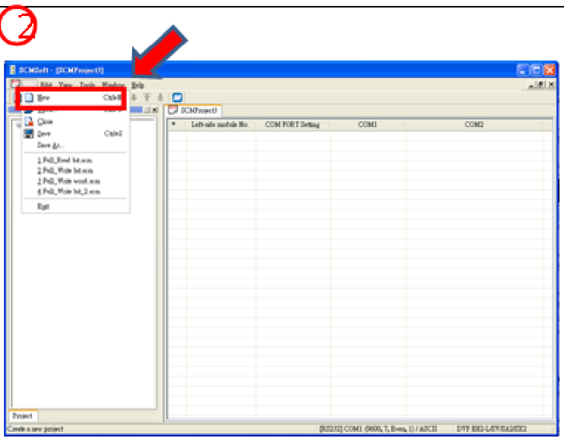
【Opening a SCM project and “MODBUS Advance”】

Click “SCMSoft” in DCISoft to open the setting page. Then, click “New Project” in SCMSoft to establish a SCM project. Finally, click “MODBUS Advance Wizard” to open the setting page for the reading/writing.

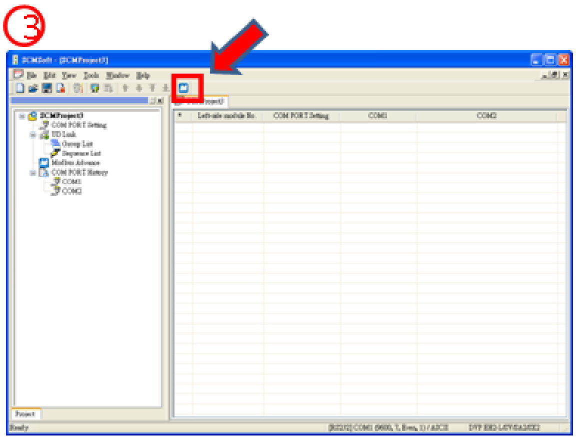
①



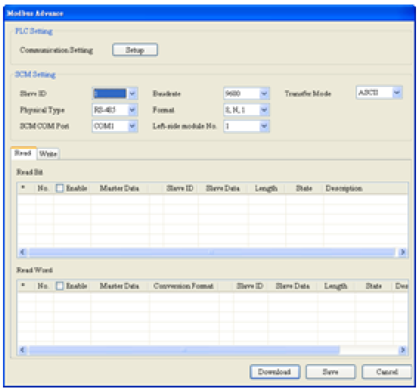
②



③



④



【Setting “MODBUS Advance”】

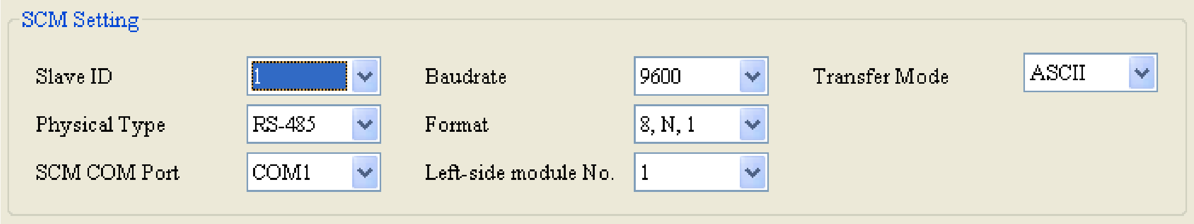
In order to expedite the communication using MODBUS, SCMSoft provides “MODBUS Advance Wizard”. The user only needs to designate the registers for the data sending and the data receiving, or the absolute positions. The settings will be downloaded to the SCM module through the communication port chosen by the user. After the flag is enabled, the designated reading and writing are complete. The following are the steps of setting the wizard.

(1) MODBUS Advance－PLC Setting

Click “Setup” to set the communication between the MPU of the PLC and SCMSoft. If the setting has been completed at【Communication setting】, the user does not have to set the communication here again.

(2) SCM Setting

When setting the communication format of the communication port on the SCM module, the user can designate the left-side module number, and the communication port, and set the station address, the baudrate, the physical type, the transfer mode, and the format.

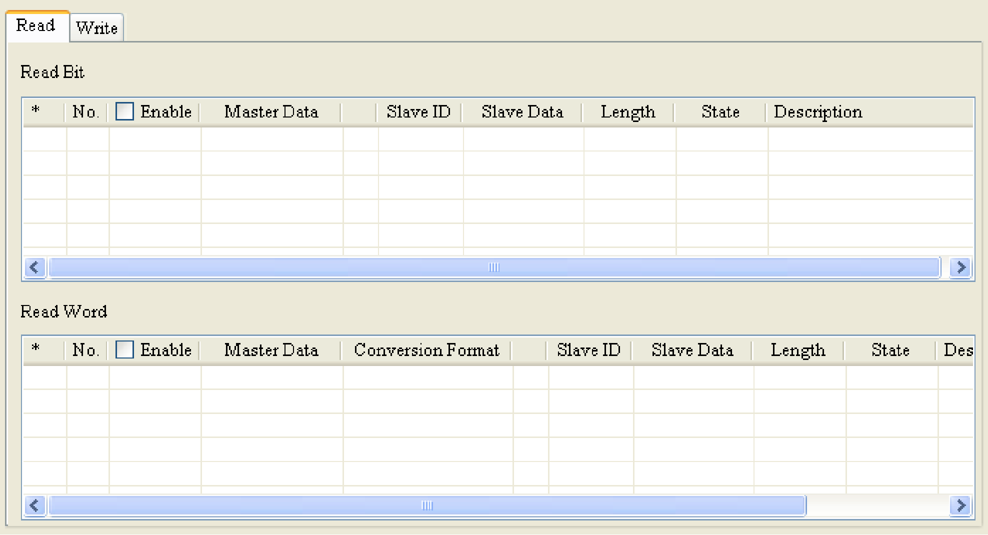


SCM Setting

Slave ID	1	Baudrate	9600	Transfer Mode	ASCII
Physical Type	RS-485	Format	8, N, 1		
SCM COM Port	COM1	Left-side module No.	1		

(3) MODBUS Advance – Reading/Writing

Set "Read Bit"/"Read Word" and "Write Bit"/ "Write Word".



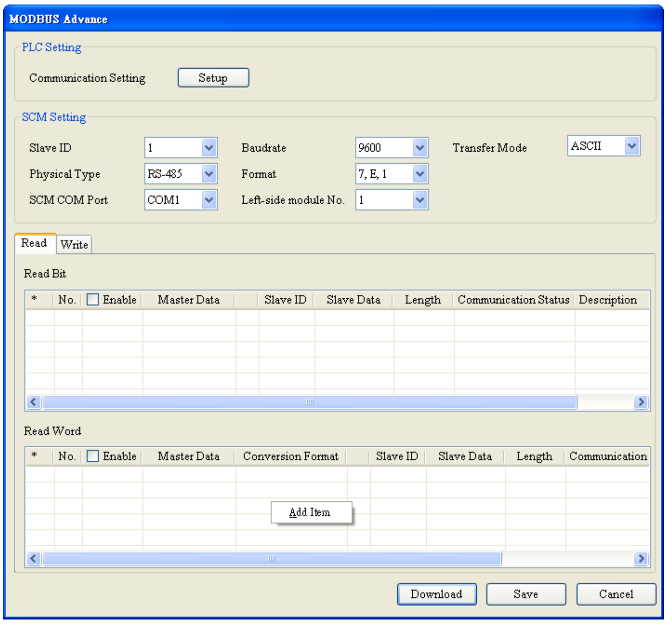
Read Bit

*	No.	□ Enable	Master Data	Slave ID	Slave Data	Length	State	Description

Read Word

*	No.	□ Enable	Master Data	Conversion Format	Slave ID	Slave Data	Length	State	Des

Press the right key of the mouse, and click "Add Item" to increase bits and words. The bits are listed in the upper column, and the words are listed in the lower column.



MODBUS Advance

PLC Setting

Communication Setting

SCM Setting

Slave ID	1	Baudrate	9600	Transfer Mode	ASCII
Physical Type	RS-485	Format	7, E, 1		
SCM COM Port	COM1	Left-side module No.	1		

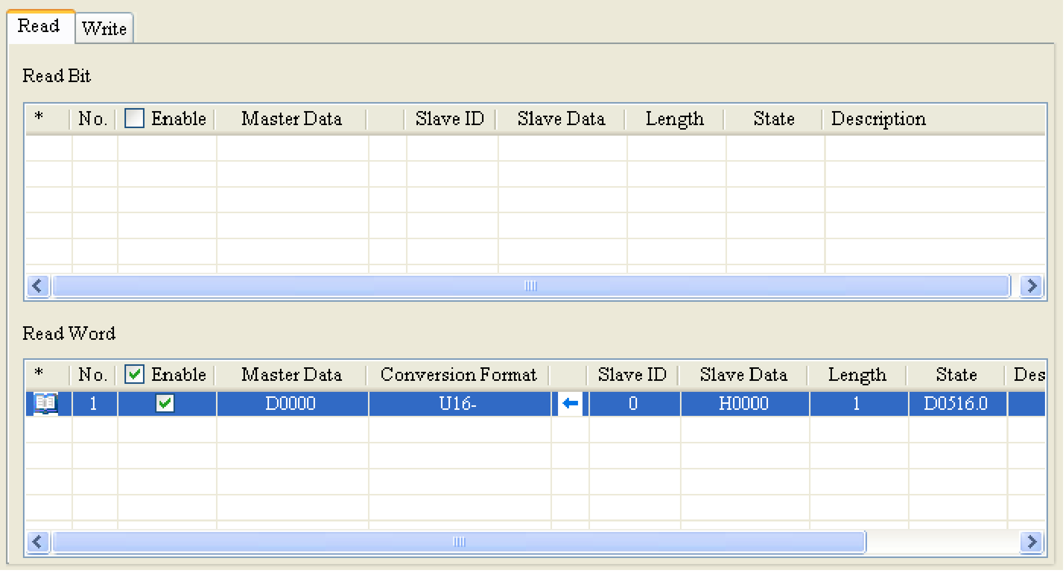
Read Bit

*	No.	□ Enable	Master Data	Slave ID	Slave Data	Length	Communication Status	Description

Read Word

*	No.	□ Enable	Master Data	Conversion Format	Slave ID	Slave Data	Length	Communication

After double-clicking the added item, the user can edit the parameter.

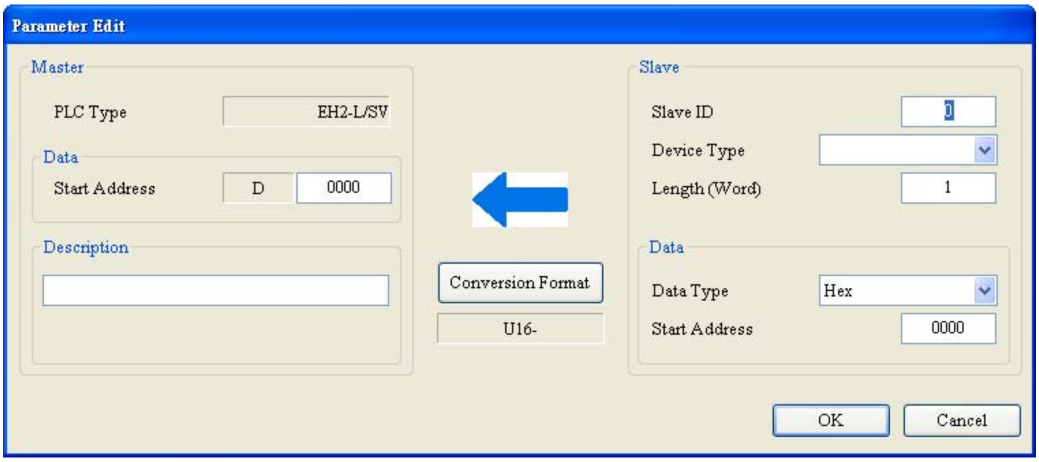


Read Bit

*	No.	□ Enable	Master Data	Slave ID	Slave Data	Length	State	Description

Read Word

*	No.	☑ Enable	Master Data	Conversion Format		Slave ID	Slave Data	Length	State	Des
	1	☑	D0000	U16-	←	0	H0000	1	D0516.0	



Master

PLC Type: EH2-L/SV

Data

Start Address: D 0000

Description

Conversion Format: U16-

Slave

Slave ID: 0

Device Type:

Length (Word): 1

Data

Data Type: Hex

Start Address: 0000

Master:

PLC Type: It displays the PLC type. The user can click “Tools” in SCMSoft to change the PLC type.

Data: Enter the address of the data register in the PLC to store the value read from the slave.

Description: Enter the description of the device. The maximum length is 30 bytes.

Salve:

Slave ID: The number of the salve device from which the data is read

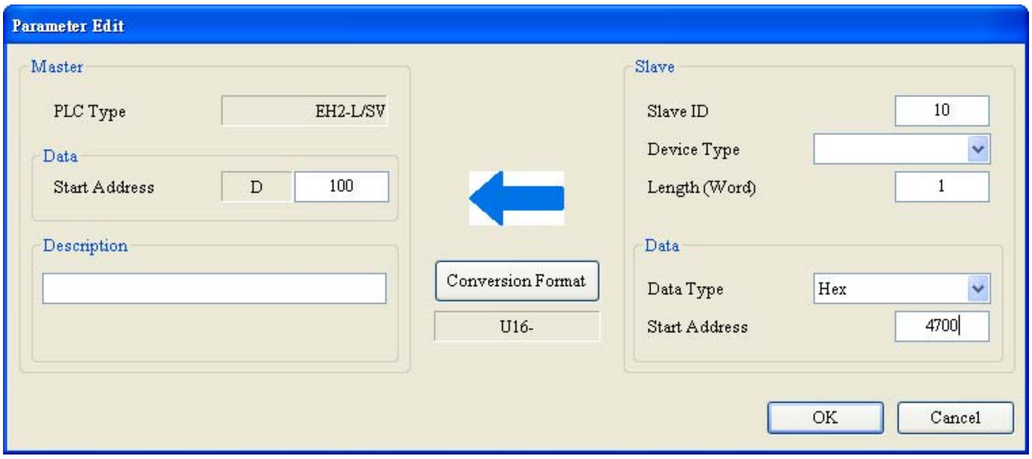
Device Type: The user can choose the Delta PLC type. If the PLC used is not a Delta PLC, please leave the column blank.

Length (bit): It indicates the length of the data being read. The maximum length is 100 bits.

Data Type: The user can choose either “Hex” or “Modbus 6 Digit”. “Hex” represents 6 hexadecimal digits, and “Modbus 6 Digit” represents 6 decimal digits. If the device type is a Delta PLC type, the data type in this column will automatically become the data register.

Start Address: The start address of the data

If the absolute position of the present value of the Delta DTA temperature controller is the hexadecimal value, 4700 (H'4700), and the station address is 10, the present value can be read and stored in D100 in the MPU of the PLC through COM1 on the SCM module. The settings are as follows:



Parameter Edit

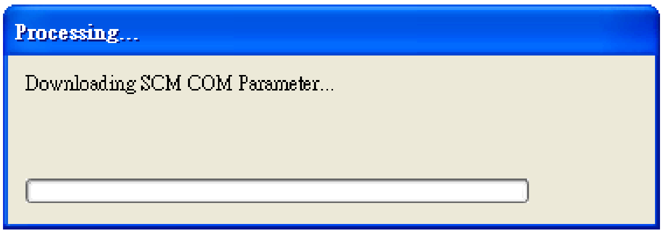
Master
PLC Type: EH2-L/SV
Data
Start Address: D 100
Description:

Conversion Format
U16-

Slave
Slave ID: 10
Device Type:
Length (Word): 1
Data
Data Type: Hex
Start Address: 4700

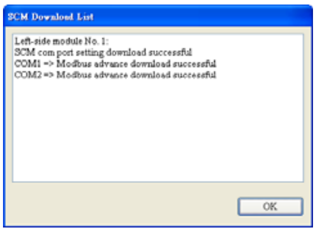
【Downloading】

After the setting is complete, check whether the other parameter settings conform to the slave setting. Then, click "Download".



Processing...

Downloading SCM COM Parameter...



SCM Download List

```
Left-side module No. 1:
SCM com port setting download successful
COM1 => Modbus advance download successful
COM2 => Modbus advance download successful
```

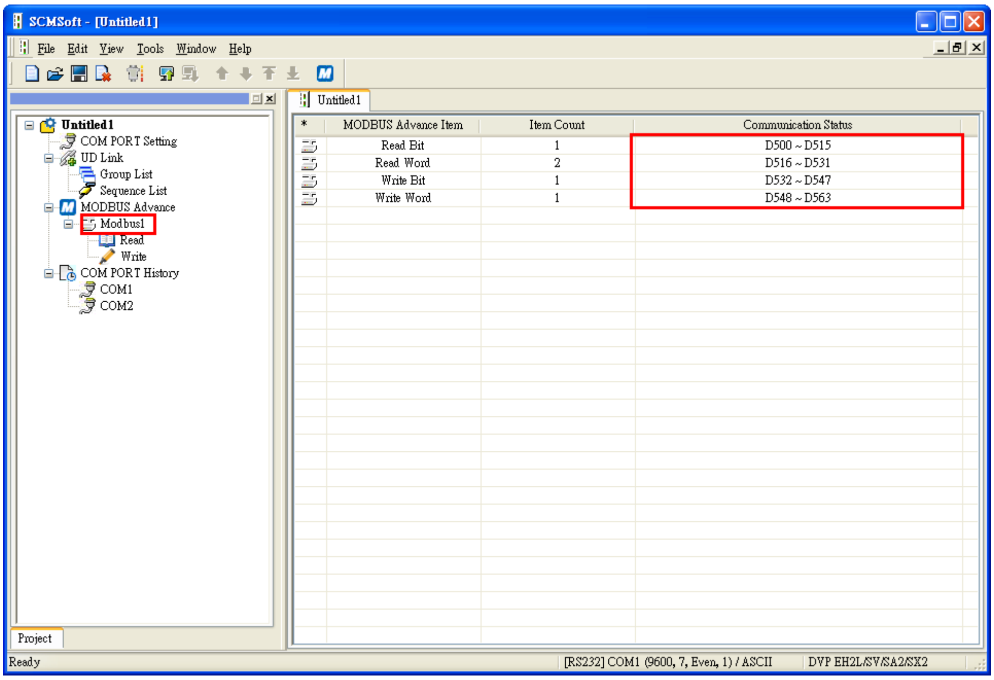
【Communication state】

The SCM module provides the communication state of MODBUS Advance. There are four sections – Read Bit, Read Word, Write Bit, and Write Word. The execution status in each line is stored in the bits in the data registers. If D100 is entered into No.1, the execution status of the data exchange in No.1 will be displayed in the first bit (b0) in D100, and by analogy, the execution status of the data exchange in No.2 will be displayed in the second bit (b1) if D100 is entered into N0.2.

Dn																
Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
No.	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1

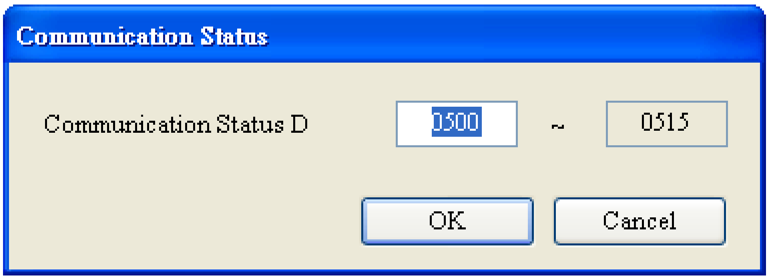
D (n+1)																
Bit	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
No.	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17

The default address is D500. The user can change the start address in MODBUS Advance.



MODBUS Advance Item	Item Count	Communication Status
Read Bit	1	D500 ~ D515
Read Word	2	D516 ~ D531
Write Bit	1	D532 ~ D547
Write Word	1	D548 ~ D563

Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2



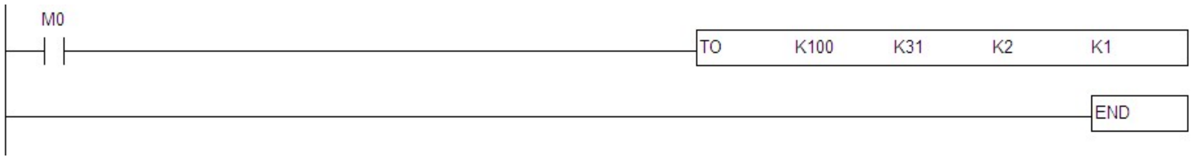
Communication Status D 0500 ~ 0515

【Enabling】

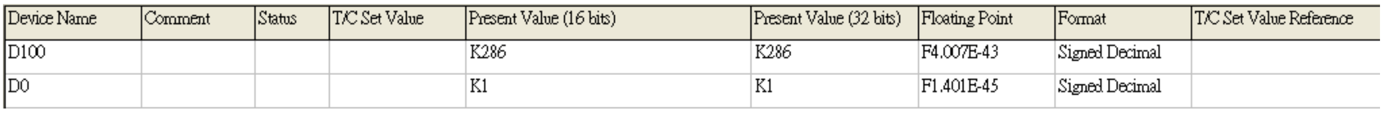
Control the data exchange through the instruction TO in WPLSoft to read bits/read words/write bits/write words (CR#31~CR#34).

CR#	Attribute	Name of the register	Description
31	R/W	Triggering the data exchange through COM1 to read bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered
32	R/W	Triggering the data exchange through COM2 to read bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered
33	R/W	Triggering the data exchange through COM1 to write bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered
34	R/W	Triggering the data exchange through COM2 to write bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered

If the user wants to keep executing the word-reading, the user can enter K2 into CR#31. If the user wants to execute the word-reading once, the user can enter K1 intro CR#31.



After M0 is triggered, COM1 on the SCM module will keep reading the present value which will be stored in D100, and the status value of bit0 in D0 is 1.



Device Name	Comment	Status	T/C Set Value	Present Value (16 bits)	Present Value (32 bits)	Floating Point	Format	T/C Set Value Reference
D100				K286	K286	F4.007E-43	Signed Decimal	
D0				K1	K1	F1.401E-45	Signed Decimal	

MEMO

6. Introduction of SCMSoft

This chapter will introduce the setting software of the SCM module – SCMSoft.

6.1 SCM Project

Through establishing an SCM project, the SCM module makes the execution plan for COM1 and COM2. An SCM project includes four parts – COM PORT Setting, UD Link, MODBUS Advance, and COM port history.

COM PORT Setting: The user can set the communication formats and the parameters that COM1and COM2 on the SCM module execute (Ch 6.2).

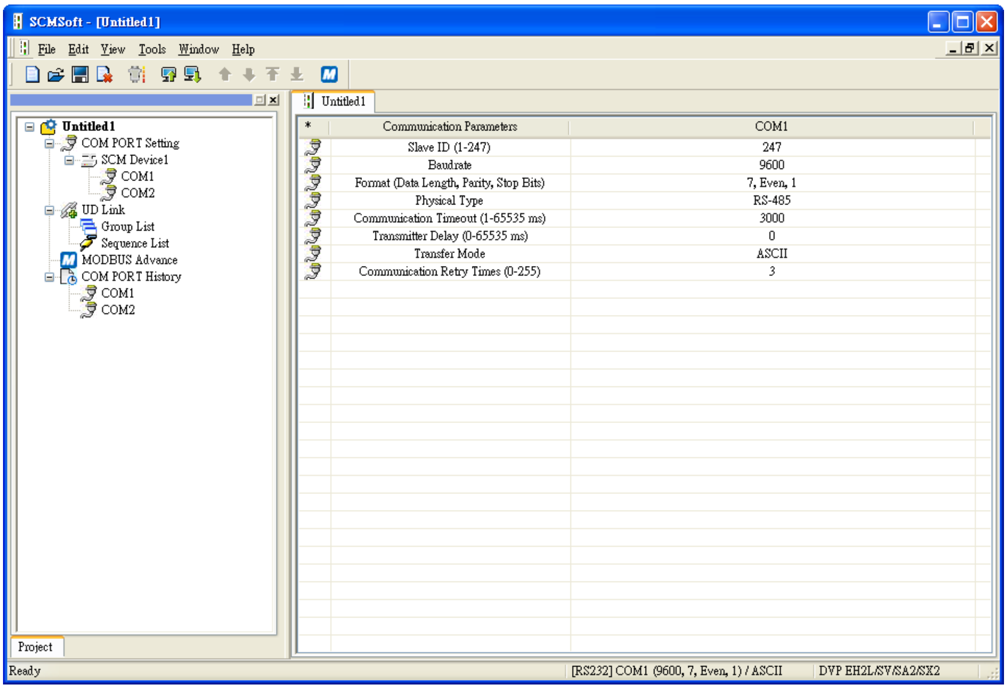
UD Link: The user can define the contentS of the RS-485/RS-422 packet (Ch 6.3).

MODBUS Advance: It can connect to the standard MODBUS RS-485/422 device. If other Delta automation products and other standard MODBUS communication devices are used, the user can use this function (Ch 6.4).

COM port history: The user can set whether to record the history of the communication port on the SCM module (Ch 6.5).

6.2 COM PORT Setting

Setting the serial communication format:



* Communication Parameters	COM1
Slave ID (1-247)	247
Baudrate	9600
Format (Data Length, Parity, Stop Bits)	7, Even, 1
Physical Type	RS-485
Communication Timeout (1-65535 ms)	3000
Transmitter Delay (0-65535 ms)	0
Transfer Mode	ASCII
Communication Retry Times (0-255)	3

[RS232] COM1 (9600, 7, Even, 1) / ASCII   DVP EH2L/SV/SA2/SX2

Protocol: If the standard MODBUS is used, the user can select MODBUS. If the user-defined RS-485/RS-422 format is used, the user can select UD Link.

Slave ID: The user can set the slave IDs of COM1 and COM2. The superior device connects to the SCM module through the slave ID. The default slave ID of COM1 is 247, and that of COM2 is 246.

Baudrate: It supports communication rates 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 230400, 460800 bps.

Physical Type: RS-485 or RS-422

Communication Timeout: If there has been no response for a certain period of time after the instruction is transmitted through the communication port, that period of time is called the communication timeout. The default communication timeout is 3000 ms.

Transmitter Delay: The default time interval between the instructions is 0 ms, that is, the next instruction is transmitted immediately after the reply is received.

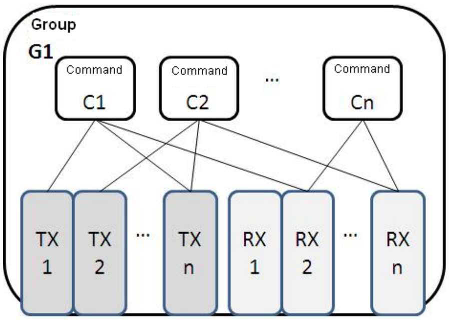
Transfer Mode: ASCII or RTU

Communication Retry Times: It means the number of times the communication has been retried after the communication fails. If there is still no response, the communication stops.

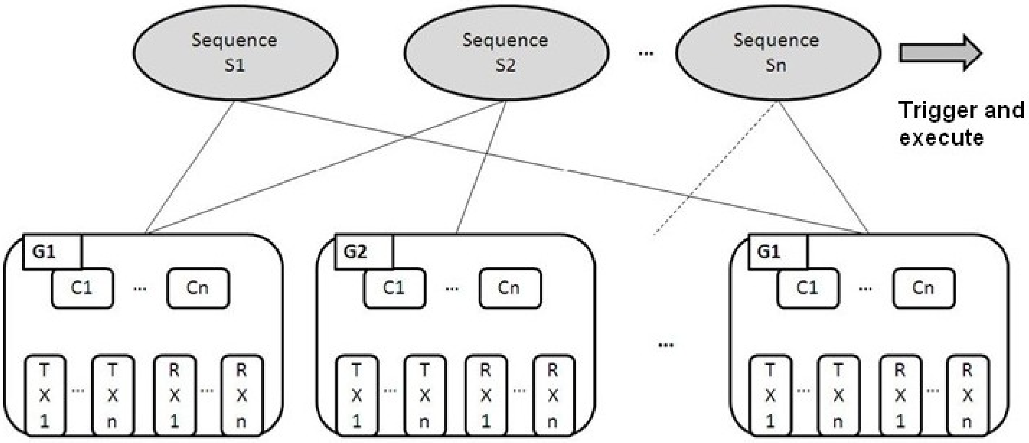
6.3 UD Link (User-defined Link)

UD Link provides non-Modbus RS-485/RS-422 link function. The packets can be edited according to the communication formats. The steps of establishing UD Link are as follows:

(1) Create a group→ Edit TX packets and RX packets→ Create commands→ Trigger and execute the instructions as a group



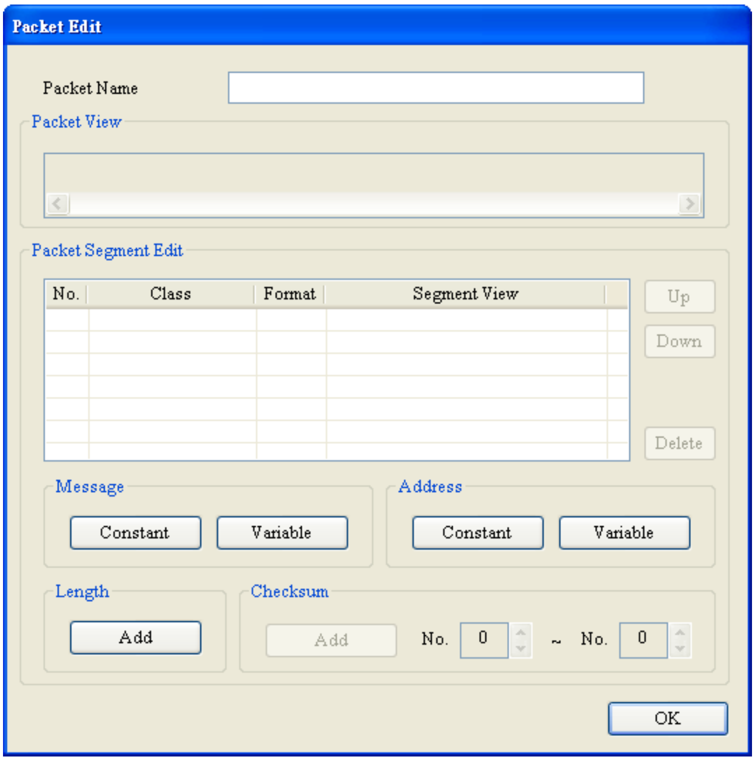
(2) Create a group → Edit TX packets and RX packets → Create commands→ Create other groups→ Create sequences→ Trigger and execute the instructions as sequences



First of all, establish the transmission instructions (TXs) and the reception instructions (RXs) in the group. Then, set the execution sequence and the number of times for TXs and RXs through the commands. Finally, trigger and execute the instructions as a group. In addition, if various groups of group packets are required in a large system, the user can create the groups in the sequences, and set the execution sequence.

6.3.1 TX Packets and RX Packets

The user can create various TX packets and RX packets in a group. The contents of TX packets and RX packets may include several messages, one address, one length, and one checksum.



- Packet Name: The user can edit the name of the packet.
- Packet View: It displays the contents of the packet.
- Packet Segment Edit: The user can adjust the sequence of the packet segment, and add/delete the packet segment.

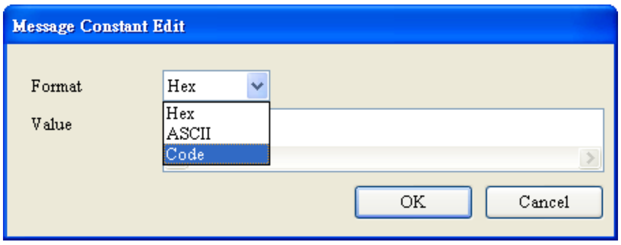
  No.: It is the packet segment number. The user can edit at most 64 segments in a packet.

  Class: The class of the segment includes the message, the address, the length and the checksum.

  Format: The format of the segment includes Hex, ASCII, Code, and etc.

  Segment View: The description of the segment
- Message: The user can edit the constant message and the variable message. Both the constant message and the variable message can be used with a packet head, a start bit, an end bit, or a data segment. One packet can include many messages.
- Address: The user can edit either the constant address or the variable address. One packet includes only one address segment.
- Length: The user can edit the length of the packet. One packet includes only one length segment.

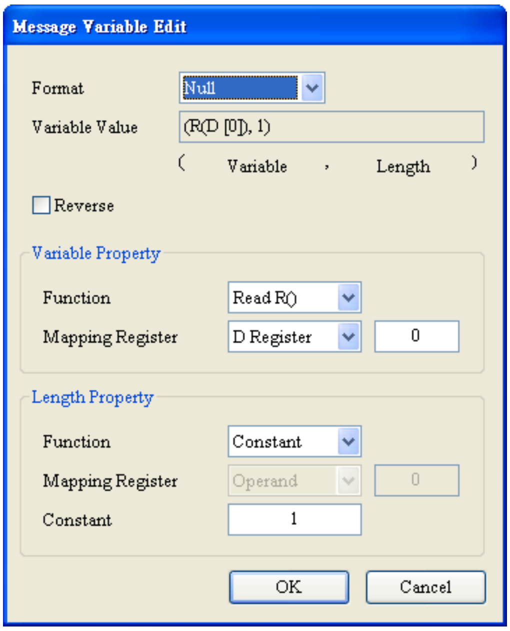
- Checksum: The user can edit the checksum. One packet includes only one checksum segment.



Format: Hex

Value:

- Constant: The data is a fixed value.
- Format: The format of the data can be Hex, ASCII, or Code. When the format of the data is Code, it indicates that the data uses the control code.
- Value: The user can enter the constant value.



Format: Null

Variable Value: (R(D [0]), 1)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Variable: The data is a variable whose mapping register can be the internal register in the SCM module or the register in the PLC.
- Format: The user can set the format of the data.

  Null: The user does not make any change to the format of the data.

  Hex: The ASCII data can be converted into the hexadecimal value. The words which can not be converted will become zeros.

  ASCII: The hexadecimal value can be converted into the ASCII data. The words which can not be converted will become zeros.
- Variable Property

  Function: The variable functions include "Read R ()", "Write W ()", and no action "*". The user can choose "Read R ()" for TX packets while the user can choose "Read R ()", "Write W ()", or no action "*" for RX packets.

  Mapping register: The user can choose the internal register in the SCM12 module or the register in the MPU of the PLC. The internal registers in the SCM module include I1, I2, O1, and

O2. The registers in the PLC include the data registers and “Base+Offset”.

Register	Definition	Register	Definition
D	Internal D register in the PLC	Base+Offset	It is used with the control register.
I1	It is used to receive/send the data through COM1.	O1	It is used to send the data through COM1.
I2	It is used to receive/send the data through COM2.	O2	It is used to send the data through COM2.

- Length

  Class: The length segment can be either 1 byte or 2 bytes.

  Format: The format of the length segment can be the hexadecimal value or the ASCII data.

  Value: The user can enter the length value according to the format setting.

- Checksum

  Class: The user can choose the class of the checksum segment.

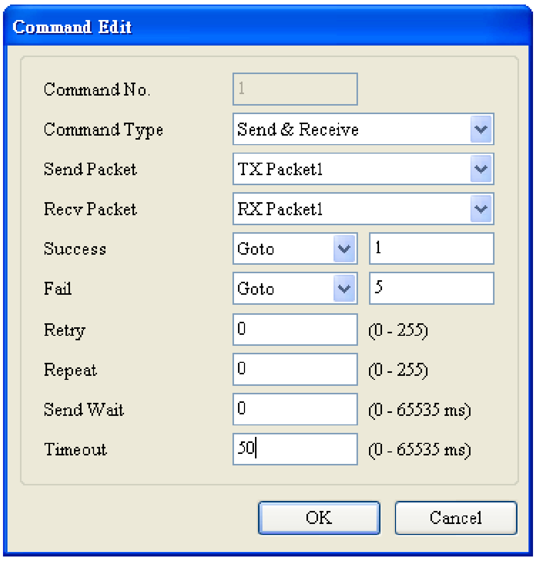
  Format: The user can choose the format of the checksum segment.

  Initial Value: The user can set the initial value of the checksum.

  Reverse: Reverse the checksum (word) in bytes.

6.3.2 Command

After creating many TX packets and RX packets, the user can choose the packets to be sent and received through creating the commands, and plan the sequence of executing the commands.



Command No. 1

Command Type: Send & Receive

Send Packet: TX Packet1

Recv Packet: RX Packet1

Success: Goto 1

Fail: Goto 5

Retry: 0 (0 - 255)

Repeat: 0 (0 - 255)

Send Wait: 0 (0 - 65535 ms)

Timeout: 50 (0 - 65535 ms)

Command No.: Each command has its own number. The user can designate the sequence of executing the commands through these numbers.

Command Type: The user can choose “Send”, “Receive”, or “Send & Receive”.

Send Packet: The user can choose the group name which has been created in the groups.

Receive Packet: The user can choose the group name which has been created in the groups.

Success: Designate the action following the execution of a command. The user can choose “Next”, ”Goto”, or “End”.

Next: Execute the next command. If the number of the command being executed is one, the number of the next command which will be executed is two.

Goto: The user can directly designate the command whose number is much larger.

End: The execution of commands comes to an end.

Fail: Designate the action following the execution of a command. The user can choose “Next”, ”Goto”, or “Abort”.

Next: Execute the next command. If the number of the command being executed is one, the number of the next command which will be executed is two.

Goto: The user can directly designate the command whose number is much larger.

Abort: The execution of commands comes to an end.

Retry: It means the number of times the sending of a command has been retried after the sending fails.

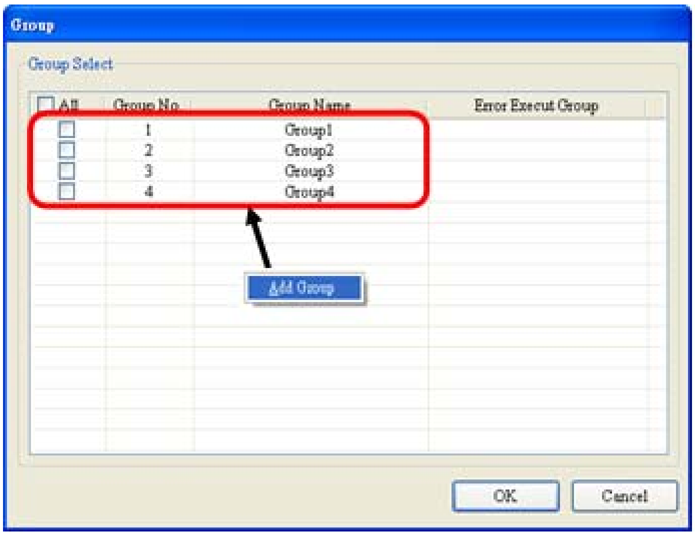
Repeat: It means the number of times the sending of a command has been repeated after the command has been executed successfully.

Send Wait: The default time interval between the instructions is 0 ms, that is, the next instruction is transmitted immediately after the reply is received.

Timeout: If there has been no response for a certain period of time after the instruction is transmitted through the communication port, that period of time is called the communication timeout. The default communication timeout is 50 ms.

6.3.3 Sequence

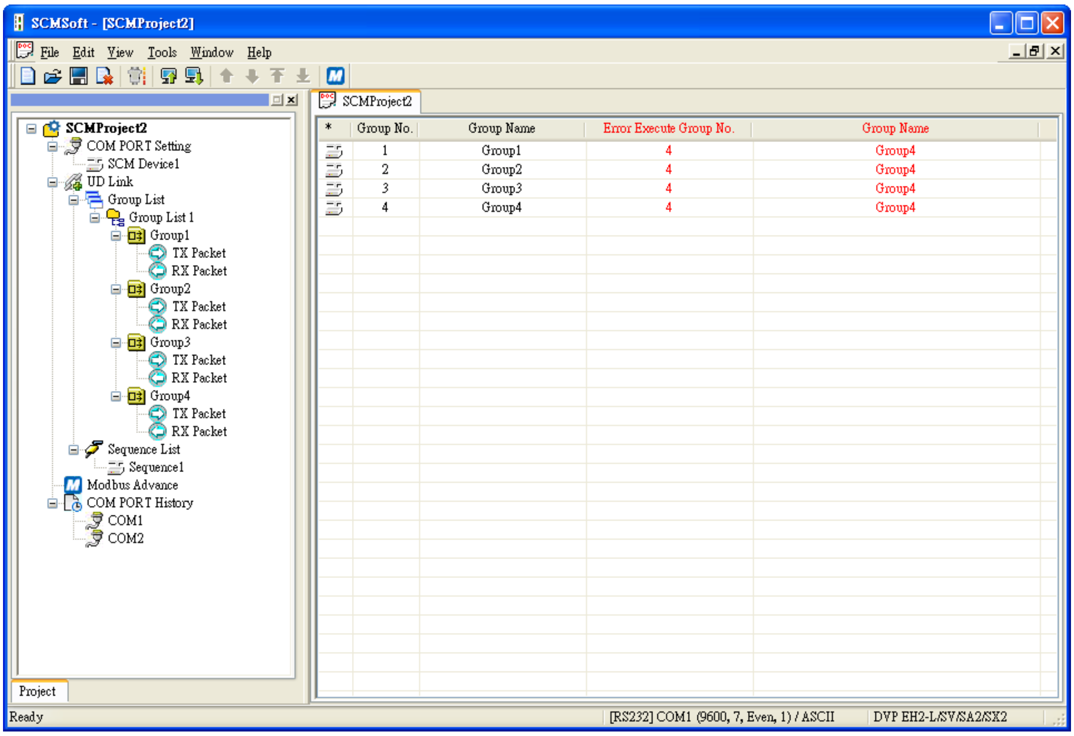
The user can click “Add Group” by pressing the right key of the mouse in Sequence to check the groups which will be executed. These groups will be downloaded as a sequence and executed through the serial port. In addition, the user can click “Error Execute Group No.” twice to set the group which will be executed when an error occurs. When there is an error in executing a group, the group which is set in “Error Execute Group No.” will be executed.



Group

Group Select

□ All	Group No	Group Name	Error Execut Group
□	1	Group1	
□	2	Group2	
□	3	Group3	
□	4	Group4	



*	Group No.	Group Name	Error Execute Group No.	Group Name
	1	Group1	4	Group4
	2	Group2	4	Group4
	3	Group3	4	Group4
	4	Group4	4	Group4

6.4 MODBUS Advance

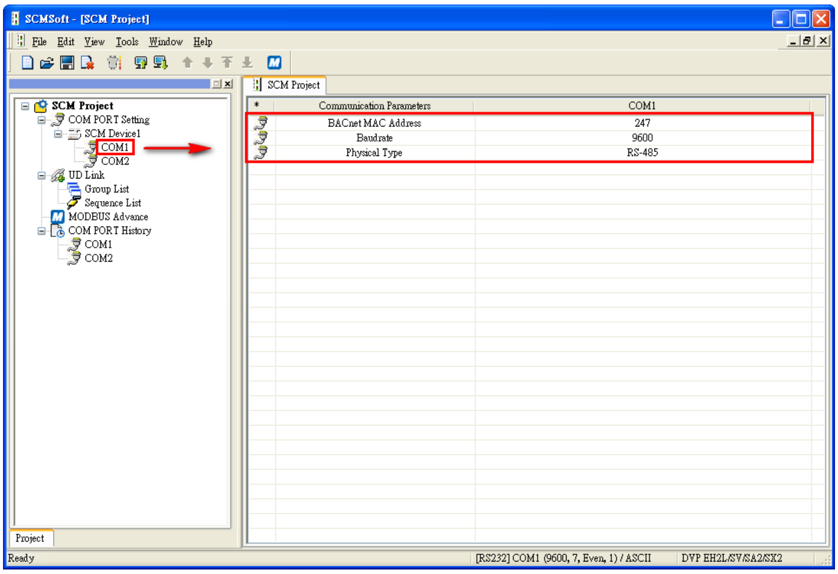
Please refer to chapter 5 for more related introduction.

6.5 BACnet MS/TP Slave Function (Supported by DVPSCM52-SL)

If the user wants to connect an SCM module to a BACnet MPU, the user has to set the BACnet parameters and the BACnet object for the SCM module.

6.5.1 BACnet Parameters

The BACnet parameters include the BACnet MAC address, the baud rate, and the physical type.



*	Communication Parameters	COM1
	BACnet MAC Address	247
	Baudrate	9600
	Physical Type	RS-485

BACnet MAC address: 1 ~ 247 (Default: 247)

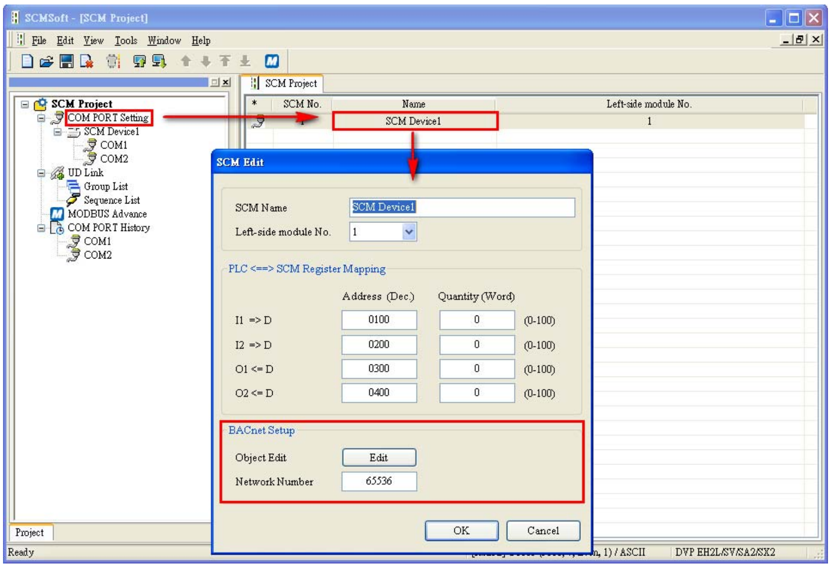
Please notice that the maximum MAC address that some masters support is 127.

Baud rates supported by BACnet: 9600 (default), 19200, 38400, and 76800 bps

Physical Type: The user can select RS-485 or RS-422.

6.5.2 BACnet Object

Network Number: The network number on the BACnet network is unique. It can not be used repeatedly. (Default: 65536)



*	SCM No.	Name	Left-side module No.
	1	SCM Device1	1

SCM Edit

SCM Name: SCM Device1

Left-side module No.: 1

PLC <==> SCM Register Mapping

	Address (Dec.)	Quantity (Word)	
I1 => D	0100	0	(0-100)
I2 => D	0200	0	(0-100)
O1 <= D	0300	0	(0-100)
O2 <= D	0400	0	(0-100)

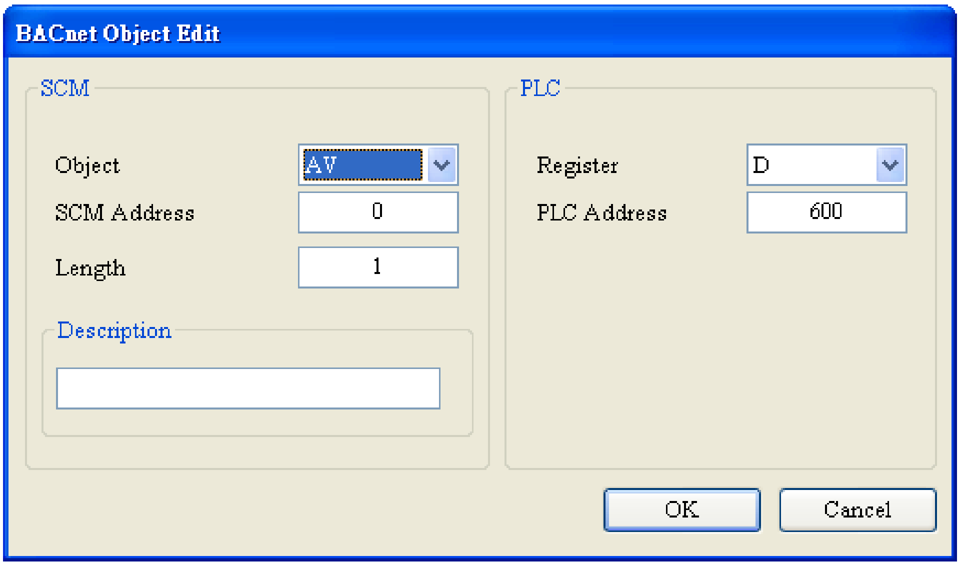
BACnet Setup

Object Edit: Edit

Network Number: 65536

BACnet object edit: Editing the AV and BV values which correspond to the data registers and coils in the Delta PLC master connecting to the SCM module

The lenghth of tha AV value corresponds to two data registers in the Delta PLC, and the lenghth of the BV value corresponds to one coil in the Delta PLC.


BACnet Object Edit

SCM

Object: AV

SCM Address: 0

Length: 1

Description:

PLC

Register: D

PLC Address: 600

Object: The user can select “AV” or “BV”. “AV” corresponds to the data registers in the PLC, and “BV” corresponds the coil in the PLC.

SCM address: The user can set the address of the AV, or the address of the BV. The setting range is 0~383.

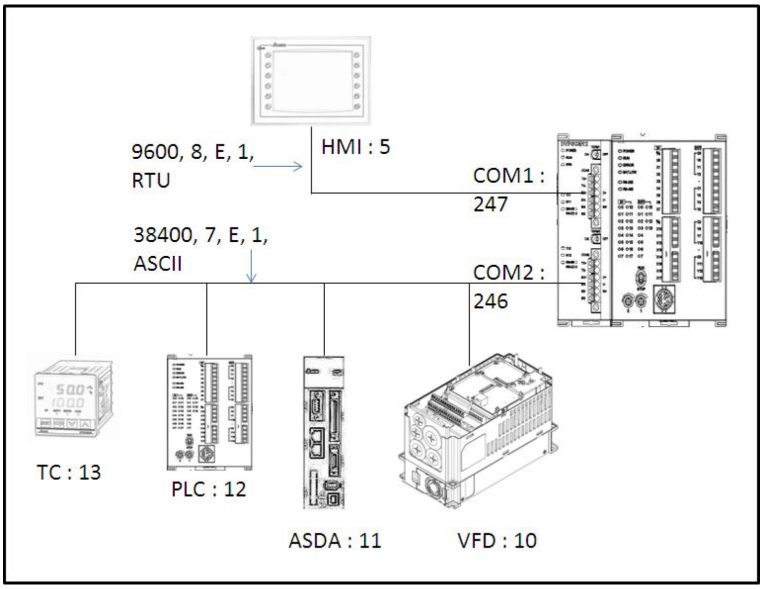
Length: A unit is a double word.

PLC: The start address in the Delta PLC.

7. Application

7.1 MODBUS

This chapter introduces how the SCM module connects to other Delta industrial products such as the human-machine interfaces, the text panels, the PLCs, the motor drives, and the servo motors through the standard MODBUS. The connection diagram is as below:

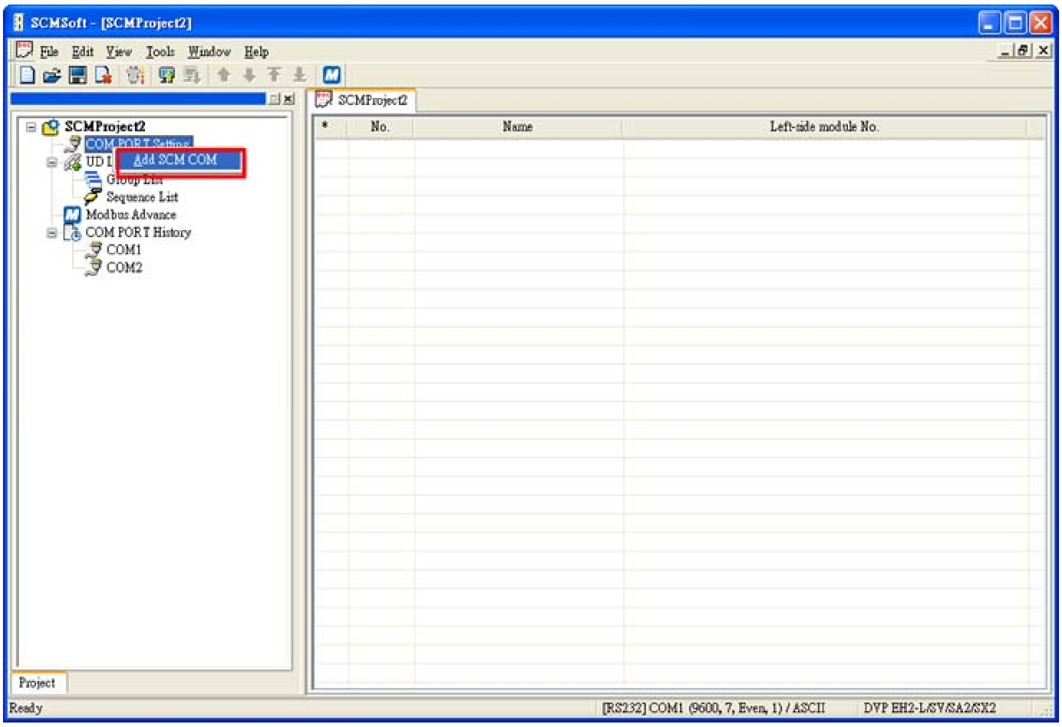


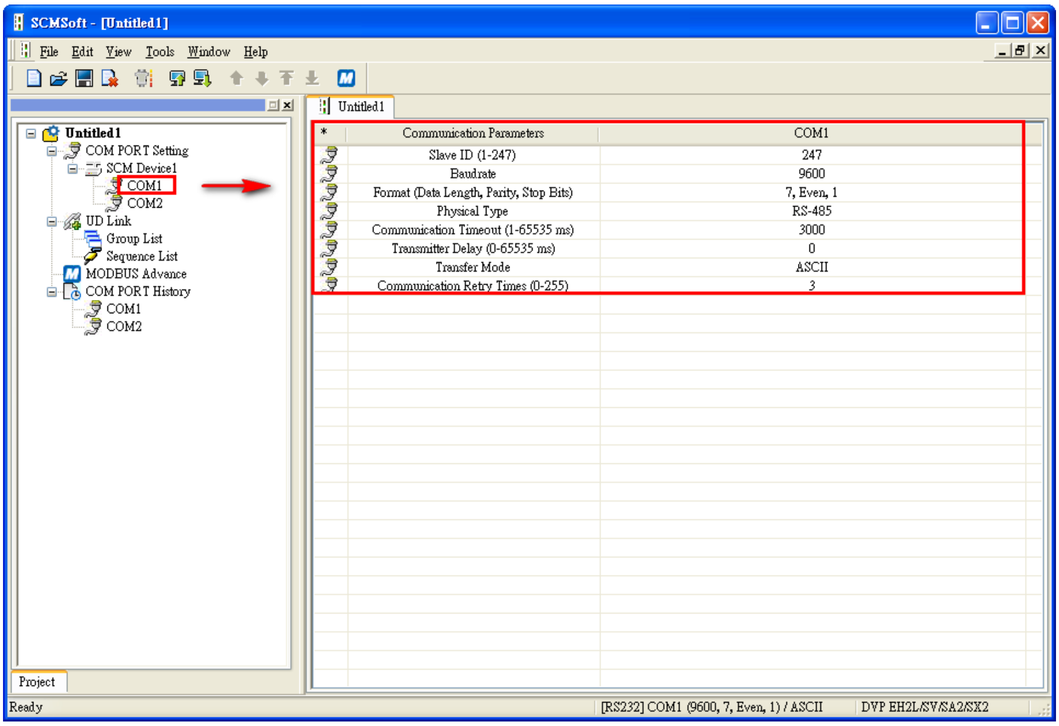
Product	Station address	Communication protocol	Address from which the data is read	Register in the MPU	Address into which the data is written	Register in the MPU
HMI	5	9600, RTU, 8, E, 1	-		-	
VFD	10	38400, ASCII, 7, E, 1	2103H	D100	2000H 2001H	D150~D151
ASDA	11	38400, ASCII, 7, E, 1	0101H 020AH	D200,D201	0101H 020AH	D250, D251
PLC	12	38400, ASCII, 7, E, 1	D100~D109	D300~D309	D200~D204	D350~D354
TC	13	38400, ASCII, 7, E, 1	1000H (PV)	D400	1001H (SV)	D451

7.1.1 Connection between the MODBUS Slave and the Delta Product

(1) For SCM as the MODBUS slave, the user only has to set the parameters such as the station address and the baudrate to allow the connection with the master.

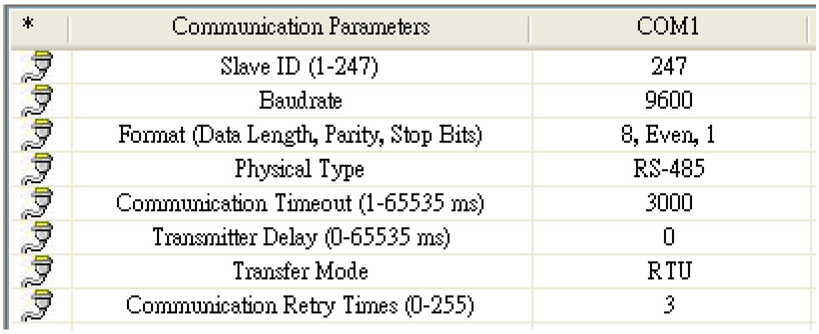
Open SCMSoft → 『New Project』→ COM PORT setting: 『Add SCM COM』→ Set the communication parameters.





Communication Parameters	COM1
Slave ID (1-247)	247
Baudrate	9600
Format (Data Length, Parity, Stop Bits)	7, Even, 1
Physical Type	RS-485
Communication Timeout (1-65535 ms)	3000
Transmitter Delay (0-65535 ms)	0
Transfer Mode	ASCII
Communication Retry Times (0-255)	3

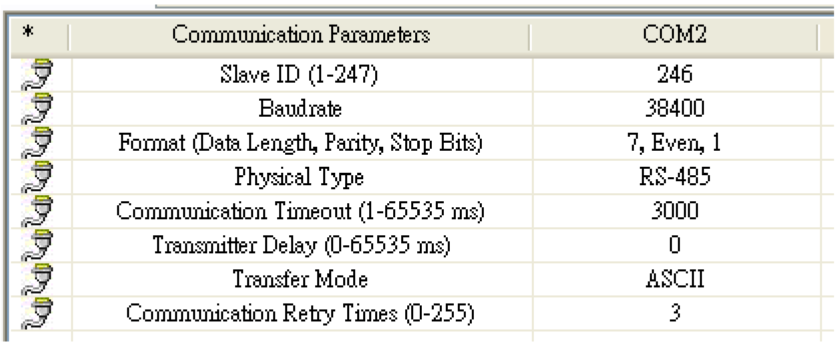
Set the communication parameters of COM1: station address 247 (default), Modbus RTU, 9600, 8, Even, 1.



*	Communication Parameters	COM1
	Slave ID (1-247)	247
	Baudrate	9600
	Format (Data Length, Parity, Stop Bits)	8, Even, 1
	Physical Type	RS-485
	Communication Timeout (1-65535 ms)	3000
	Transmitter Delay (0-65535 ms)	0
	Transfer Mode	RTU
	Communication Retry Times (0-255)	3

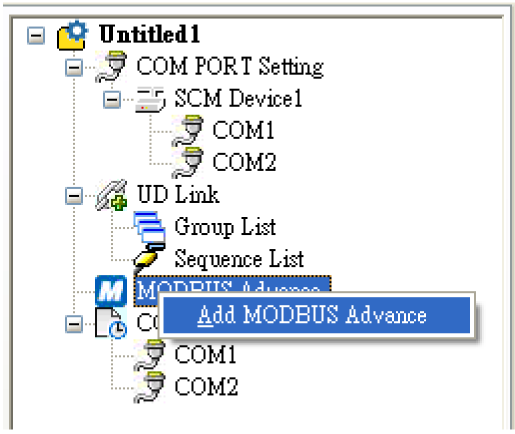
7.1.2 Connection between the MODBUS Master and the Delta Product

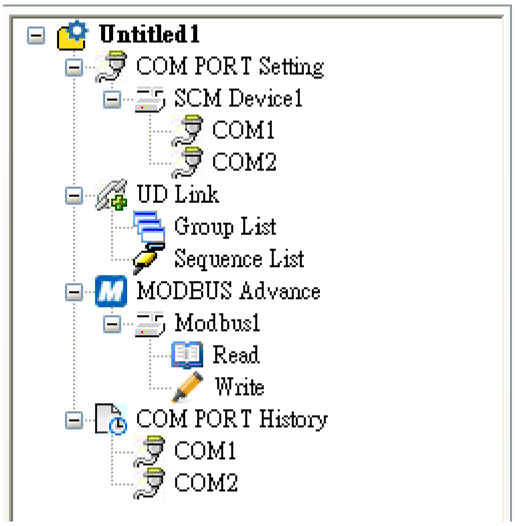
(1) Set the communication parameters of COM2: station address 246 (default), Modbus ASCII, 38400, 7, Even, 1.



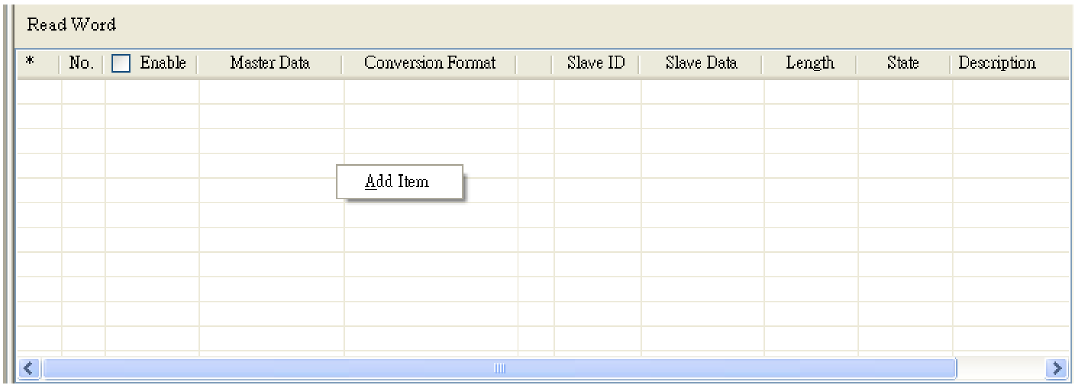
*	Communication Parameters	COM2
	Slave ID (1-247)	246
	Baudrate	38400
	Format (Data Length, Parity, Stop Bits)	7, Even, 1
	Physical Type	RS-485
	Communication Timeout (1-65535 ms)	3000
	Transmitter Delay (0-65535 ms)	0
	Transfer Mode	ASCII
	Communication Retry Times (0-255)	3

(2) Add MODBUS Advance.



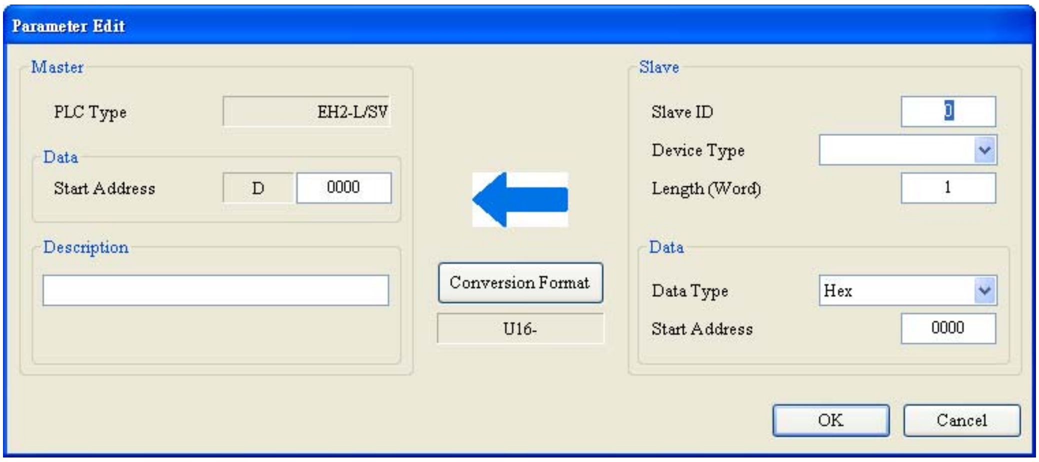


(3) Set the data exchange in the slave: Add Item → Click the added item twice to set the reading/writing information in the slave.



Read Word

*	No.	□ Enable	Master Data	Conversion Format	Slave ID	Slave Data	Length	State	Description

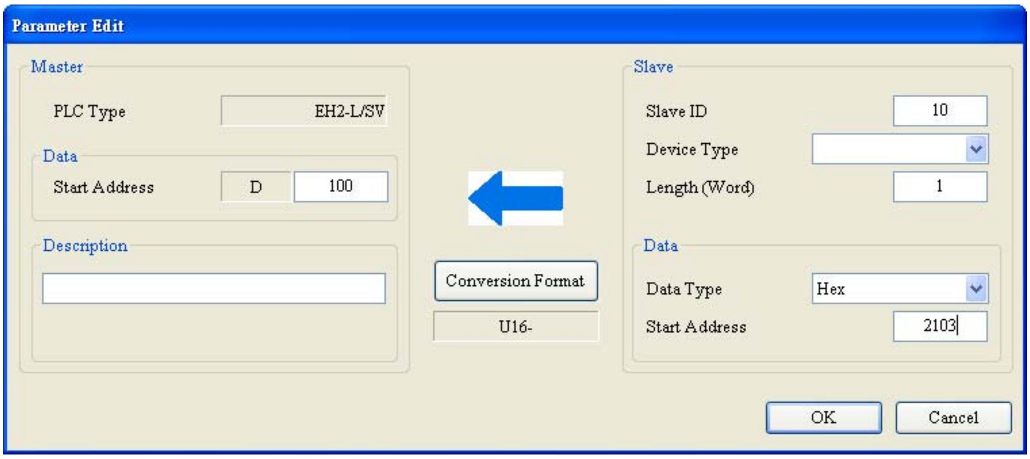


Parameter Edit

Master
PLC Type: EH2-L/SV
Data
Start Address: D 0000
Description

Slave
Slave ID: 0
Device Type:
Length (Word): 1
Data
Data Type: Hex
Start Address: 0000

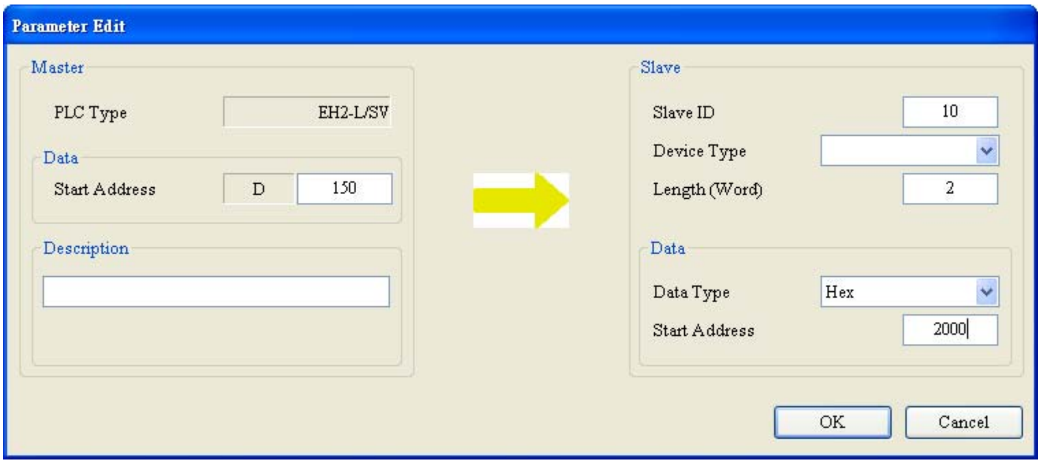
VFD (D100←2103H), (D150, D151→H2000, H2001)



Parameter Edit

Master
PLC Type: EH2-L/SV
Data
Start Address: D 100
Description

Slave
Slave ID: 10
Device Type:
Length (Word): 1
Data
Data Type: Hex
Start Address: 2103



Parameter Edit

Master

PLC Type: EH2-L/SV

Data

Start Address: D 150

Description

Slave

Slave ID: 10

Device Type:

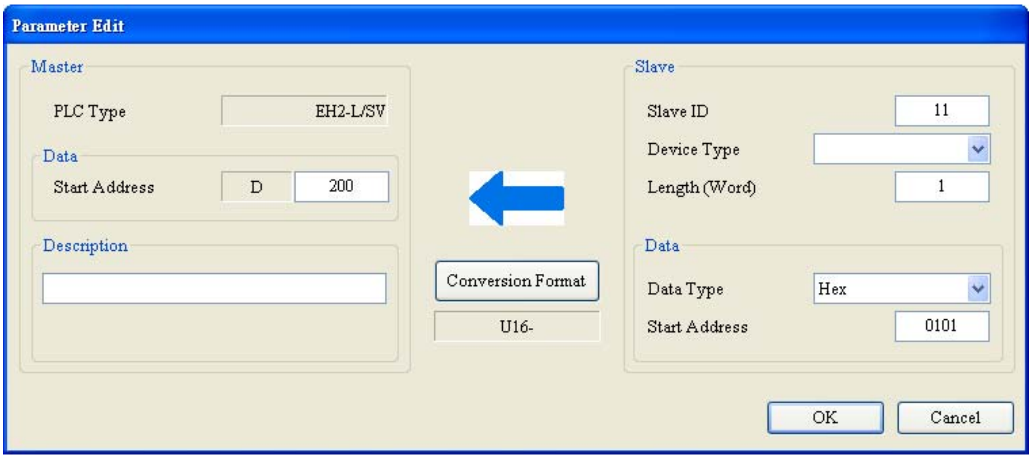
Length (Word): 2

Data

Data Type: Hex

Start Address: 2000

ASDA (D200←0101H, D201←020AH)



Parameter Edit

Master

PLC Type: EH2-L/SV

Data

Start Address: D 200

Description

Conversion Format: U16-

Slave

Slave ID: 11

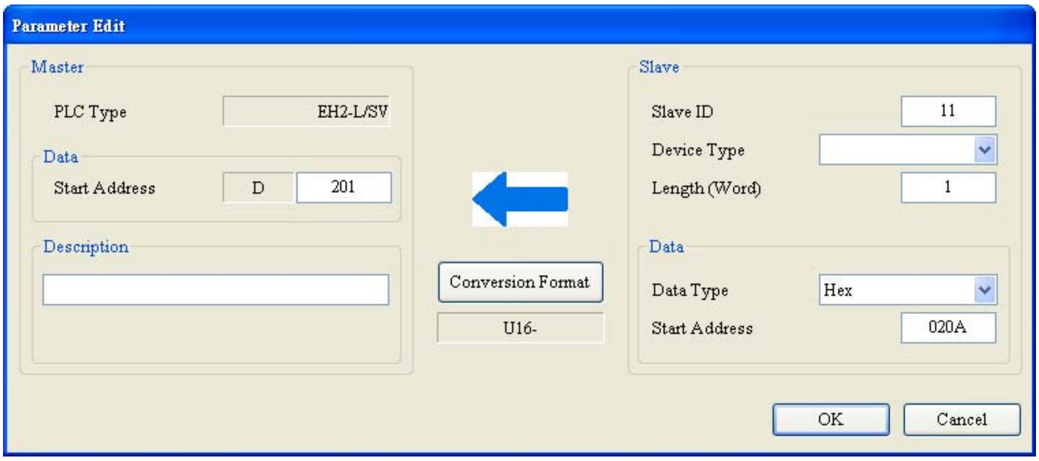
Device Type:

Length (Word): 1

Data

Data Type: Hex

Start Address: 0101



Parameter Edit

Master

PLC Type: EH2-L/SV

Data

Start Address: D 201

Description

Conversion Format: U16-

Slave

Slave ID: 11

Device Type:

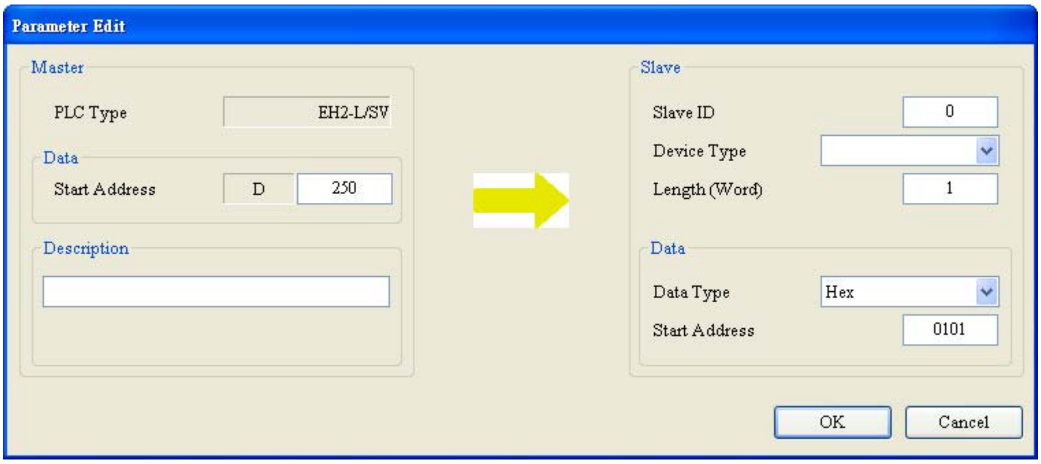
Length (Word): 1

Data

Data Type: Hex

Start Address: 020A

(D250→0101H, D251→020AH)



Parameter Edit

Master

PLC Type: EH2-L/SV

Data

Start Address: D 250

Description

Slave

Slave ID: 0

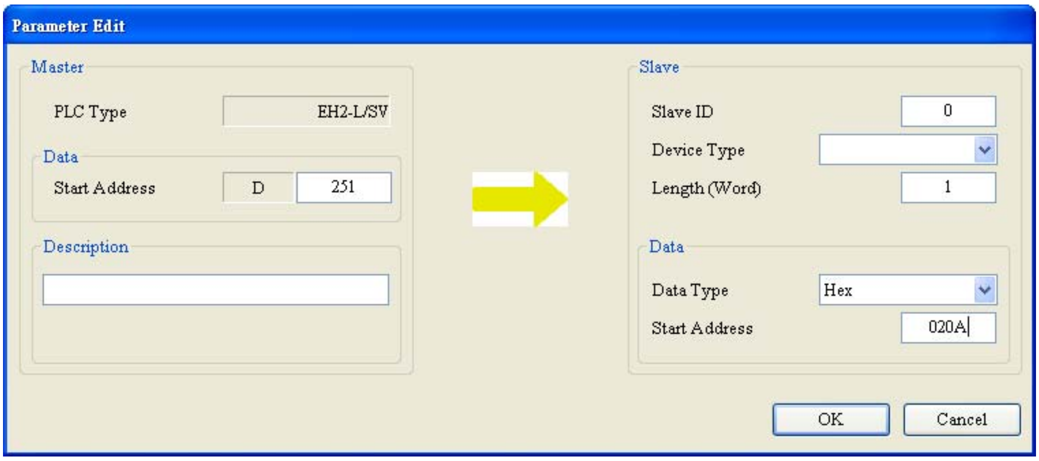
Device Type:

Length (Word): 1

Data

Data Type: Hex

Start Address: 0101



Parameter Edit

Master

PLC Type: EH2-L/SV

Data

Start Address: D 251

Description

Slave

Slave ID: 0

Device Type:

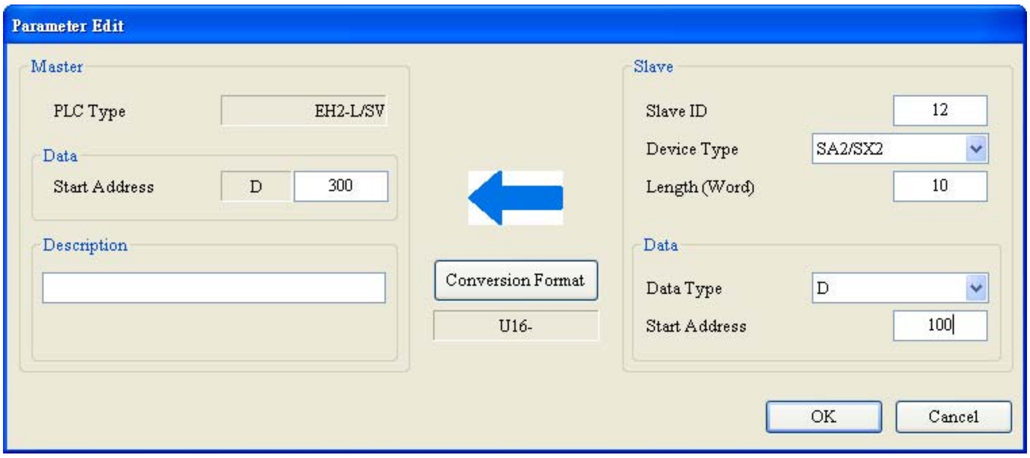
Length (Word): 1

Data

Data Type: Hex

Start Address: 020A

PLC (D300~D309 in the master←D100~D109 in the slave), (D350~D354 in the master→D150~D154 in the slave)



Parameter Edit

Master

PLC Type: EH2-L/SV

Data

Start Address: D 300

Description

Conversion Format: U16-

Slave

Slave ID: 12

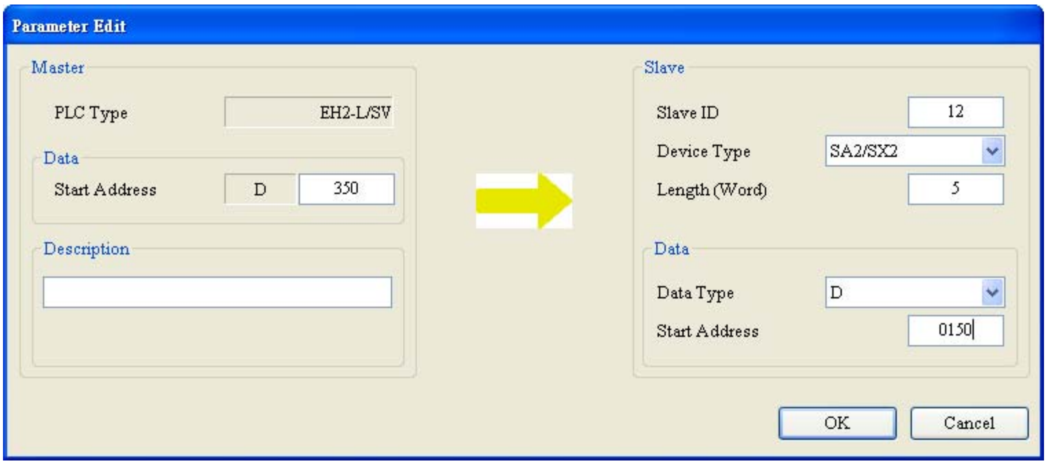
Device Type: SA2/SX2

Length (Word): 10

Data

Data Type: D

Start Address: 100

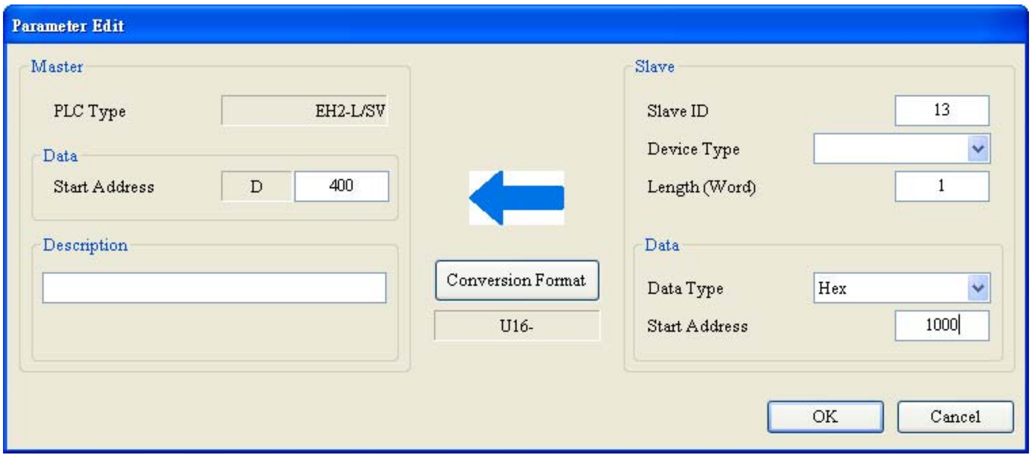


Parameter Edit

Master
PLC Type: EH2-L/SV
Data
Start Address: D 350
Description:

Slave
Slave ID: 12
Device Type: SA2/SX2
Length (Word): 5
Data
Data Type: D
Start Address: 0150

TC (D400←1000H), (D451→1001H)

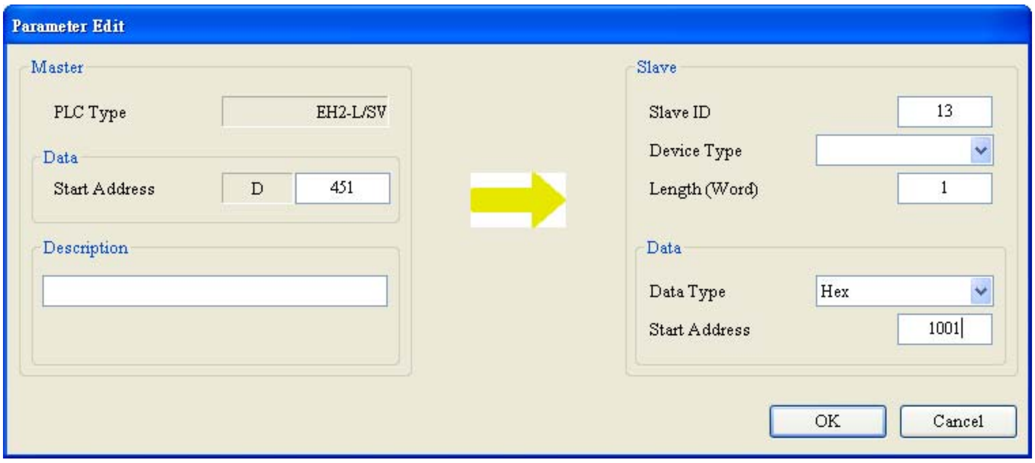


Parameter Edit

Master
PLC Type: EH2-L/SV
Data
Start Address: D 400
Description:

Conversion Format: U16-

Slave
Slave ID: 13
Device Type:
Length (Word): 1
Data
Data Type: Hex
Start Address: 1000

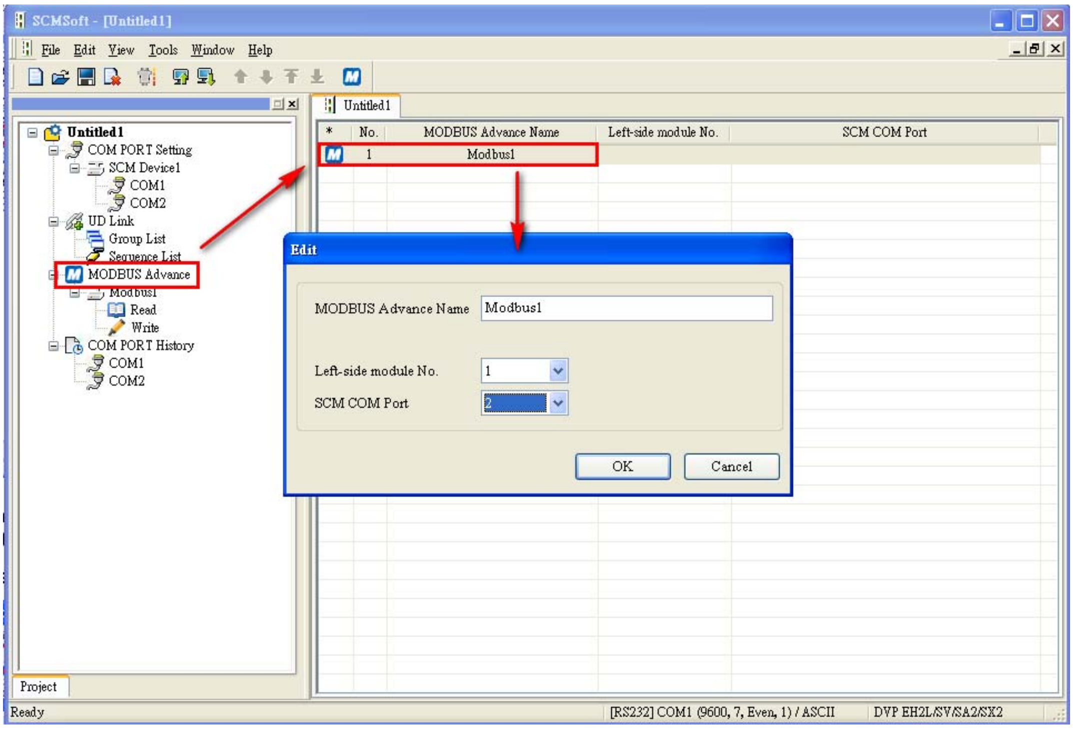


Parameter Edit

Master
PLC Type: EH2-L/SV
Data
Start Address: D 451
Description:

Slave
Slave ID: 13
Device Type:
Length (Word): 1
Data
Data Type: Hex
Start Address: 1001

After the setting is complete, the user can designate the communication port using MODBUS Advance – COM port 2 on the first left-side module.



*	No.	MODBUS Advance Name	Left-side module No.	SCM COM Port
	1	Modbus1		

MODBUS Advance Name: Modbus1

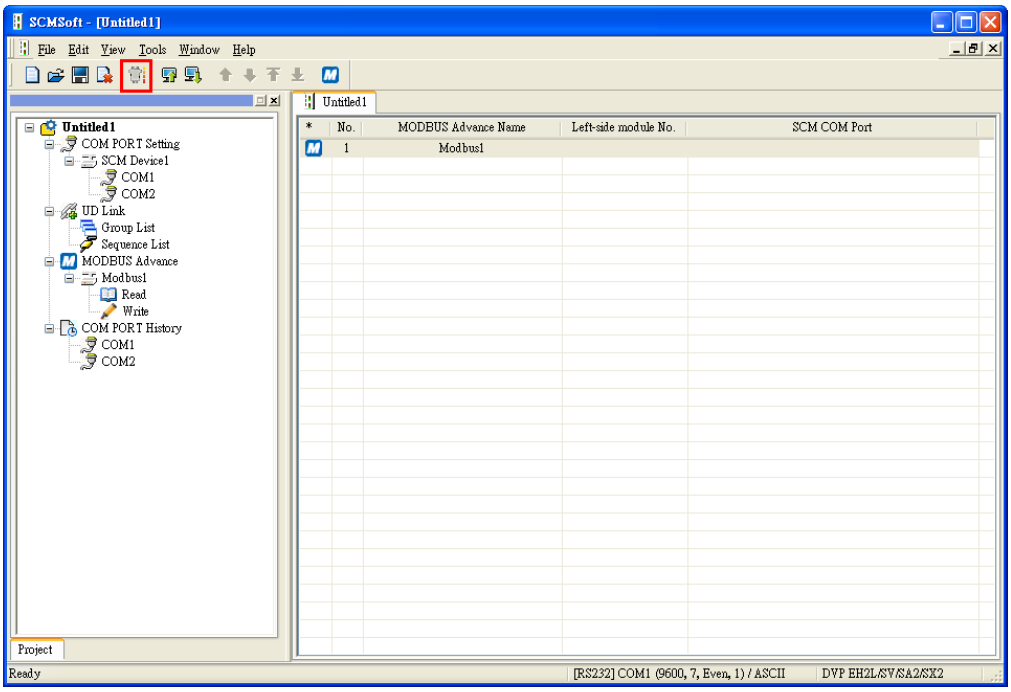
Left-side module No.: 1

SCM COM Port: 2

Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2

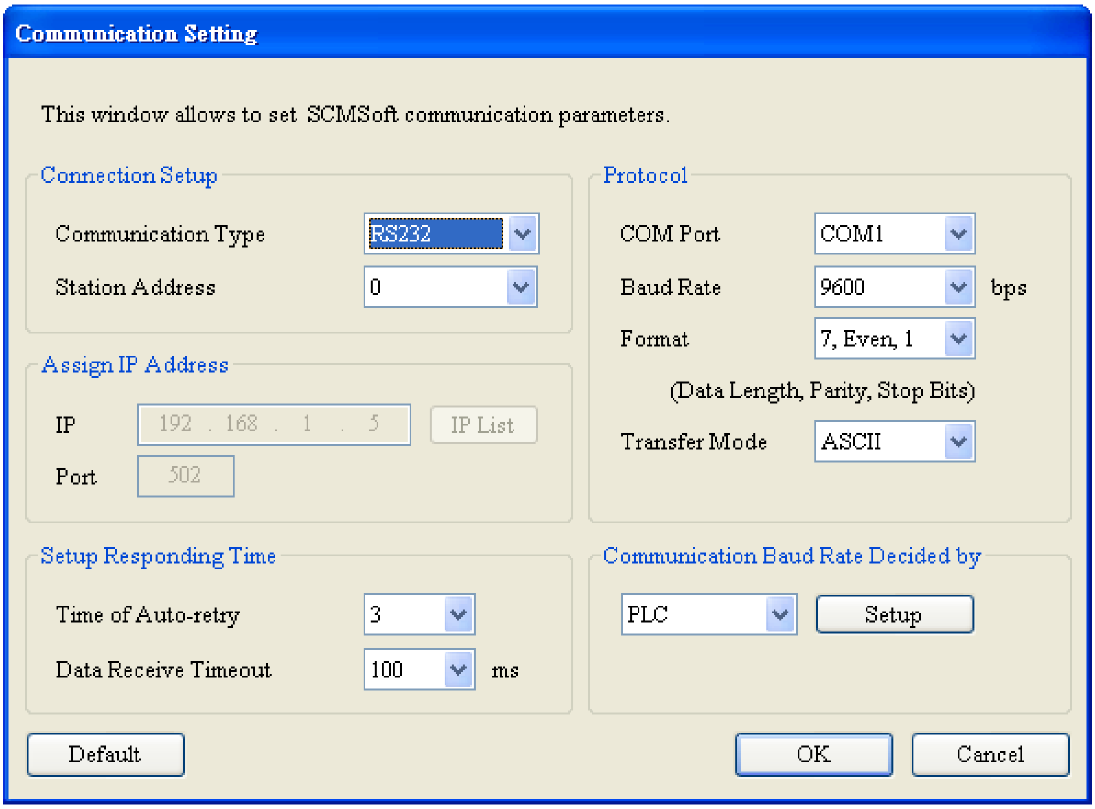
(4) Download the parameters.

The user can set the communication. After the setting is complete, click “OK” to exit from the communication setting, and the parameters are set.



*	No.	MODBUS Advance Name	Left-side module No.	SCM COM Port
	1	Modbus1		

Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2



This window allows to set SCMSoft communication parameters.

Communication Type: RS232

Station Address: 0

IP: 192 . 168 . 1 . 5

Port: 502

Time of Auto-retry: 3

Data Receive Timeout: 100 ms

COM Port: COM1

Baud Rate: 9600 bps

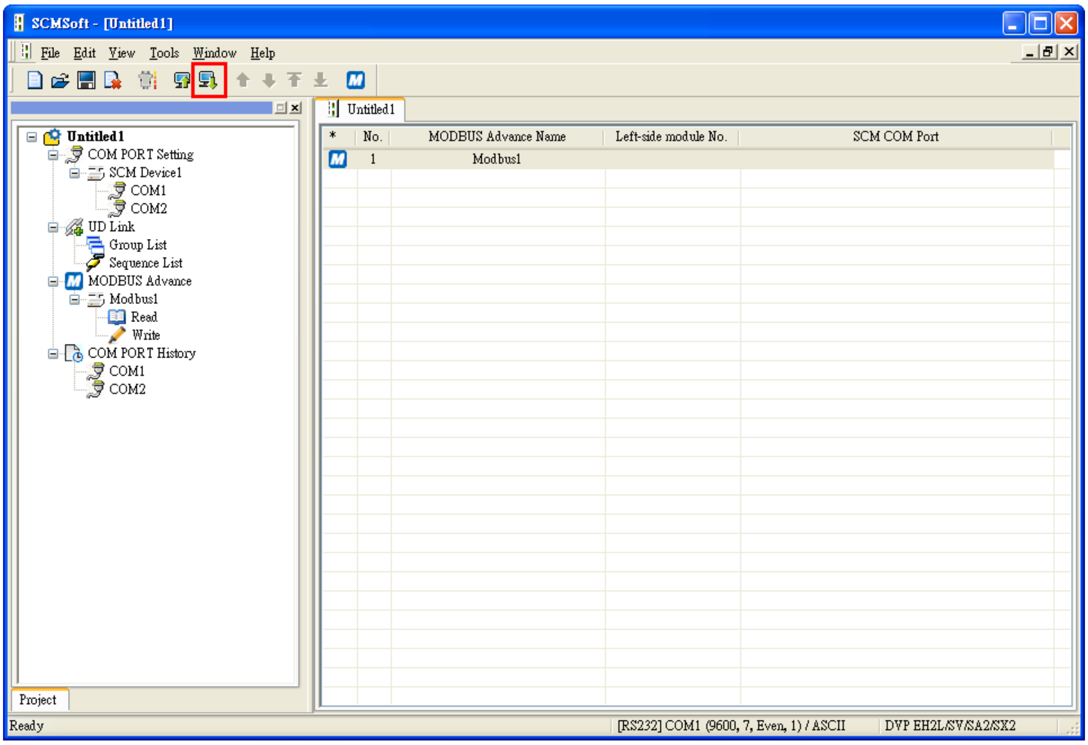
Format: 7, Even, 1

(Data Length, Parity, Stop Bits)

Transfer Mode: ASCII

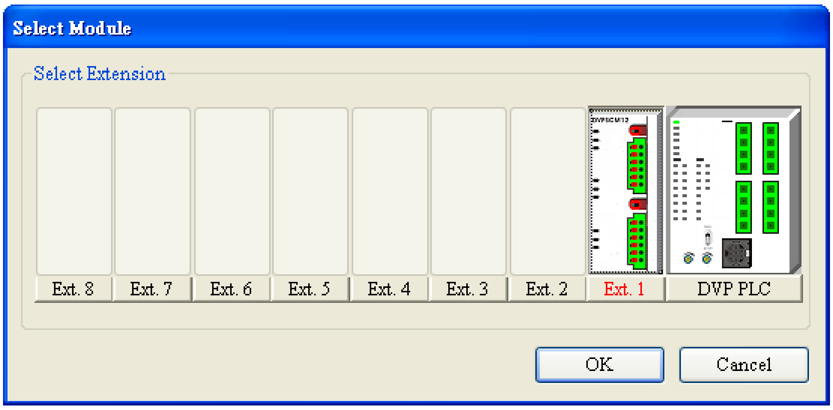
Communication Baud Rate Decided by: PLC

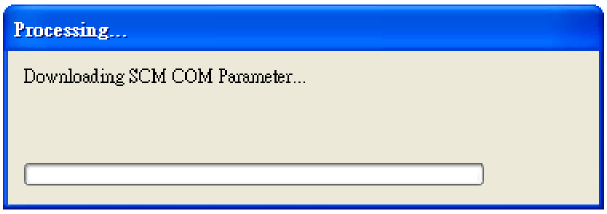
Click “Download”, choose the left-side module which will be downloaded, and click “OK”. If only one device is connected, click “OK” directly.



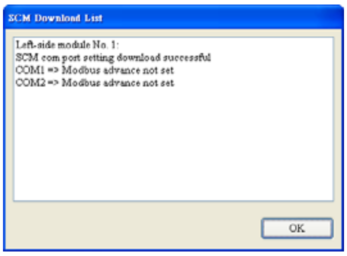
*	No.	MODBUS Advance Name	Left-side module No.	SCM COM Port
M	1	Modbus1		

Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2





Downloading SCM COM Parameter...



Left-side module No. 1:
SCM com port setting download successful
COM1 => Modbus advance not set
COM2 => Modbus advance not set

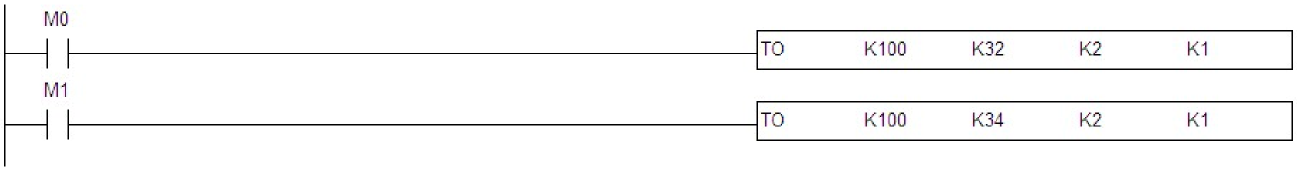
(5) Enable the data echange function.

Control the data exchange through the instruction TO in WPLSoft to read bits/read words/write bits/write words (CR#31~CR#34).

31	R/W	Triggering the data exchange through COM1 to read bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered
32	R/W	Triggering the data exchange through COM2 to read bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered
33	R/W	Triggering the data exchange through COM1 to write bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered
34	R/W	Triggering the data exchange through COM2 to write bits or words.	High byte: bit; Low byte: word 0: Not triggered, 1: Triggered once, 2: Always triggered

If the user wants to keep executing the word-reading, the user can enter K2 into CR#32. If the user wants to execute the word-reading once, the user can enter K1 intro CR#32.

If the user wants to keep executing the word-writing, the user can enter K2 into CR#34. If the user wants to execute the word-writing once, the user can enter K1 into CR#34.



After M0 is triggered, the data will be read from the salve address which has been set through COM2 on the SCM module.

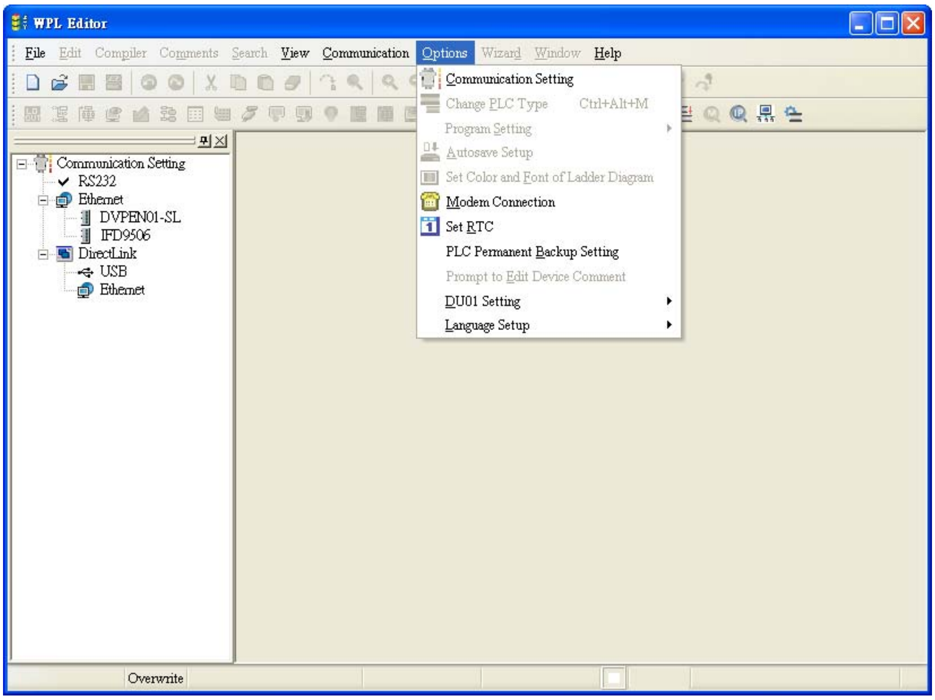
After M1 is triggered, the data will be written into the slave address which has been set through COM2 on the SCM module.

7.2 Connecting to WPLSoft

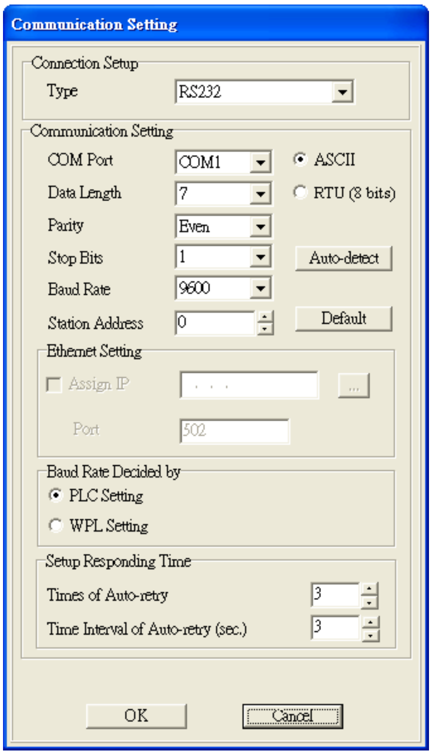
The SCM module can be used as the additional communication port of the PLC master. When RS-485 communication of the PLC master is executed, the user can use WPLSoft to monitor the master through the SCM module. The default communication format of COM1 on the SCM module is 9600, 7, Even, 1, and the station address is 247.

(1) Set WPLSoft.

Open WPLSoft. Click “Options” and choose “Communication Setting”.



(2) Choose RS-232 in Communication Setting, designate “COMP Port”, and enter the communication parameters. The communication parameters here should conform to the default setting of COM1 on the SCM module. If other communication parameters are used, they need to be modified in COM PORT Setting of the SCM module. In addition, the setting of “Station Address” should conform to COM1 on the SCM module rather than the station address of the MPU of the PLC.



Communication Setting

Connection Setup

Type: RS232

Communication Setting

COM Port: COM1 ● ASCII

Data Length: 7 ○ RTU (8 bits)

Parity: Even

Stop Bits: 1

Baud Rate: 9600

Station Address: 0

Ethernet Setting

□ Assign IP: . . .

Port: 502

Baud Rate Decided by

● PLC Setting

○ WPL Setting

Setup Responding Time

Times of Auto-retry: 3

Time Interval of Auto-retry (sec.): 3

(3) Click "OK" to upload/download WPLSoft program from/to the MPU of the PLC.

7.3 RS-485

This section introduces how SCM connects to other Delta industrial products through RS-485 (the non-standard MODBUS).

7.3.1 Connecting to the Electricity Meter

There are two common modes of connecting to the electricity meter. One is through the standard MODBUS, the other is through RS-485. This section introduces how the SCM module connects to the electricity meter through RS-485 in UD Link.

(1) The record type

Set the station address of the electricity meter to 5. The electricity meter includes three record types — abbreviated, control and full record types.

(Abbreviated)

Word number	Content	Description
1	10h	Start bit
2	0 … FAh, FFh	Device address (IA)
3		Function code (FF)
4		Checksum (CS) (IA+FF)
5	16h	End marker

(Full)

Word number	Content	Description
1	68h	Start bit
2		Length
3		Length (repeat)
4	68h	Start bit (repeat)
5	0 … FAh, FFh	Device address (IA)
6		Function code (FF)
7		Parameter index (PI)
…		n word, data block
Length+5		Checksum (CS) (Add from IA to the previous item.)
Length+6	16h	End marker

(Control)

Word number	Content	Description
1	68h	Start bit
2	03h	Length
3	03h	Length (repeat)
4	68h	Start bit (repeat)
5	0 … FAh, FFh	Device address (IA)
6		Function code (FF)
7		Parameter index (PI)
8		Checksum (CS) (Add from IA to PI.)
9	16h	End marker

(2) The usage

There are nine types of usage in which the SCM module communicates with the electricity meter through the combination of three record types.

Type	Instruction to the electricity meter	Response (through the record type)
1	Reset Abbreviated record	N/A
2	Query about the status of the device: abbreviated record	Abbreviated record
3	Measured value and error (cyclic data) Abbreviated record	Full record
4	Event data analyzed erroneously Abbreviated record	Full record
5	Measured value Control record	Full record
6	Output parameter: control record	Full record
7	Status: control record	Full record
8	Device specifications: control record	Full record
9	Real-time timing data:	Full record

(3) Edit the UD Link.

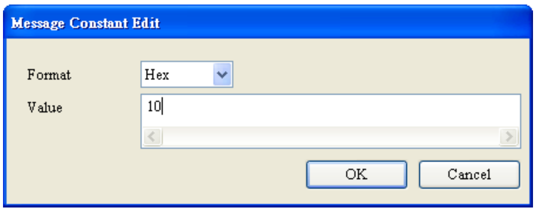
【Type 1】

Only send the abbreviated record (abbreviated record):

『Start word』 + 『device address (IA)』 + 『Function code (FF)』 + 『Checksum (CS)』 + 『End marker』

→ 10h + D0 + 09h + (IA+FF) + 16h

- Start word: 10h

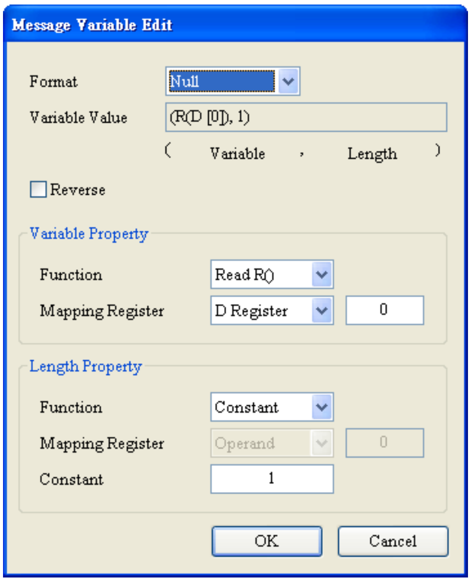


Message Constant Edit

Format: Hex

Value: 10

- Read the device address from D0 (IA).



Message Variable Edit

Format: Null

Variable Value: (R(D [0]), 1)

(Variable , Length)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

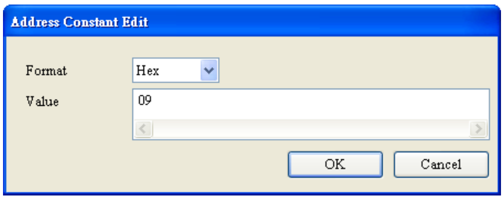
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Function code (FF): 09h

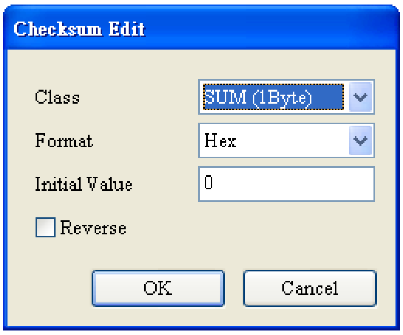


Address Constant Edit

Format: Hex

Value: 09

- Checksum (1byte; adding the previous two items up):



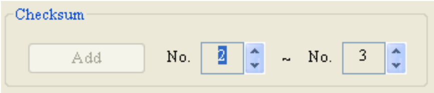
Checksum Edit

Class: SUM (1Byte)

Format: Hex

Initial Value: 0

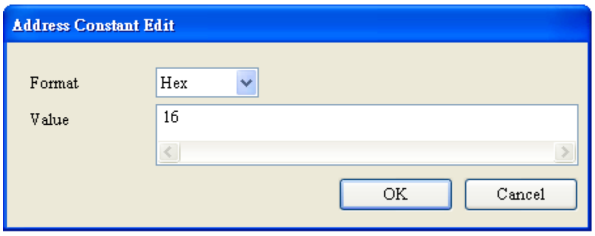
□ Reverse



Checksum

Add No. 2 ~ No. 3

- End word: 16h

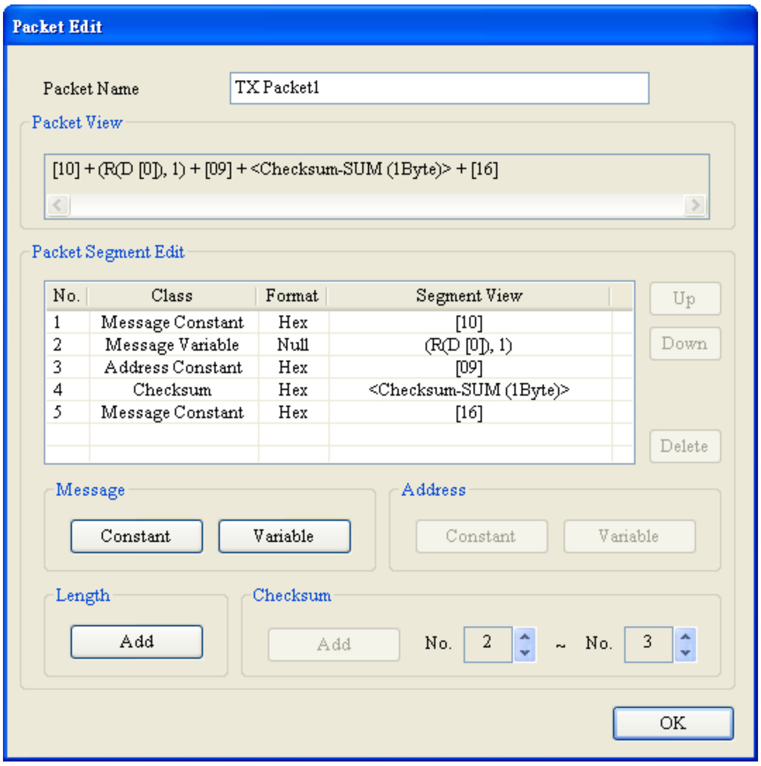


Address Constant Edit

Format: Hex

Value: 16

- The editing is complete:



Packet Edit

Packet Name: TX Packet1

Packet View

[10] + (R(D [0]), 1) + [09] + <Checksum-SUM (1Byte)> + [16]

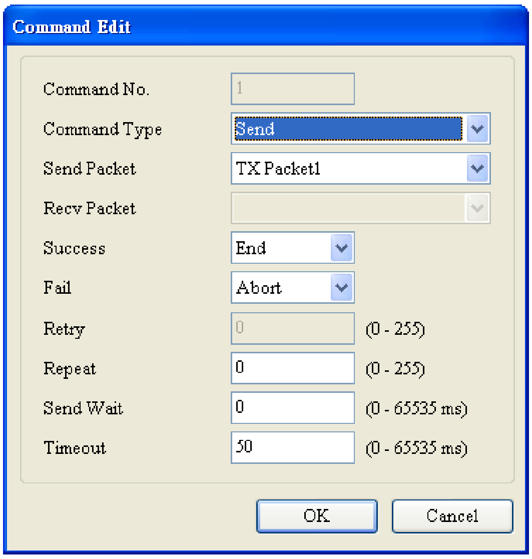
Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R(D [0]), 1)
3	Address Constant	Hex	[09]
4	Checksum	Hex	<Checksum-SUM (1Byte)>
5	Message Constant	Hex	[16]

Checksum: No. 2 ~ No. 3

There is no response address for type 1, so the user does not need to edit the function code of the response (Rx).

Edit the command: Sending Tx Packet1; no response address



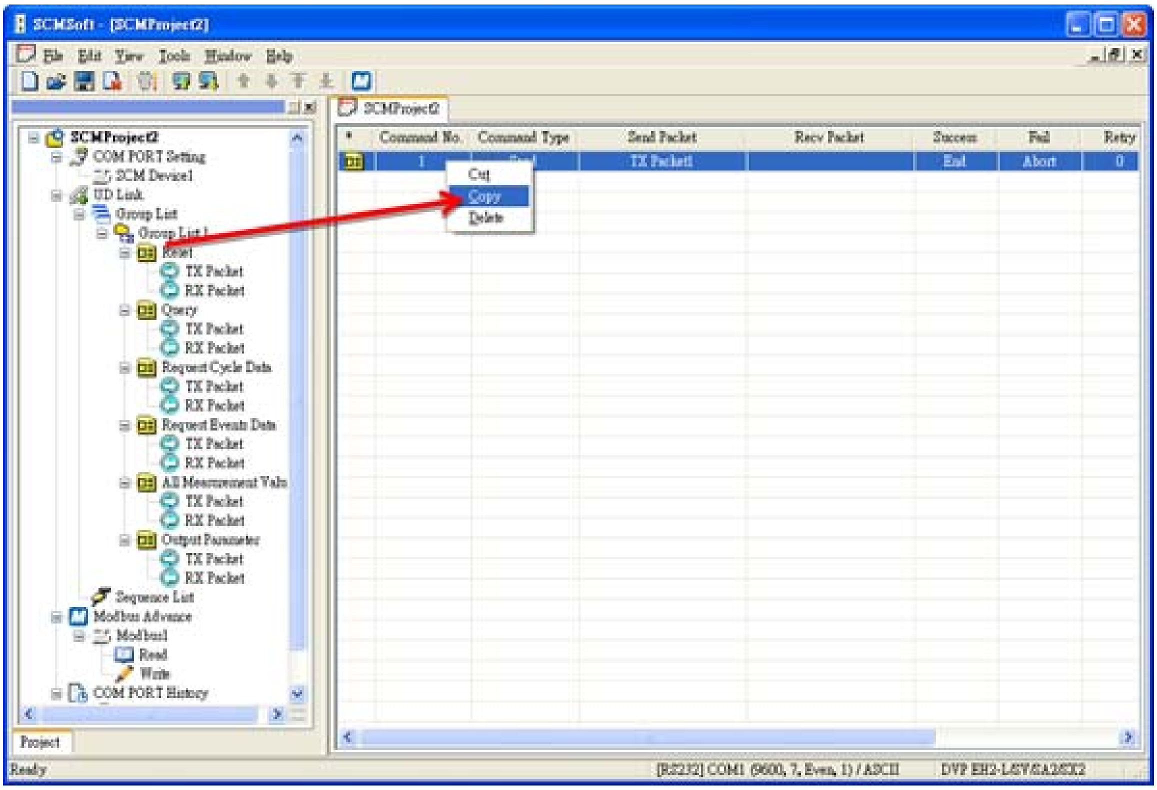
Command Edit

Command No.: 1
Command Type: Send
Send Packet: TX Packet1
Recv Packet:
Success: End
Fail: Abort
Retry: 0 (0 - 255)
Repeat: 0 (0 - 255)
Send Wait: 0 (0 - 65535 ms)
Timeout: 50 (0 - 65535 ms)

【Type 2】

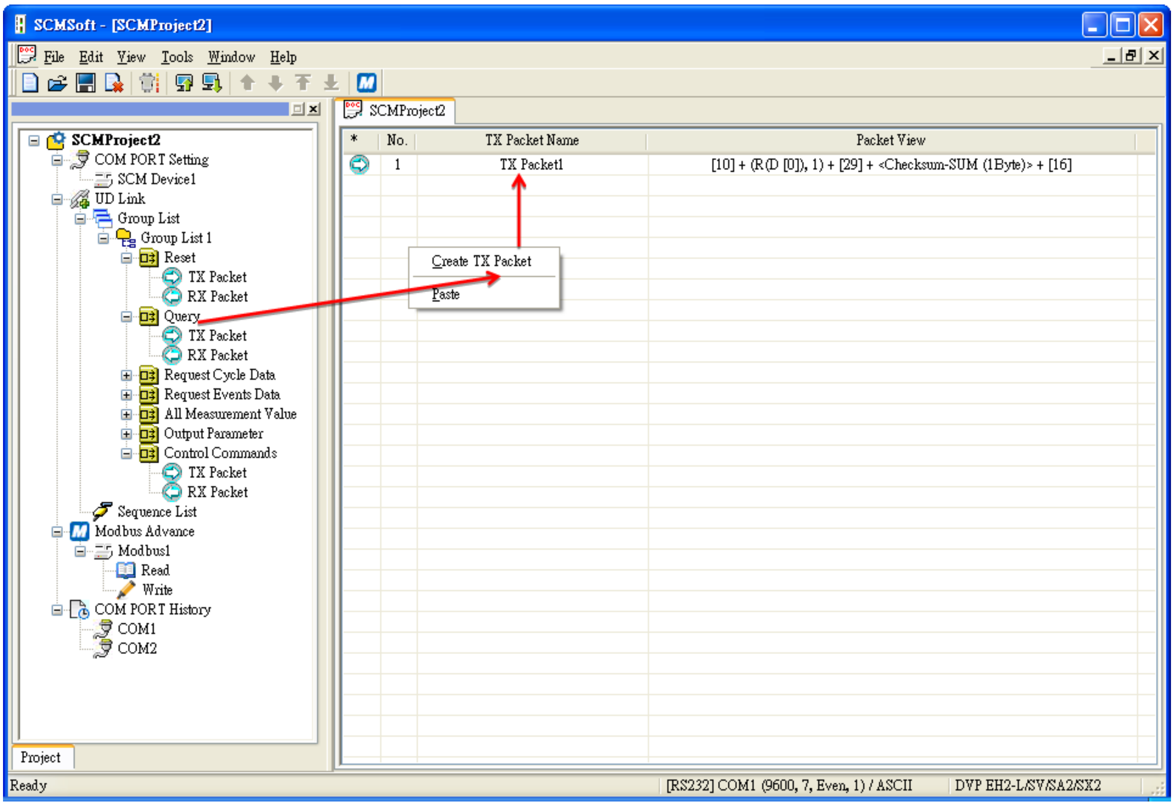
Send the abbreviated record, and respond with the abbreviated record. The setting of the sending is as that in type 1. The user can copy the setting directly. Notice that the function code is 29h.

- Copy the setting in Reset group.



*	Command No.	Command Type	Send Packet	Recv Packet	Success	Fail	Retry
	1	Send	TX Packet1		End	Abort	0

- Paste the setting to TX Packet in Query group.



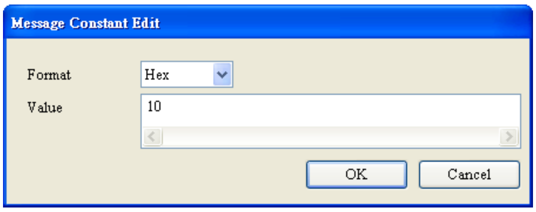
*	No.	TX Packet Name	Packet View
	1	TX Packet1	[10] + (R(D [0]), 1) + [29] + <Checksum-SUM (1Byte)> + [16]

Respond with the abbreviated record.

『Start word』+『Device address (IA)』+『Function code (FF)』+『Checksum (CS)』+『End marker』

→ 10h + D0 + 09h + (IA+FF) + 16h

- Start word: 10h

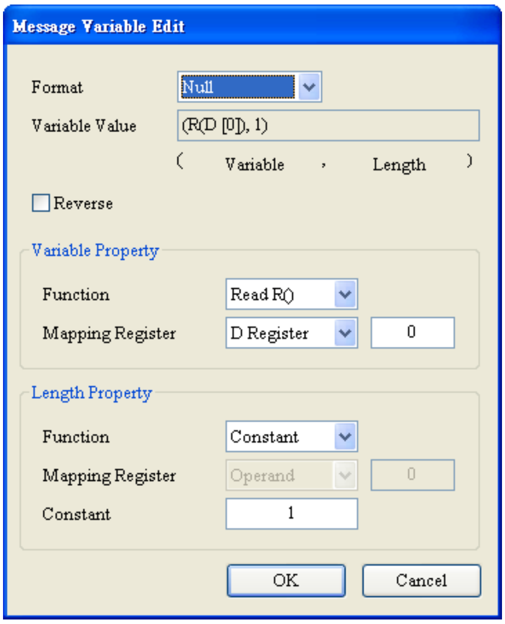


Message Constant Edit

Format: Hex

Value: 10

- Check whether the response address and the device address previously read from D0 (IA) are the same.



Message Variable Edit

Format: Null

Variable Value: (R(D [0]), 1)

(Variable , Length)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

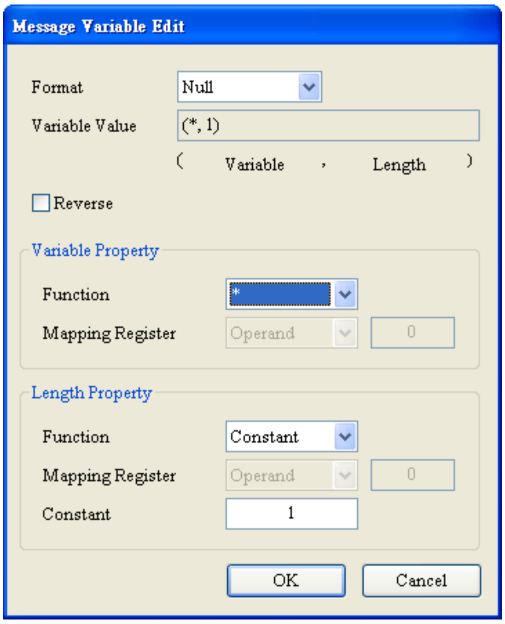
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Ignore the function code (FF) of the response: (*, 1): Ignore the word whose length is 1. If the user wants to store the function code, the user can refer to the setting of the device address (IA) to store the function code in the D register.



Message Variable Edit

Format: Null

Variable Value: (*, 1)

(Variable , Length)

□ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

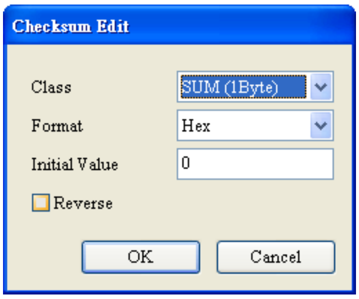
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Checksum (1byte, adding the previous two items up):



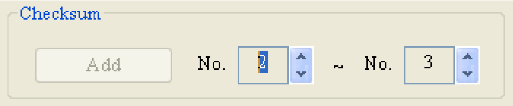
Checksum Edit

Class: SUM (1Byte)

Format: Hex

Initial Value: 0

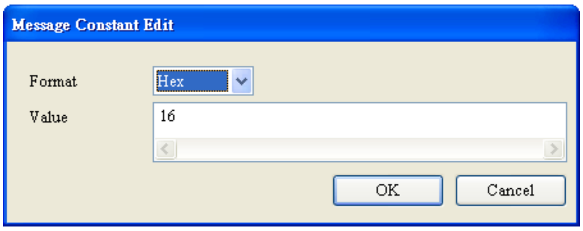
□ Reverse



Checksum

No. 2 ~ No. 3

- End word: 16h

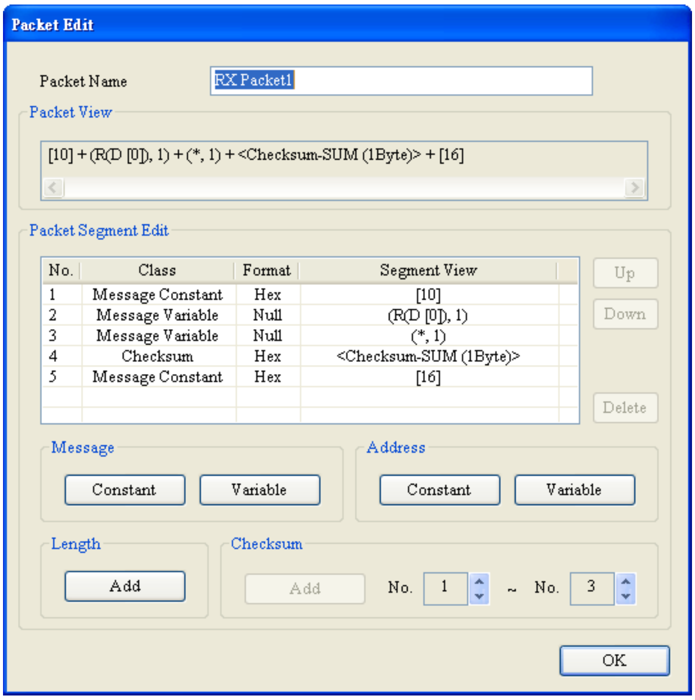


Message Constant Edit

Format: Hex

Value: 16

- The editing is complete:



Packet Edit

Packet Name: RX Packet1

Packet View

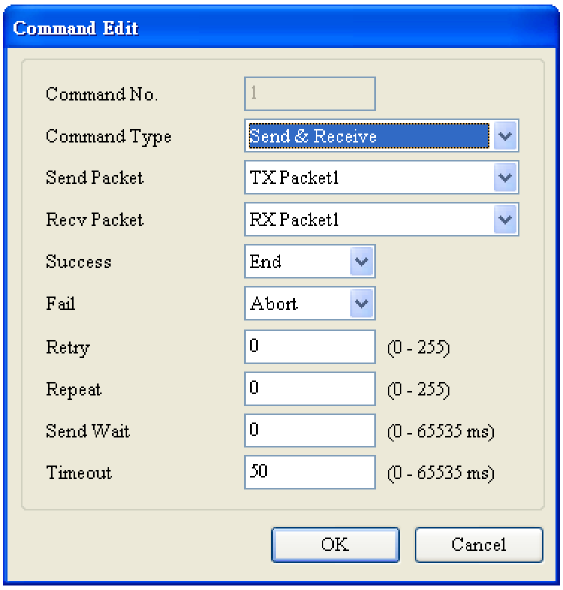
[10] + (R(D [0]), 1) + (*, 1) + <Checksum-SUM (1Byte)> + [16]

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R(D [0]), 1)
3	Message Variable	Null	(*, 1)
4	Checksum	Hex	<Checksum-SUM (1Byte)>
5	Message Constant	Hex	[16]

Checksum: No. 1 ~ No. 3

Edit the command: Sending Tx Packet1, and receiving Rx Packet1



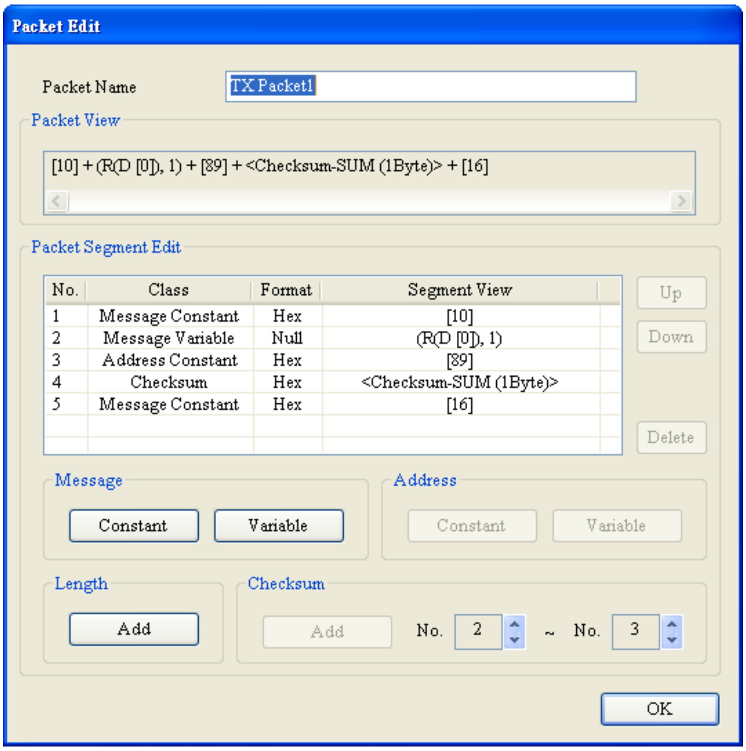
Command Edit

Command No.	1
Command Type	Send & Receive
Send Packet	TX Packet1
Recv Packet	RX Packet1
Success	End
Fail	Abort
Retry	0 (0 - 255)
Repeat	0 (0 - 255)
Send Wait	0 (0 - 65535 ms)
Timeout	50 (0 - 65535 ms)

【Type 3】

Send the abbreviated record, and respond with the full record.

For the sending of the abbreviated record, the user can copy or refer to those in type 1 and type 2. Notice that the function code (FF) is 89h.



Packet Edit

Packet Name: TX Packet1

Packet View: [10] + (R(D [0]), 1) + [89] + <Checksum-SUM (1Byte)> + [16]

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R(D [0]), 1)
3	Address Constant	Hex	[89]
4	Checksum	Hex	<Checksum-SUM (1Byte)>
5	Message Constant	Hex	[16]

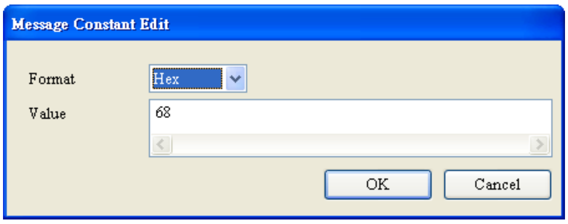
Checksum: No. 2 ~ No. 3

Respond with the full record.

『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Data block (DB)』+『Checksum (CS)』+『End marker』

→ 68h + (Null) + (Null) + 68h + D0 + (Null) + D100

- Start word: 68h

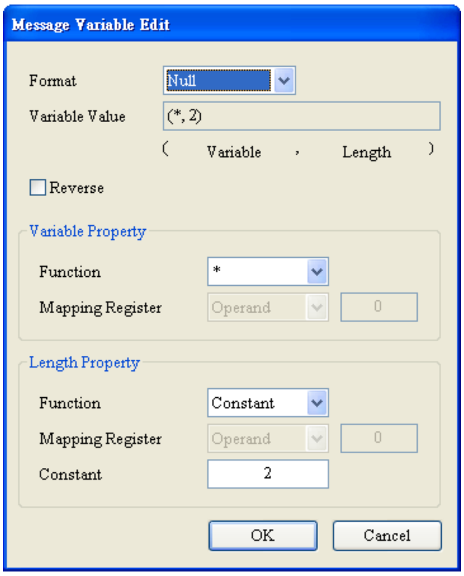


Message Constant Edit

Format: Hex

Value: 68

- Length + Length (repeat): Ignore these two words. They can be ignored or stored.



Message Variable Edit

Format: Null

Variable Value: (*, 2)

(Variable , Length)

□ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

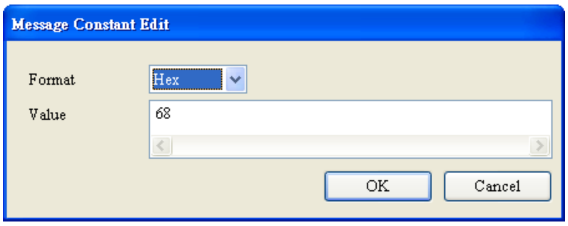
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 2

- Start word: 68h

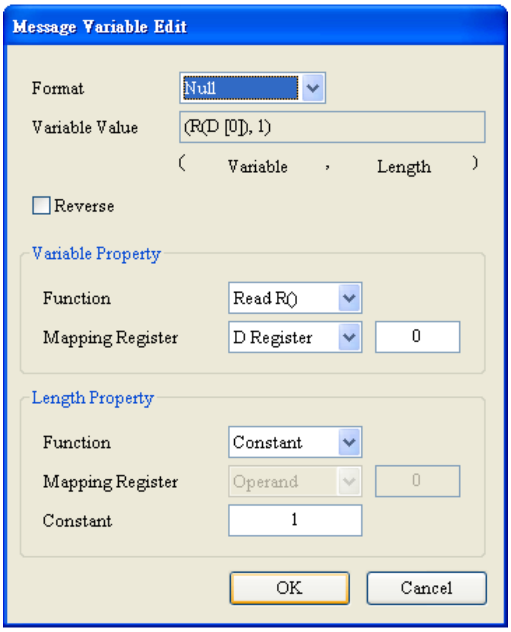


Message Constant Edit

Format: Hex

Value: 68

- Device address (IA): Check whether the response address and the device address previously read from D0 (IA) are the same.



Message Variable Edit

Format: Null

Variable Value: (R(D [0]), 1)

(Variable , Length)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

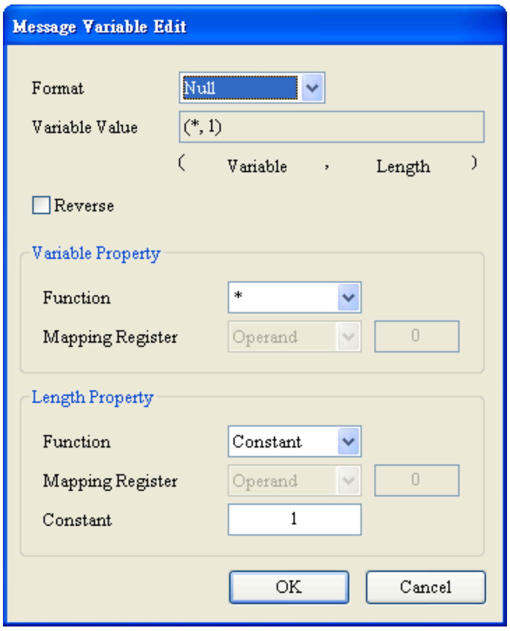
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Function code: Ignore the word.



Message Variable Edit

Format: Null

Variable Value: (*, 1)

(Variable , Length)

□ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

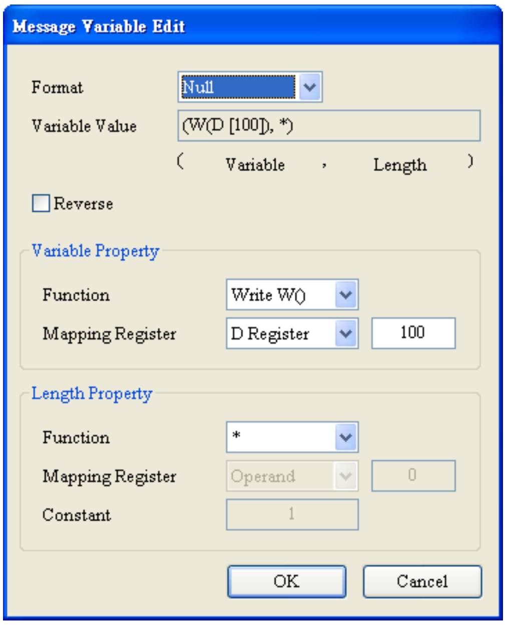
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- The data after the function code is stored in the registers starting from D100. (Note)



Message Variable Edit

Format: Null

Variable Value: (W(D [100]), *)

(Variable , Length)

□ Reverse

Variable Property

Function: Write W()

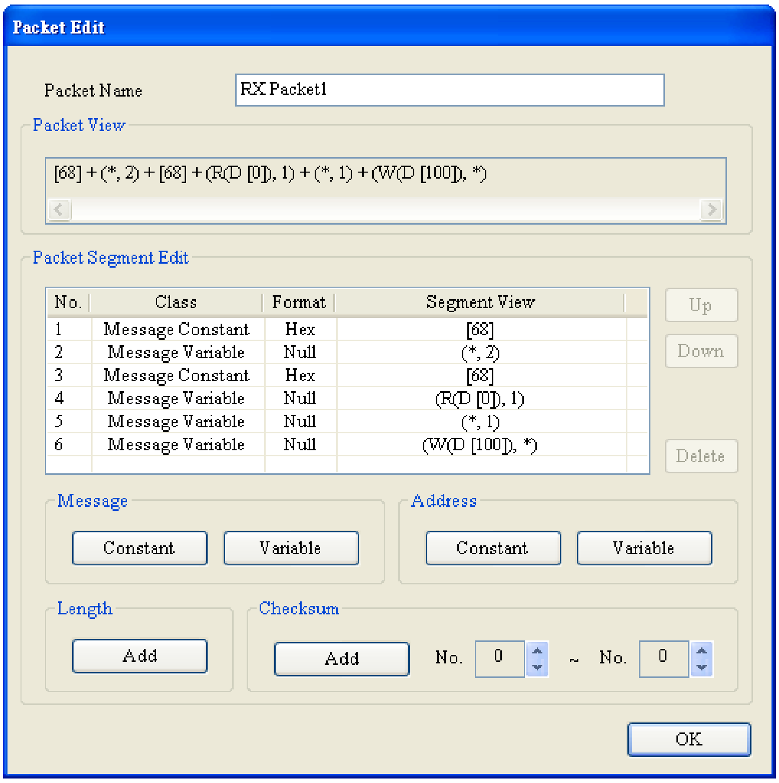
Mapping Register: D Register 100

Length Property

Function: *

Mapping Register: Operand 0

Constant: 1



Packet Edit

Packet Name: RX Packet1

Packet View

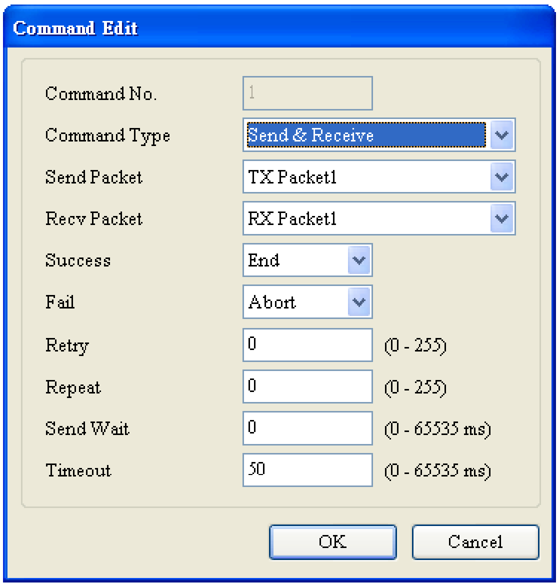
[68] + (*, 2) + [68] + (R(D [0]), 1) + (*, 1) + (W(D [100]), *)

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[68]
2	Message Variable	Null	(*, 2)
3	Message Constant	Hex	[68]
4	Message Variable	Null	(R(D [0]), 1)
5	Message Variable	Null	(*, 1)
6	Message Variable	Null	(W(D [100]), *)

Note: Some unimportant words can be ignored. The user can just store the data which is needed in the registers (Dx), and the data whose length of the response code is unknown can be stored in the registers by means of this method.

Edit the command: Sending Tx Packet1, and receiving Rx Packet1



Command Edit

Command No. 1

Command Type: Send & Receive

Send Packet: TX Packet1

Recv Packet: RX Packet1

Success: End

Fail: Abort

Retry: 0 (0 - 255)

Repeat: 0 (0 - 255)

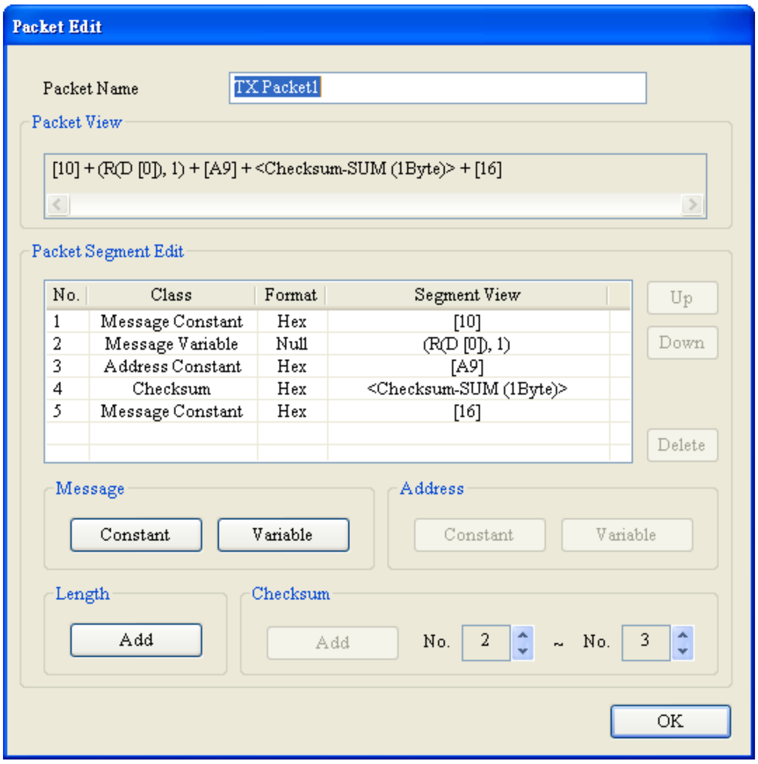
Send Wait: 0 (0 - 65535 ms)

Timeout: 50 (0 - 65535 ms)

【Type 4】

Send the abbreviated record, and respond with the full record.

For the sending of the abbreviated record, the user can copy or refer to those in type 1 and type 2. Notice that the function code (FF) is A9h.



Packet Edit

Packet Name: TX Packet1

Packet View

[10] + (R(D [0]), 1) + [A9] + <Checksum-SUM (1Byte)> + [16]

Packet Segment Edit

No.	Class	Format	Segment View
1	Message Constant	Hex	[10]
2	Message Variable	Null	(R(D [0]), 1)
3	Address Constant	Hex	[A9]
4	Checksum	Hex	<Checksum-SUM (1Byte)>
5	Message Constant	Hex	[16]

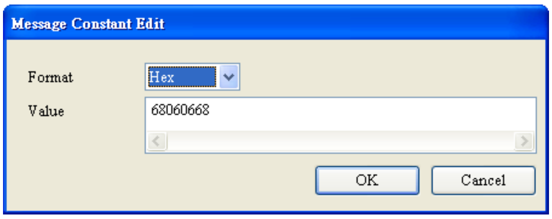
Checksum: No. 2 ~ No. 3

Respond with the full record.

『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Data block (DB)』+『Checksum (CS)』+『End marker』

→ 68h + 06h + 06h + 68h + D0 + (1 word) + (3 words) + (the content gotten from adding from IA to the end) + 16h

- Star word-Length-Length-Star word

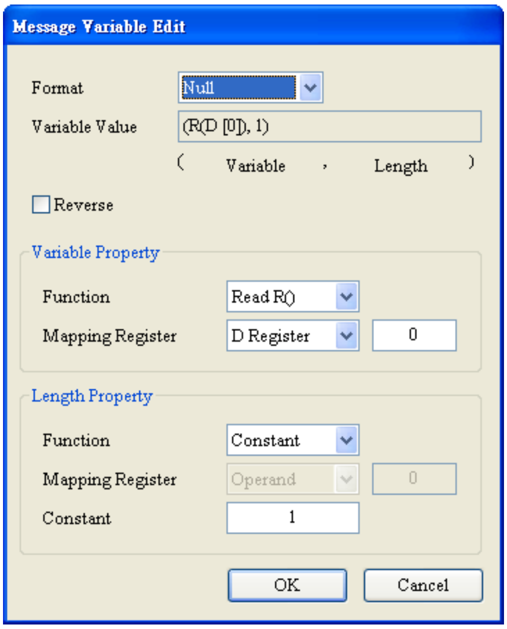


Message Constant Edit

Format: Hex

Value: 68060668

- Check whether the response address and the device address previously read from D0 (IA) are the same.



Message Variable Edit

Format: Null

Variable Value: (R(D [0]), 1)

(Variable , Length)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

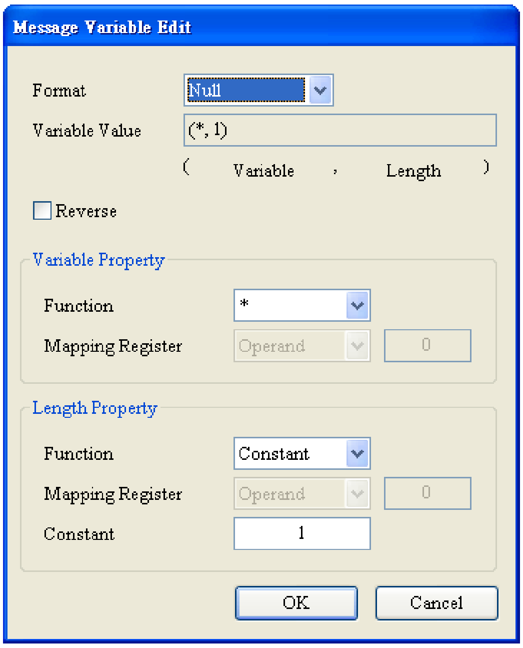
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- FF：Ignore the function code.



Message Variable Edit

Format: Null

Variable Value: (*, 1)

(Variable , Length)

□ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

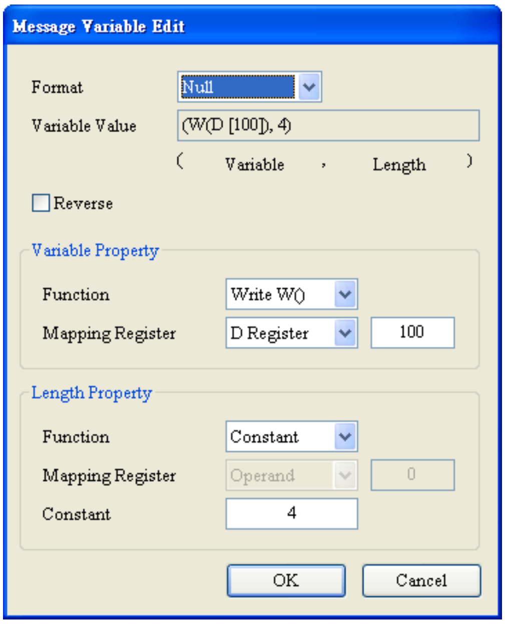
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Store PI+DB in D100.



Message Variable Edit

Format: Null

Variable Value: (W(D [100]), 4)

(Variable , Length)

□ Reverse

Variable Property

Function: Write W()

Mapping Register: D Register 100

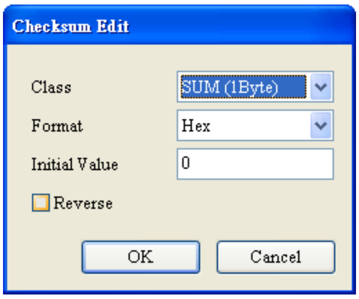
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 4

- Checksum:



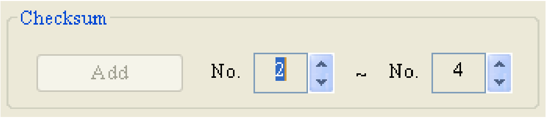
Checksum Edit

Class: SUM (1Byte)

Format: Hex

Initial Value: 0

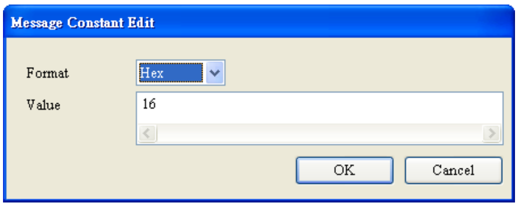
☐ Reverse



Checksum

No. 2 ~ No. 4

- End word:

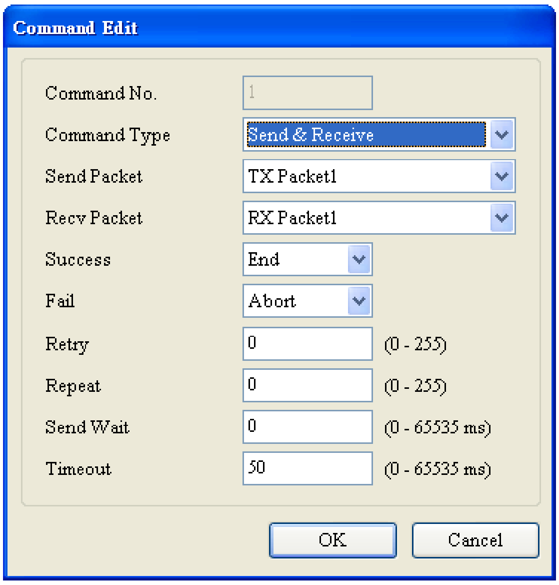


Message Constant Edit

Format: Hex

Value: 16

Edit the command: Sending Tx Packet1, and receiving Rx Packet1



Command Edit

Command No.: 1

Command Type: Send & Receive

Send Packet: TX Packet1

Recv Packet: RX Packet1

Success: End

Fail: Abort

Retry: 0 (0 - 255)

Repeat: 0 (0 - 255)

Send Wait: 0 (0 - 65535 ms)

Timeout: 50 (0 - 65535 ms)

【Type 5】

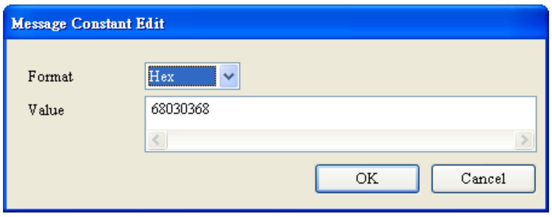
Send the abbreviated record, and respond with the full record.

When the control record is sent, the function code (FF) is 89h.

『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Checksum (CS)』+『End marker』

→ 68h + 03h + 03h + 68h + D0 + 89h + D1 + (the content gotten from adding from IA to the end) + 16h

- Start word-Length-Length-Start word

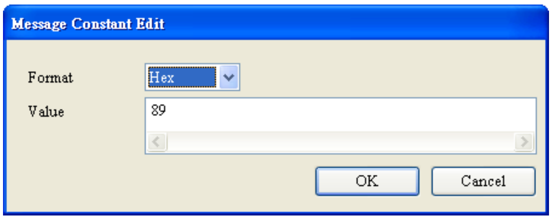


Message Constant Edit

Format: Hex

Value: 68030368

- Function code: 89h

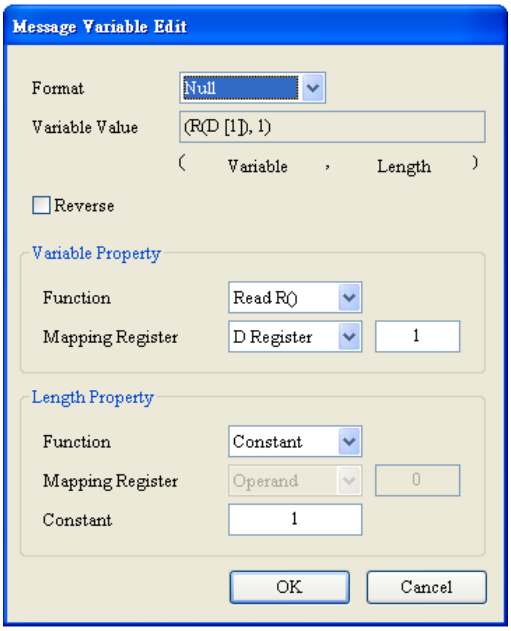


Message Constant Edit

Format: Hex

Value: 89

- The device address is read from D0.



Message Variable Edit

Format: Null

Variable Value: (R(D [1]), 1)

(Variable , Length)

☐ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 1

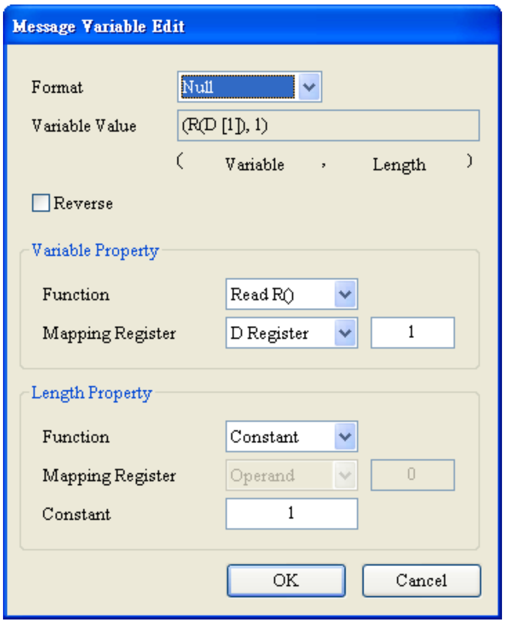
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- The parameter index is read from D1.



Message Variable Edit

Format: Null

Variable Value: (R(D [1]), 1)

(Variable , Length)

☐ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 1

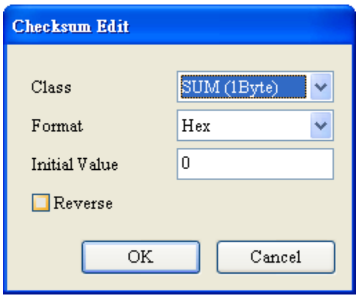
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Checksum:



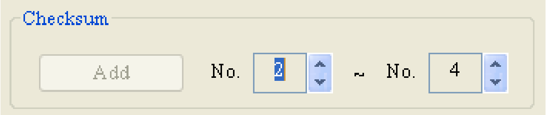
Checksum Edit

Class: SUM (1Byte)

Format: Hex

Initial Value: 0

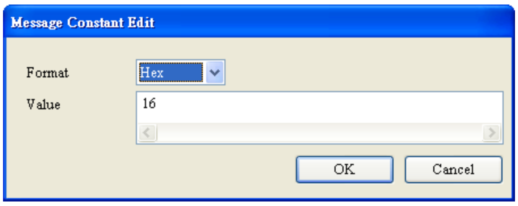
□ Reverse



Checksum

No. 2 ~ No. 4

- End word:



Message Constant Edit

Format: Hex

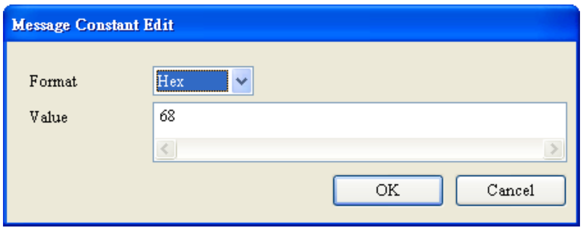
Value: 16

Respond with the full record.

『Start word』 + 『Length』 + 『Length (repeat)』 + 『Start word』 + 『Device address (IA)』 + 『Function code (FF)』 + 『Parameter index (PI)』 + 『Data block (DB)』 + 『Checksum (CS)』 + 『End marker』

→ 68h + (Null) + (Null) + 68h + D0 + (Null) + D1 + D100 + (the content gotten from adding from IA to the end) + 16h

- Start word:

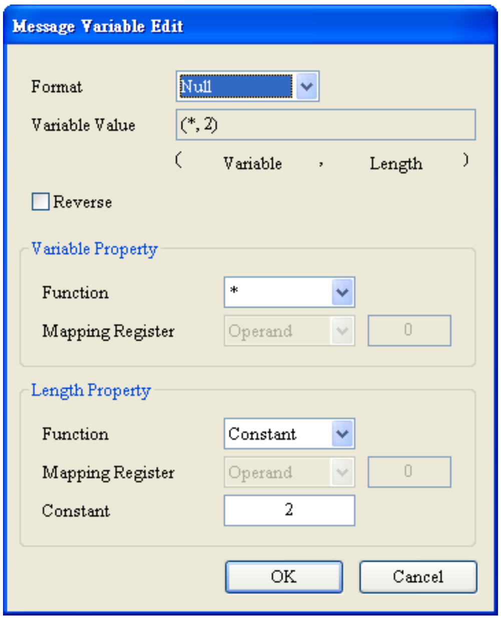


Message Constant Edit

Format: Hex

Value: 68

- Length-Length (two words): Ignore the two words.



Message Variable Edit

Format: Null

Variable Value: (*, 2)

(Variable , Length)

□ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

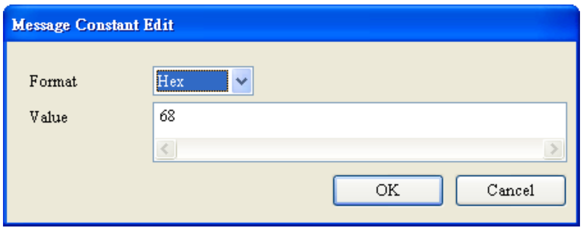
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 2

- Start word: 68h

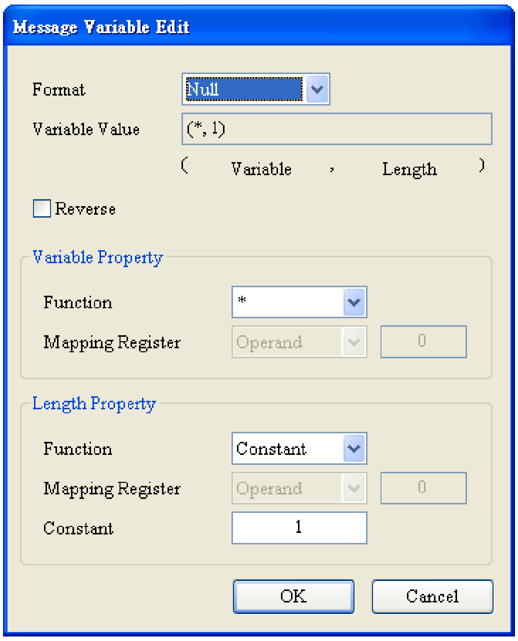


Message Constant Edit

Format: Hex

Value: 68

- Function code:



Message Variable Edit

Format: Null

Variable Value: (*, 1)

(Variable , Length)

□ Reverse

Variable Property

Function: *

Mapping Register: Operand 0

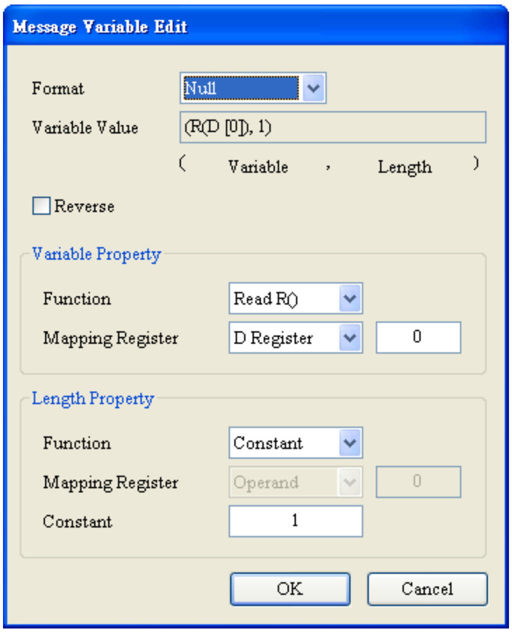
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Check whether the response address and the device address previously read from D0 (IA) are the same.



Message Variable Edit

Format: Null

Variable Value: (R(D [0]), 1)

(Variable , Length)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 0

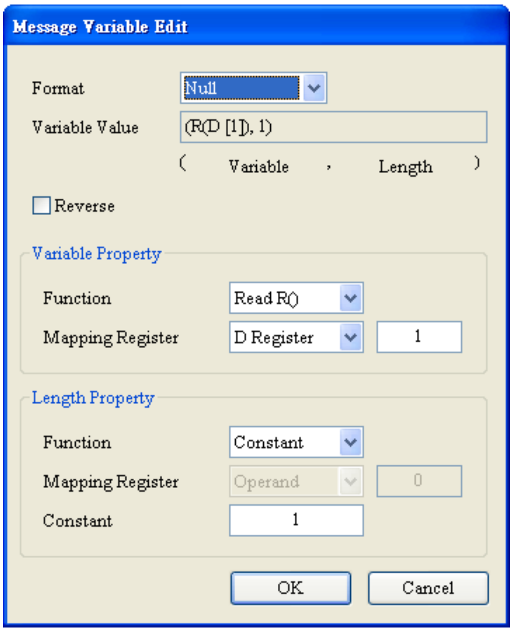
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Check whether the parameter index of the receiving and that of the sending are the same.



Message Variable Edit

Format: Null

Variable Value: (R(D [1]), 1)

(Variable , Length)

□ Reverse

Variable Property

Function: Read R()

Mapping Register: D Register 1

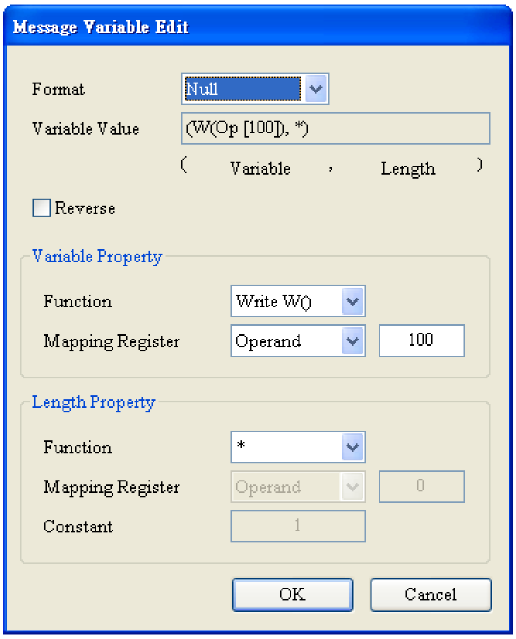
Length Property

Function: Constant

Mapping Register: Operand 0

Constant: 1

- Data block: The response data is stored in the registers starting from D100.



Format: Null

Variable Value: (W(Op [100]), *)

(Variable , Length)

□ Reverse

Variable Property

Function: Write W()

Mapping Register: Operand 100

Length Property

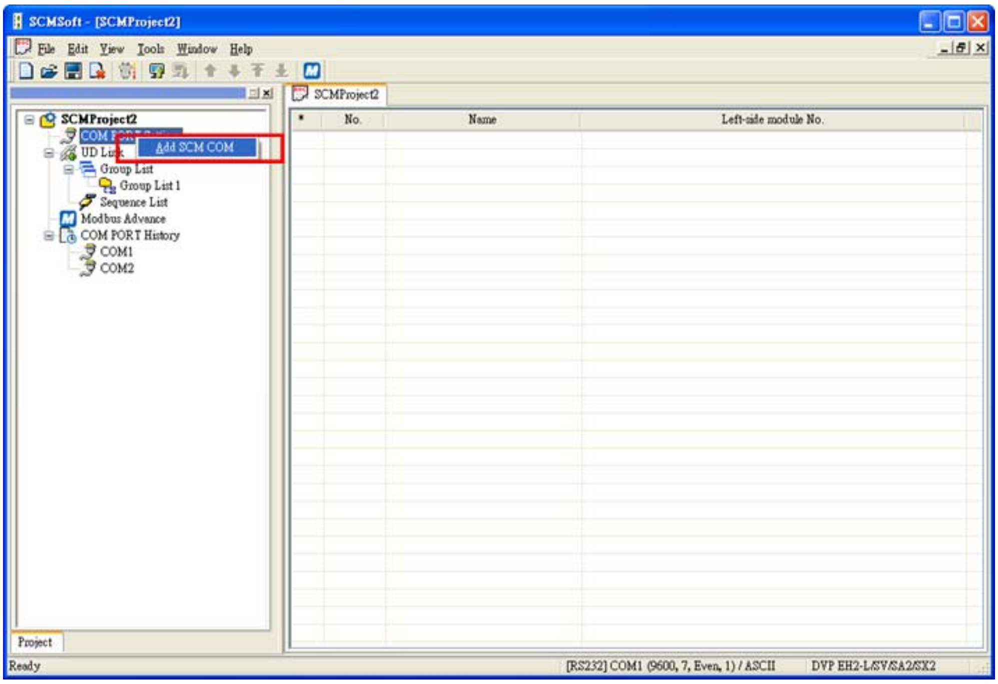
Function: *

Mapping Register: Operand 0

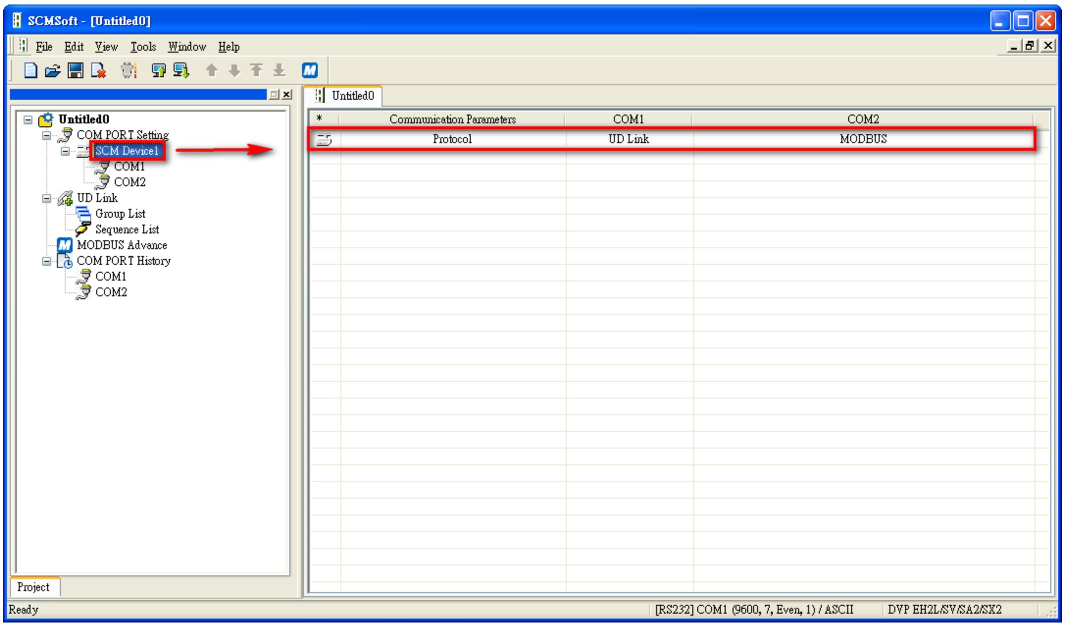
Constant: 1

(4) Download

After setting all types, the user can download the UD Link to the SCM module. Open SCMSoft → 『New Project』 → COM PORT Setting: 『Add SCM COM』 → Set the communication parameters

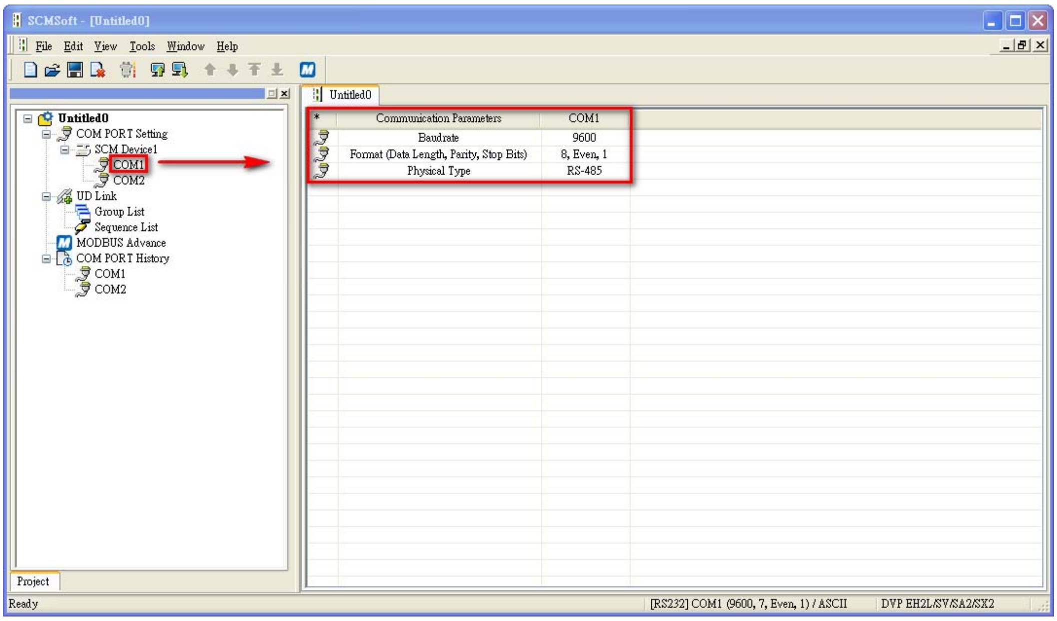


Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2-L/SV/SA2/SX2



*	Communication Parameters	COM1	COM2
	Protocol	UD Link	MODBUS

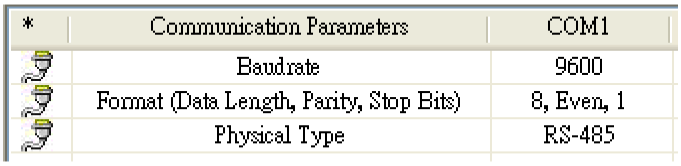
[RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2



*	Communication Parameters	COM1
	Baudrate	9600
	Format (Data Length, Parity, Stop Bits)	8, Even, 1
	Physical Type	RS-485

[RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2

Set the communication parameters of COM1: Station address 247 (default), UD Link, 9600, 8, Even, 1.



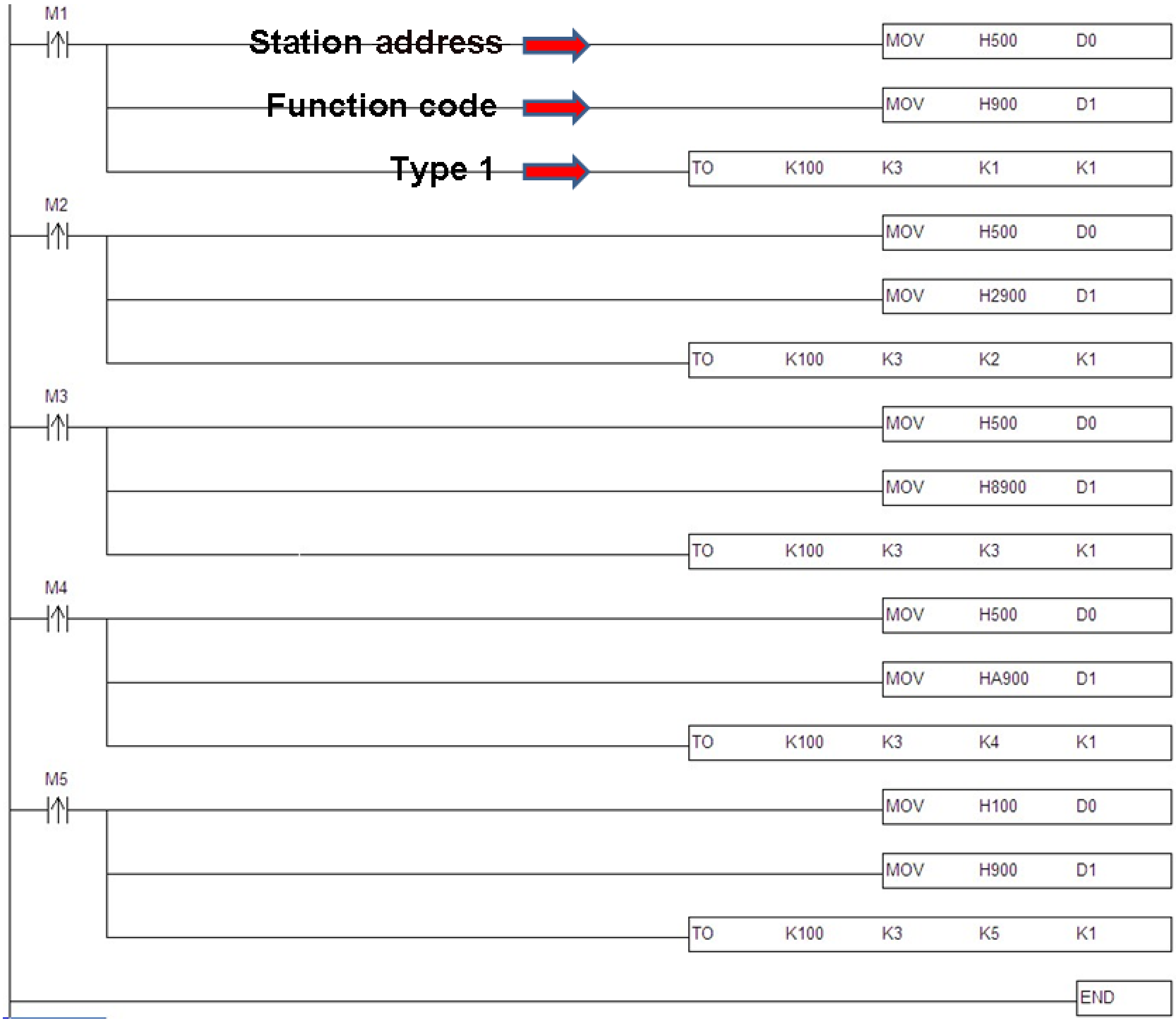
*	Communication Parameters	COM1
	Baudrate	9600
	Format (Data Length, Parity, Stop Bits)	8, Even, 1
	Physical Type	RS-485

(5) WPLSoft triggers UD Link

The group number set in each type is triggered by “To instruction” in WPLSoft which triggers the execution of UD Link. K1 is written into CR3 if the group number is 1 and by analogy, K2 is written into CR3 if the group number is 2.

CR#	Attribute	Name of the register	Description
3	R/W	Group number triggered by COM1 UD Link	The Group number triggered by COM1 UD Link

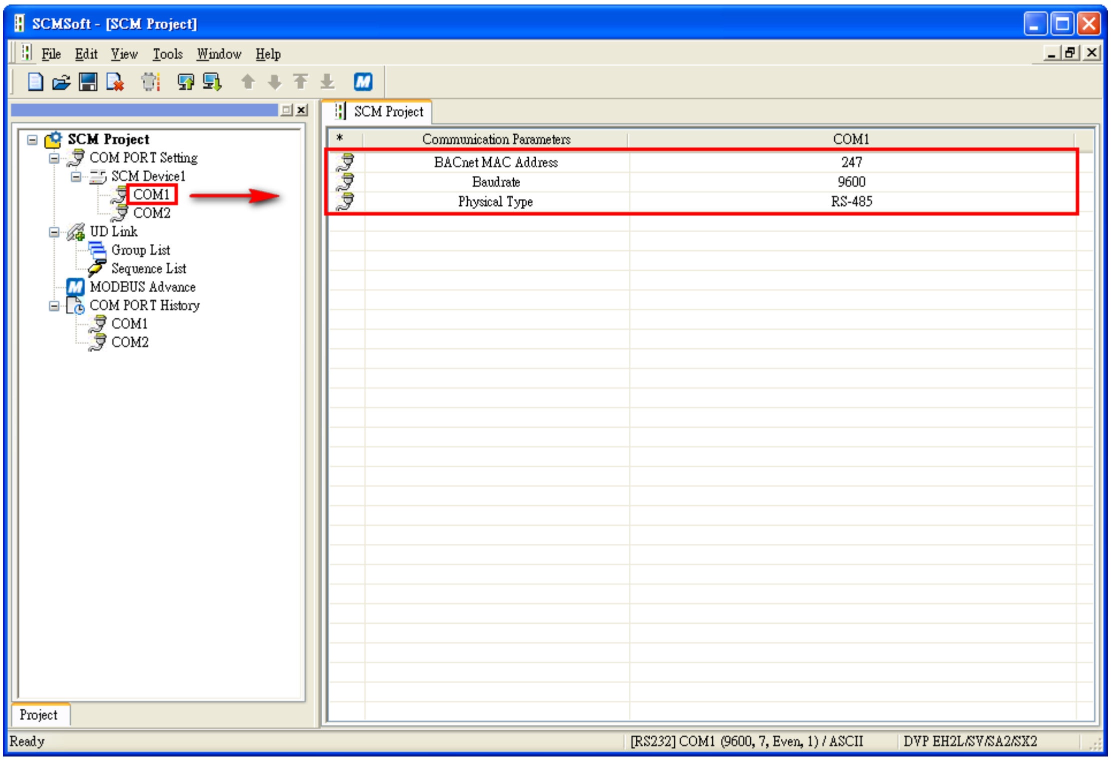
The sending of type 1~5 is controlled by M1~M5. Each triggering includes writing the station address of the electricity meter and the function code into D0 and D1 respectively. When the data is written into the registers, the higher bit precedes the lower bit. For example, the user has to enter H’0555 when the station address is 5, and the same applies to the reading of the response address from D100.



7.4 BACnet MS/TP Slave Function (Supported by DVPSCM52-SL)

Set the BACnet parameters and the BACnet object for the SCM module, and then download them to the SCM module to connect to the BACnet MS/TP module.

【BACnet parameters】



*	Communication Parameters	COM1
	BACnet MAC Address	247
	Baudrate	9600
	Physical Type	RS-485

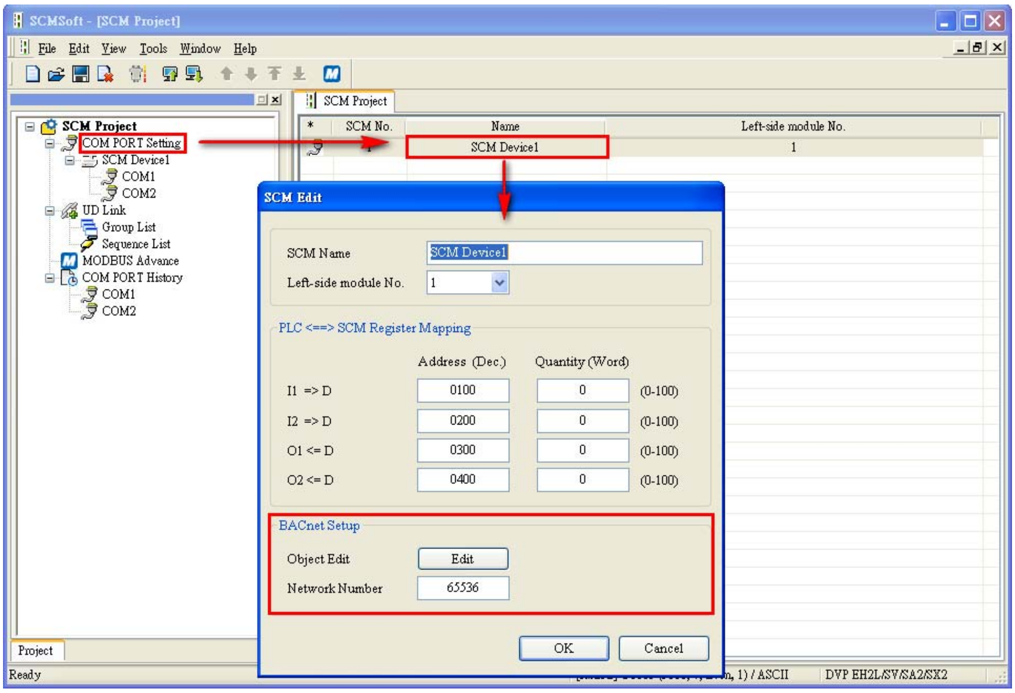
BACnet MAC address: 1 ~ 247 (Default: 247). It can not be the same as the address of other device on the BACnet network.

Baud rate: 1200 bps~460800 bps (Default: 9600). It must be the same as the setting for the BACnet MS/TP MPU.

Physical Type: The user can select RS-485 or RS-422.

【BACnet object】

Network Number: The network number on the BACnet network is unique. It can not be used repeatedly. (Default: 65536)



SCMSoft - [SCM Project]

*	SCM No.	Name	Left-side module No.
	1	SCM Device1	1

SCM Edit

SCM Name: SCM Device1

Left-side module No.: 1

PLC <==> SCM Register Mapping

	Address (Dec.)	Quantity (Word)	
I1 => D	0100	0	(0-100)
I2 => D	0200	0	(0-100)
O1 <= D	0300	0	(0-100)
O2 <= D	0400	0	(0-100)

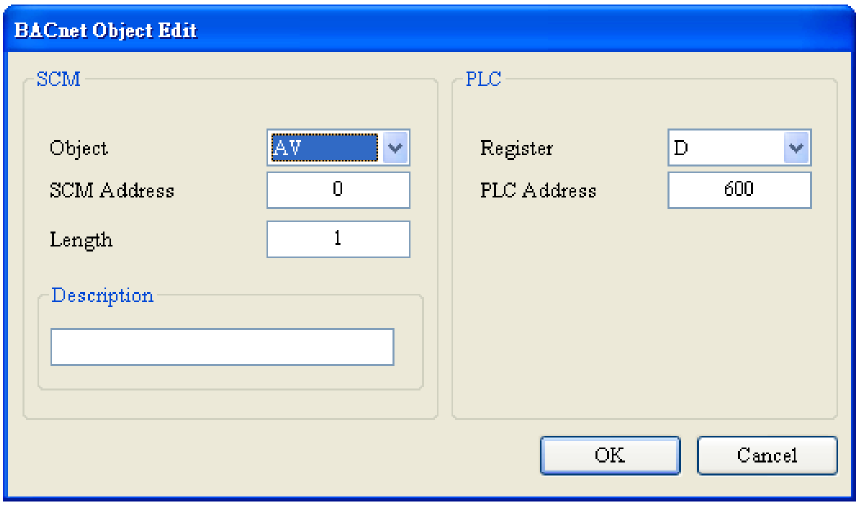
BACnet Setup

Object Edit: Edit

Network Number: 65536

BACnet object edit: Editing the AV and BV values which correspond to the data registers and coils in the Delta PLC master connecting to the SCM module

The lenghth of tha AV value corresponds to two data registers in the Delta PLC, and the lenghth of the BV value corresponds to one coil in the Delta PLC.



BACnet Object Edit

SCM

Object: AV

SCM Address: 0

Length: 1

Description

PLC

Register: D

PLC Address: 600

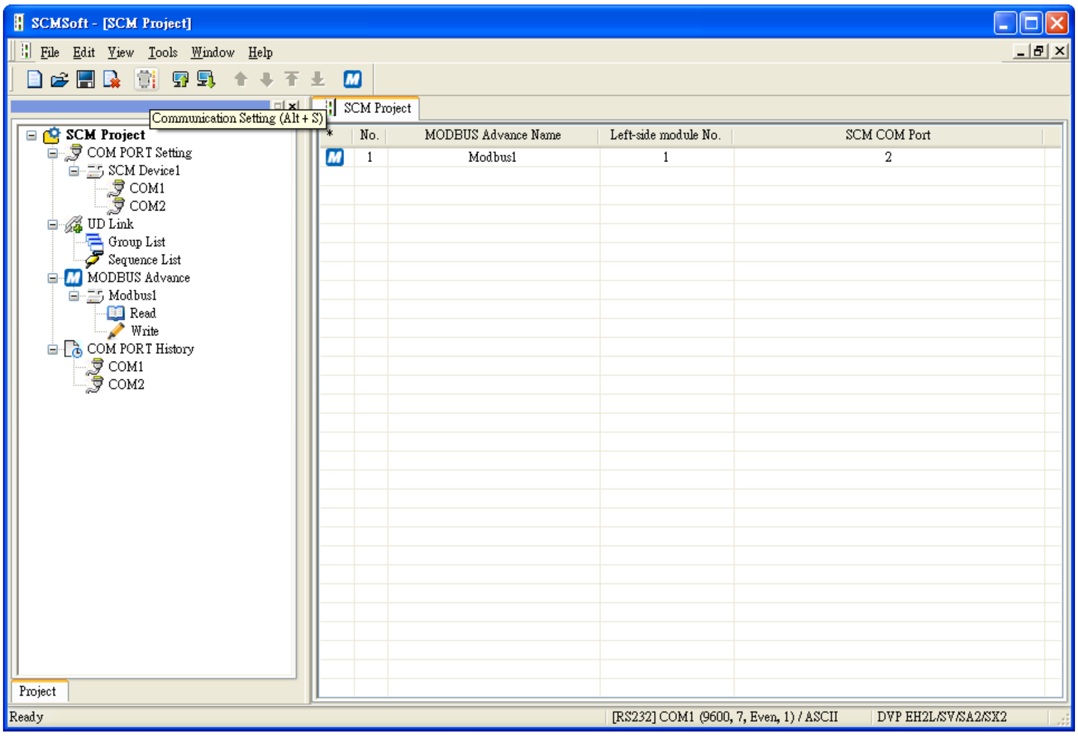
Object: The user can select “AV” or “BV”. “AV” corresponds to data the registers in the PLC, and “BV” corresponds the coil in the PLC.

SCM address: The user can set the address of the AV, or the address of the BV. The setting range is 0~383.

Length: A unit is a double word.

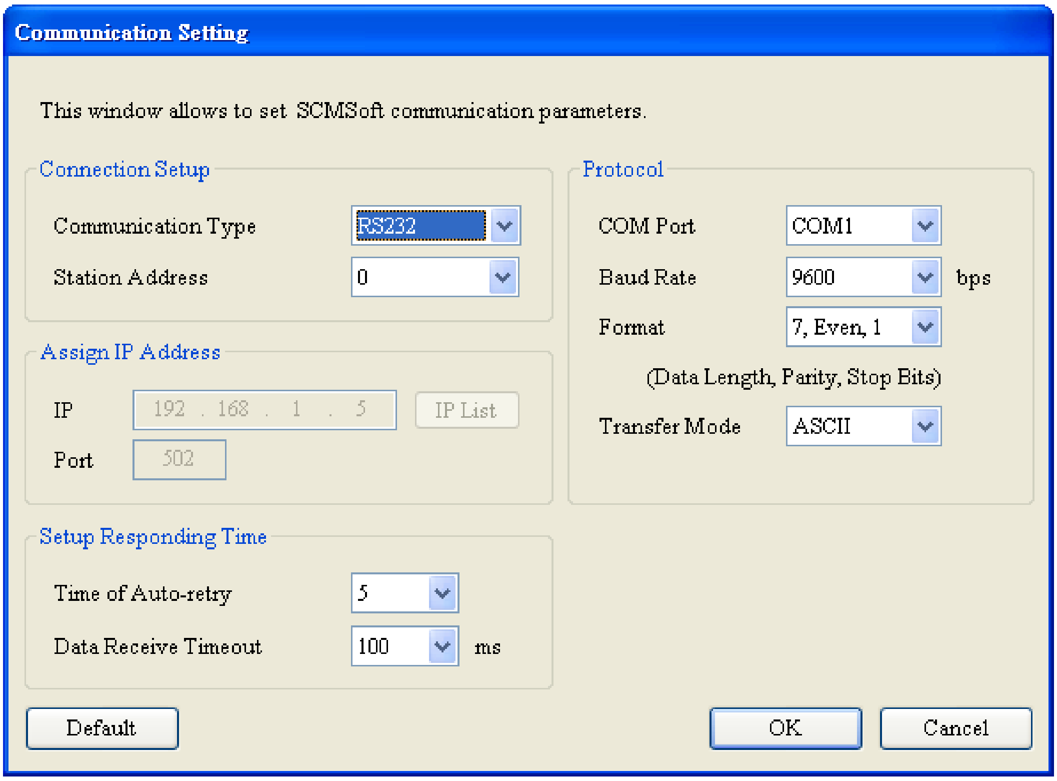
PLC: The start address in the Delta PLC.

【Downloading the parameters】



No.	MODBUS Advance Name	Left-side module No.	SCM COM Port
1	Modbus1	1	2

Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2


Communication Setting

This window allows to set SCMSoft communication parameters.

Connection Setup

Communication Type: RS232

Station Address: 0

Assign IP Address

IP: 192 . 168 . 1 . 5

Port: 502

Setup Responding Time

Time of Auto-retry: 5

Data Receive Timeout: 100 ms

Protocol

COM Port: COM1

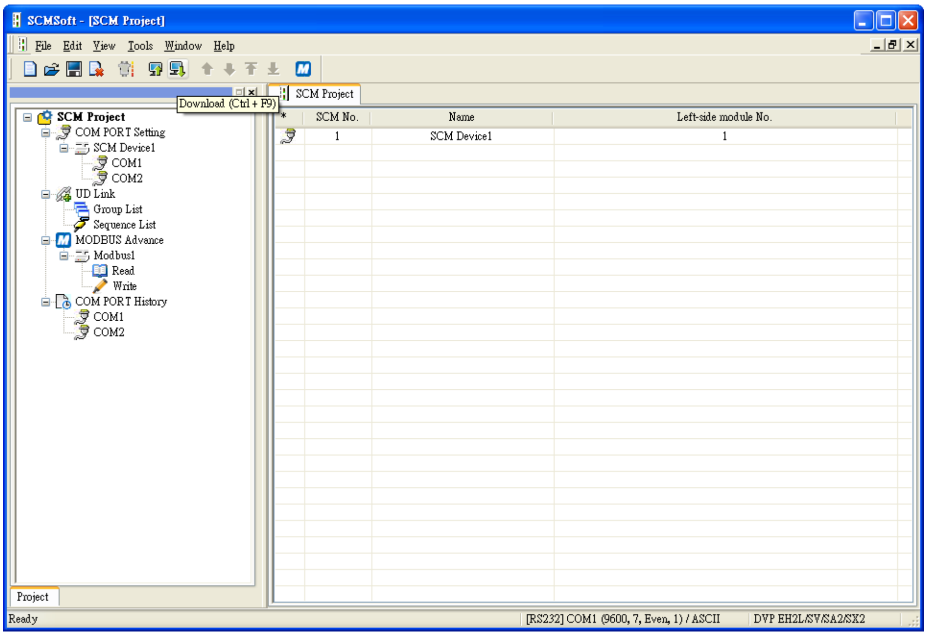
Baud Rate: 9600 bps

Format: 7, Even, 1

(Data Length, Parity, Stop Bits)

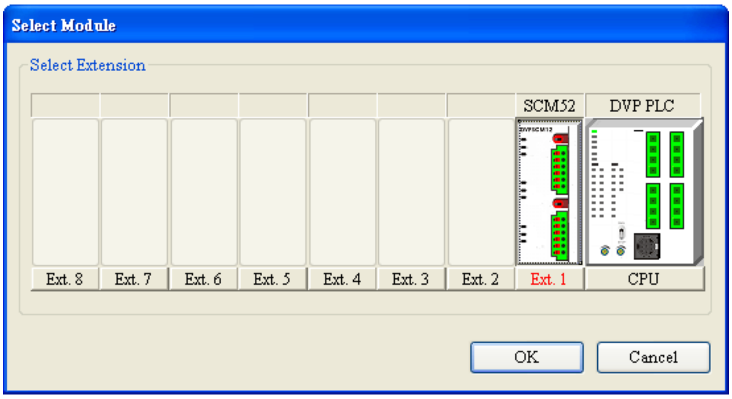
Transfer Mode: ASCII

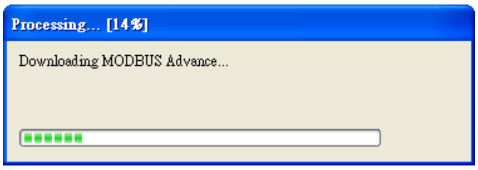
Click “Download”, choose the left-side module which will be downloaded, and click “OK”. If only one device is connected, click “OK” directly.



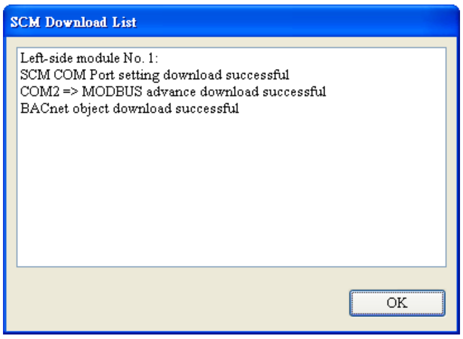
SCM No.	Name	Left-side module No.
1	SCM Device1	1

Ready [RS232] COM1 (9600, 7, Even, 1) / ASCII DVP EH2L/SV/SA2/SX2





Downloading MODBUS Advance...



Left-side module No. 1:
SCM COM Port setting download successful
COM2 => MODBUS advance download successful
BACnet object download successful

After the parameters are downloaded, the AV and BV values in the software correspond to the registers and bit in the PLC connected to the SCM module.

8. Error Flags

CR#	Description
CR#11	Error code
CR#12	Hardware error flag
CR#13	COM1 UD Link error flag
CR#14	COM2 UD Link error flag
CR#15	COM1 Modbus error flag
CR#16	COM2 Modbus error flag
CR#17	COM1 communication error flag
CR#18	COM2 communication error flag
CR#19	Internal communication error flag

Contents of the error flags

CR#11

Error code	Description
0x0001	Hardware error
0x0002	UD Link error
0x0004	There is a communication error in the communication port.
0x0008	MODBUS communication error

CR#12

Bit	15 ~ 4	3	2	1	0
Description	Reserved	LV occurs.	SRAM is damaged.	GPIO is damaged.	FLASH is damaged.

CR#13, CR#14

Bit	3	2	1	0
Description	There is a comparison error in the data received.	Packet editing error	The command number is not found.	The group number is not found.
Bit	7	6	5	4
Description	The data received is beyond expectation.	The data received is not sufficient for the comparison of the data	Reserved	Checksum error
Bit	11	10	9	8
Description	Unknown Rx packet segment format	Unknown Rx packet segment format	Unknown processing procedure	UD Link data check error
Bit	15	14	13	12
Description	Reserved	Reserved	The length written into the register exceeds the range of the module.	The length read from the register exceeds the range of the module.

CR#15, CR#16

Error code	Name	Description
0x0001	Illegal function	Unsupported function code
0x0002	Illegal data address	Unsupported address
0x0003	Illegal data value	Unsupported data value
0x0004	Slave device failure	The slave fails.
0x0005	Transform failure	Value conversion error

CR#17, CR#18, CR#19

Bit	3	2	1	0
Description	Communication timeout error	It is too late to receive the data.	Parity check error	There is an error in the sending format.
Bit	7	6	5	4
Description	Reserved	Internal communication error	Internal communication timeout	Checksum error
Bit	11	10	9	8
Description	Reserved		The buffer for the receiving is full.	The buffer for the sending is full.
Bit	15	14	13	12
Description	Reserved			