DELTA ELECTRONICS, INC.

# DVPSCM12-SL

***DVP Serial Communication Module-RS422, RS485***

# Operation Manual

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

DVP-0186520-01

# ⚠ 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

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

DVPSCM12-SL is a serial communication module. It supports Modbus RS-422, RS-4854, and the user-defined format of RS-485, and can be used as RS-422 communication port or RS-485 communication port to upload or download WPLSoft. SCMSoft, the setting software of DVPSCM12-SL, is built in Delta communication software DCISoft. Please download DCISoft_v1.07 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 can be used as PLC COM3 to upload or download WPLSoft.
- It has Modbus RS-422 data exchange and RS-485 data exchange (MODBUS Advance).
- It has user-defined communication protocols, and the process planning (UD Link) function.
- The MPUs it supports: DVPSA2 (V1.0), DVPSX2 (V1.2), DVPSV (V2.2) series.

## 1.2 Specifications

■ The serial communication interface (RS422/RS-485)

| Item | Specifications |
|---|---|
| Terminal | European terminal blocks with spring plugs |
| Transmission | RS-485/RS-422 |
| Transmission speed | 1200, 2400, 4800, 9600, 19200, 38400, 57600, 76800, 115200, 230400, 460800 bps |
| Communication format | Stop bit: 1, 2; Parity bit: None, Odd, Even; Data bit: 7, 8 |
| Communication protocol | Modbus ASCII/RTU, UD Link |

■ Environmental specifications

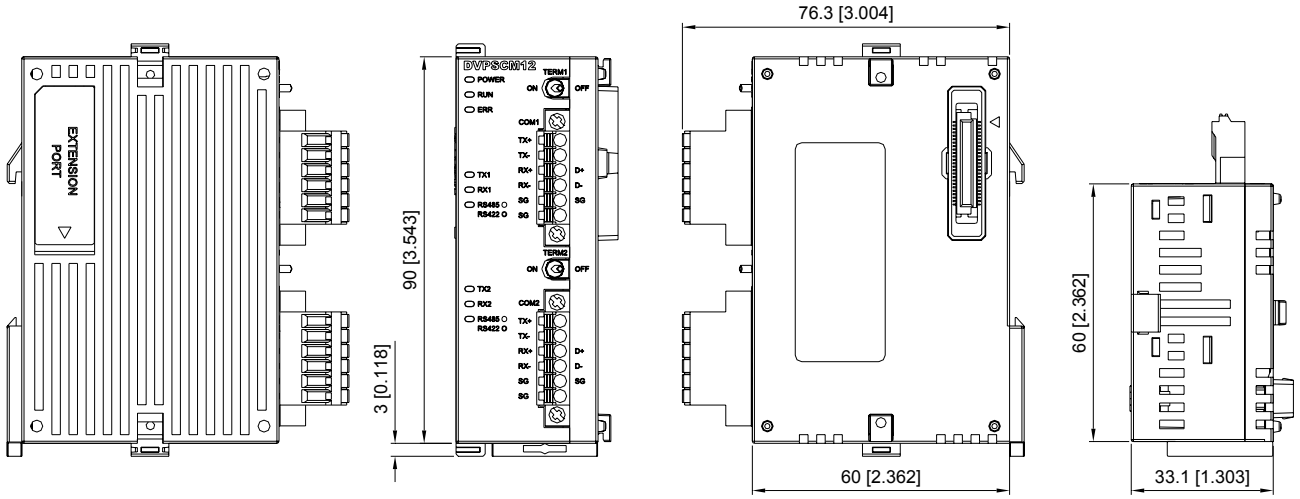
| Item | Specifications |
|---|---|
| Noise immunity | ESD (IEC 61131-2, IEC 61000-4-2): 8 kV Air Discharge<br>EFT (IEC 61131-2, IEC 61000-4-4): Communication I/O：±2 kV<br>CS (IEC 61131-2, IEC 6100-4-6): 0.15 ~ 80 MHz, 3 Vrms<br>RS (IEC 61131-2, IEC 61000-4-3): 80 ~ 100 MHz, 10 V/m, 1.4 ~ 2.0 GHz |
| Operating/Storage temperature | Operation: 0 ~ 55°C (temperature), 50 ~ 95% (humidity), the class of pollution is 2.<br>Storage: -20 ~ 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 VDC (-15 ~ 20%) (supplied by the internal bus through the MPU) |
| Power consumption | 1.5 W |
| Insulation voltage | 2500 VDC |
| Weight (approximately, g) | 95g |

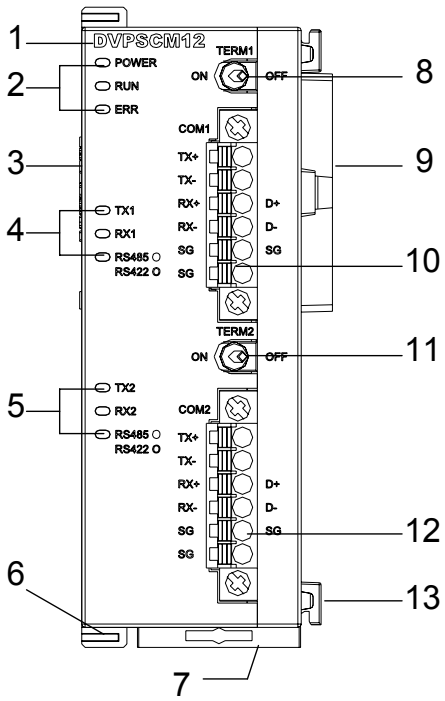
# 2. Product Appearance and Product Profile

## 2.1 Dimensions




Unit: mm [inches]

## 2.2 Product profile

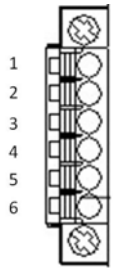



| | |
|---|---|
| 1. Model name | 8. Switch for terminal resistor 1 |
| 2. POWER, RUN, ERR LED indicators | 9. Extension port for the MPU/left-side module |
| 3. Extension port for the left-side module | 10. RS-485/RS-422 communication port 1 |
| 4. TX1, RX1, RS-485/RS-422 LED indicators | 11. Switch 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 SCM module is RUN. | No action is required. |
| | | Off | The status of 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 the factory default. |
| | | Off | No error | No action is required. |
| TX1/TX2 | Orange light | Flash | RS-485/RS-422 is transmitting the data. | No action is required. |
| | | Off | RS-485/RS-422 is not transmitting the data. | No action is required. |
| RX1/RX2 | Orange light | Flash | RS-485/RS-422 is receiving the data. | No action is required. |
| | | Off | RS-485/RS-422 is not receiving the data. | 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

| Diagram of RJ-45 | Terminal no. | RS-485 | RS-422 |
|---|---|---|---|
|   | 1 | | TX+ |
| | 2 | | TX- |
| | 3 | D+ | RX+ |
| | 4 | D- | RX- |
| | 5 | SG | SG |
| | 6 | | SG |

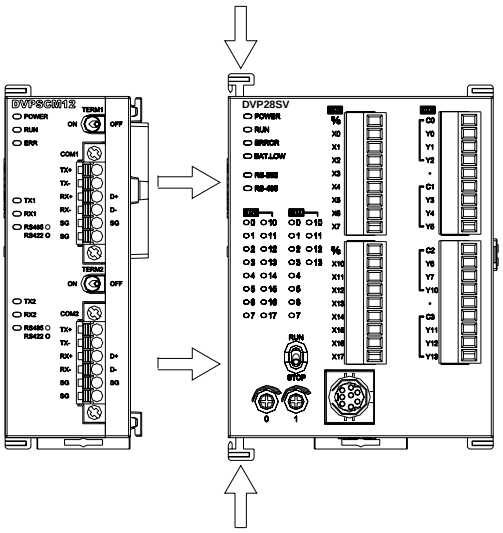
# 3. Installation and Wiring

This chapter introduces how DVPSCM12-SL connects to the MPU.

## 3.1 Installation

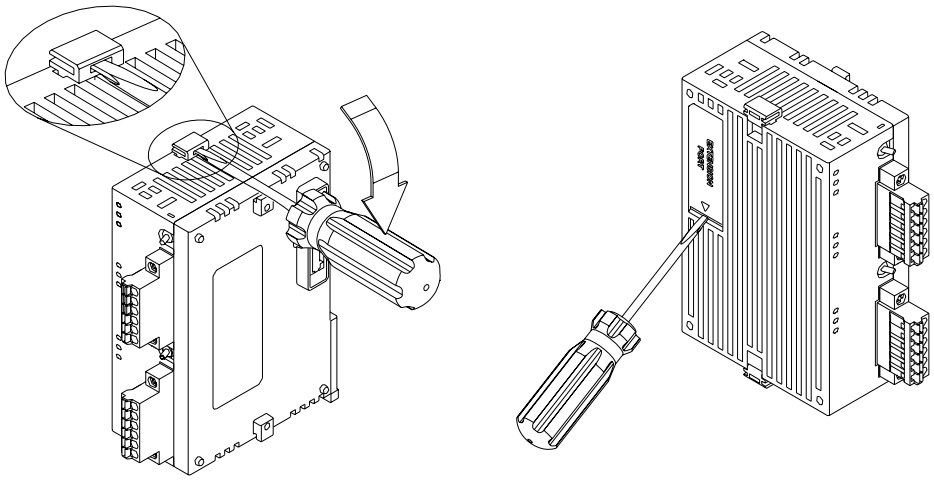
The MPU of the PLC connects to DVPSCM12-SL.

- 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.




DVPSCM12-SL connects to other I/O modules.

- Before DVPSCM12-SL 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.<br>Model code of DVPSCM12-SL=H'4041 |
| 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 sending through COM1 in UD Link | It is used when COM1 UD Link chooses an operand. "Reference address" + "Operand" defines the actual source device for the data sending. |
| 5 | R/W | Reference address of the data receiving through COM1 in UD Link | It is used when the packet chooses an operand in COM1 UD Link. "Reference address" + "Operand" 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 sending through COM2 in UD Link | It is used when the packet chooses an operand in COM2 UD Link. "Reference address" + "Operand" defines the actual source device for the data receiving. |
| 9 | R/W | Reference address of the data receiving through COM2 in UD Link | It is used when COM2 UD Link chooses an operand. "Reference address" + "Operand" defines the actual source device for the data sending. |
| 10 | R | Module status | RUN or STOP |
| 11 ~ 19 | R | Error Flag | The flag for an error in DVPSCM12-SL |
| 20 ~ 27 | R | UD Link status | The execution status of UD Link |
| 28 ~ 29 | Reserved | | |
| 30 | R/W | Trigger UD Link Sequence. | 0: Not trigger, 1~254: number of times for triggering, 255: Always trigger |
| 31 | R/W | Trigger the data exchange through COM1 to read bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 32 | R/W | Trigger the data exchange through COM2 to read bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 33 | R/W | Trigger the data exchange through COM1 to write bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 34 | R/W | Trigger the data exchange through COM2 to write bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 35 ~ 36 | R/W | Check "exchange data through COM1 to read bits" column. | Bit = 0: Disable the function of reading bits through COM1.<br>Bit = 1: Enable the function of reading bits through COM1. |
| 37 ~ 38 | R/W | Check "exchange data through COM1 to read words" column. | Bit = 0: Disable the function of reading words through COM1.<br>Bit = 1: Enable the function of reading words through COM1. |

| CR# | Attribute | Name of the register | Description |
|---|---|---|---|
| 39 ~ 40 | R/W | Check “exchange data through COM2 to read bits” column. | Bit = 0: Disable the function of reading bits through COM2.<br>Bit = 1: Enable the function of reading bits through COM2. |
| 41 ~ 42 | R/W | Check “exchange data through COM2 to read words” column. | Bit = 0: Disable the function of reading words through COM2.<br>Bit = 1: Enable the function of reading words through COM1. |
| 43 ~ 44 | R/W | Check “exchange the data through COM1 to write bits” column. | Bit = 0: Disable the function of writing bits through COM1.<br>Bit = 1: Enable the function of writing bits through COM1. |
| 45 ~ 46 | R/W | Check “exchange the data through COM1 to write words” column. | Bit = 0: Disable the function of writing words through COM1.<br>Bit = 1: Enable the function of writing words through COM1. |
| 47 ~ 48 | R/W | Check “exchange the data through COM2 to write bits” column. | Bit = 0：Disable the function of writing bits through COM2.<br>Bit = 1：Enable the function of writing bits through COM2. |
| 49 ~ 50 | R/W | Check “exchange the data through COM2 to write words” column. | Bit = 0: Disable the function of writing words through COM2.<br>Bit = 1: Enable the function of writing words through COM2. |
| 51 ~ 115 | Reserved | | |
| 116 | R/W | Send the Modbus command. | 1: Enable the sending<br>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 | Set the length of the Modbus command. |
| 120~249 | R/W | Content 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. Code of DVPSCM-SL=H’4041
2. The code of the model 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 sending through COM1 in UD Link

[Description]

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

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

If “Operand” is chosen in the packet editor, “Reference address” + “Operand” defines the actual source device for the data sending.

Example: Enter 1 in CR#4 → D1,

Choose “Operand” in the packet editor, and enter 10 for the offset and 2 for the length → (R (Op [10], 2)).

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

This control register is used when “operand” is set in the packet editor.

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

[Description]

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

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

If “Operand” is chosen in the packet editor, “Reference address” + “Operand” 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 sending through COM2 in UD Link

[Description]

Please refer to the description of CR#4.

**CR#9**: Reference address of the data receiving 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 DVPSCM12-SL.

**CR#11~19**: Error flag

[Description]

With regard to the error flag in DVPSCM12-SL, please refer to chapter 8.

**CR#20~27**：UD Link status

[Description]

The execution status of UD Link

**CR#30**: Trigger UD Link SEQUENCE.

[Description]

High byte: COM1; Low byte: COM2

Enter directly the number of times for the execution.

0: Not trigger; 1~254: Number of times for triggering; 255 (H' FF): Always trigger

**CR#31**: Trigger the data exchange through COM1 to read bits or words.

[Description]

High byte: COM1 Bit; Low byte: COM1 Word

0: Not execute; 1: Execute once; 2: Always execute

| | | COM1 Word | | |
|---|---|---|---|---|
| | | Not execute | Execute once | Always execute |
| COM1 Bit | Not trigger | H' 0000 | H' 0001 | H' 0002 |
| | Trigger once | H' 0100 | H' 0101 | H' 0102 |
| | Always trigger | H' 0200 | H' 0201 | H' 0202 |

**CR#32**: Trigger the data exchange through COM2 to read bits or words.

[Description]

High byte: COM2 Bit; Low byte: COM2 Word

0: Not execute; 1: Execute once; 2: Always execute

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

**CR#33**: Trigger the data exchange through COM1 to write bits or words.

[Description]

High byte: COM1 Bit; Low byte: COM1 Word

0: Not execute; 1: Execute once; 2: Always execute

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

**CR#34**: Trigger the data exchange through COM2 to write bits or words.

[Description]

High byte: COM2 Bit; Low byte: COM2 Word

0: not execute; 1: Execute once; 2: Always execute

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

**CR#35 ~ 36**: Check “exchange data through COM1 to read bits” column.

[Description]

Check “exchange the data through COM1 to read bits” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#37 ~38**: Check “exchange data through COM1 to read words” column.

[Description]

Check “exchange the data through COM1 to read words” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#39 ~ 40**: Check “exchange data through COM2 to read bits” column.

[Description]

Check “exchange the data through COM2 to read bits” column. DVPSCM12-SL 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**: Check “exchange data through COM2 to read words” column.

[Description]

Check “exchange the data through COM2 to read words” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#43 ~ 44**: Check “exchange data through COM1 to read bits” column.

[Description]

Check “exchange the data through COM1 to write bits” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#45 ~ 46**: Check “exchange data through COM1 to read words” column.

[Description]

Check “exchange the data through COM1 to write words” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#47 ~ 48**: Check “exchange data through COM2 to read bits” column.

[Description]

Check “exchange the data through COM2 to write bits” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#49 ~ 50**: Check “exchange data through COM2 to read words” column.

[Description]

Check “exchange the data through COM2 to write words” column. DVPSCM12-SL 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: Enable the function; 1: Disable the function

**CR#116**: Send the Modbus command.

[Description]

DVPSCM-SL sends the Modbus command.

1: Enable the sending.

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

**CR#117**: Processing status of Modbus command

[Description]

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

**CR#118**: Destination of Modbus command

[Description]

Designate the sending port of DVPSCM12-SL. 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**：Content of the Modbus command

[Description]

The content of the Modbus command which is sent (in a hexadecimal value)

## 4.3 Right-side module numbers

After DVPSCM12-SL is installed, the related functions of the I/O module are controlled by the PLC program. The PLC provides two instructions 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.

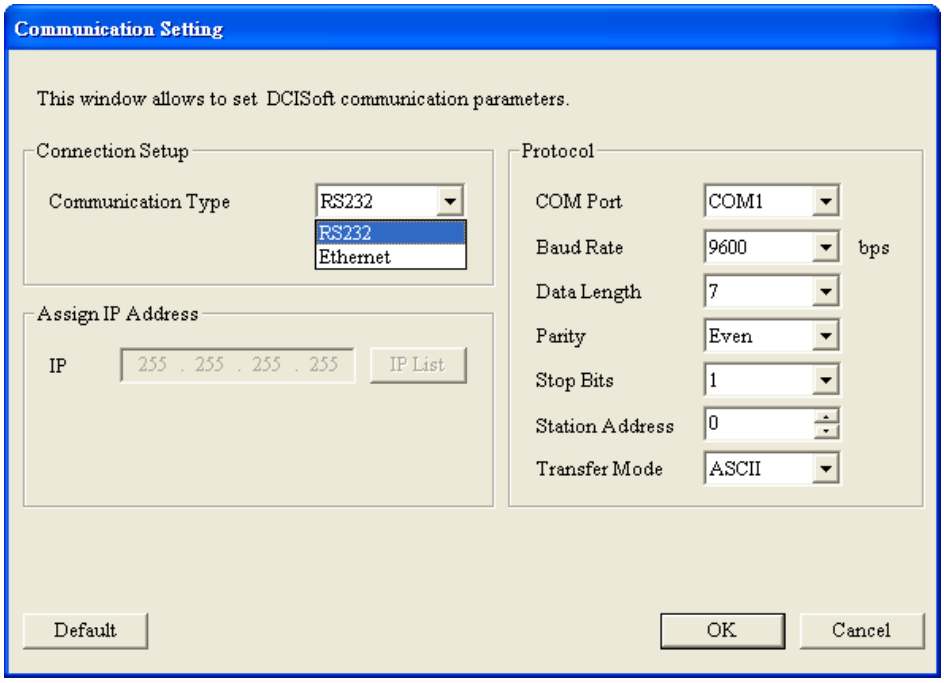
**MEMO**

## 5. Rapid Start

This chapter introduces how to execute Modbus RS-485/RS-422 communication through the communications ports on DVPSCM-SL module.

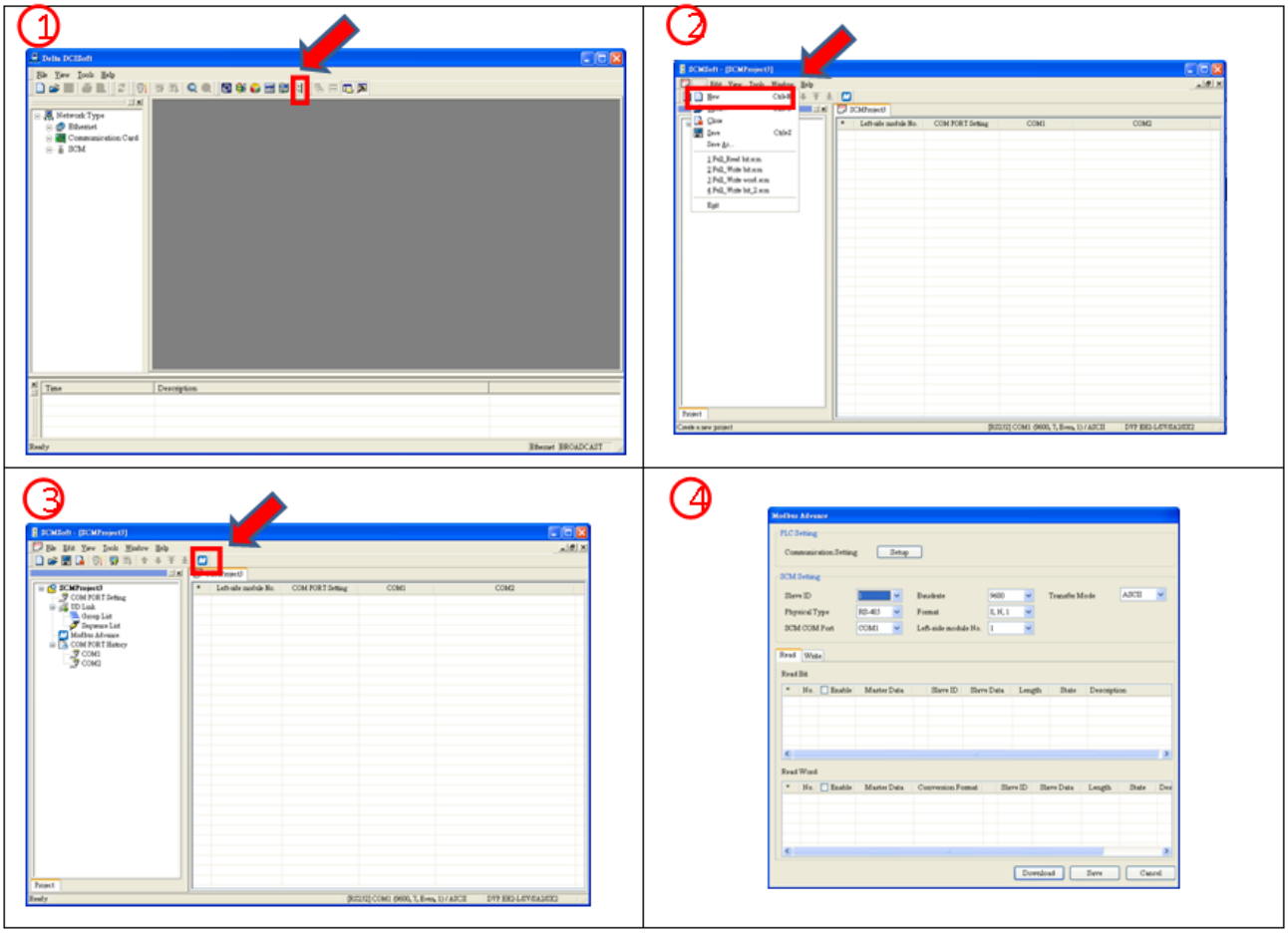
**【Communication setting】**

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 is used with DVPSCM-SL module, the user can choose “Ethernet” in “Transfer Mode” column to upload/download the program.




**【Open 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.




**【Set “Modbus Advance”】**

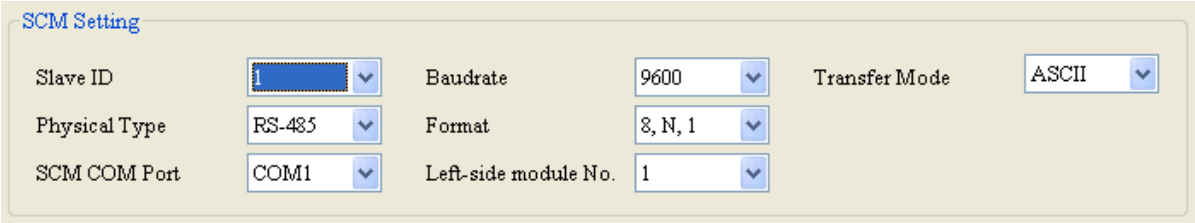
In order to expedite the communication using Modbus, SCMSoft provides “Modbus Advance Wizard”. The user only needs to designate the registers for the data sending and the data receiving, or the absolute positions. The settings will be downloaded to SCM module through the communication port chosen by 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.

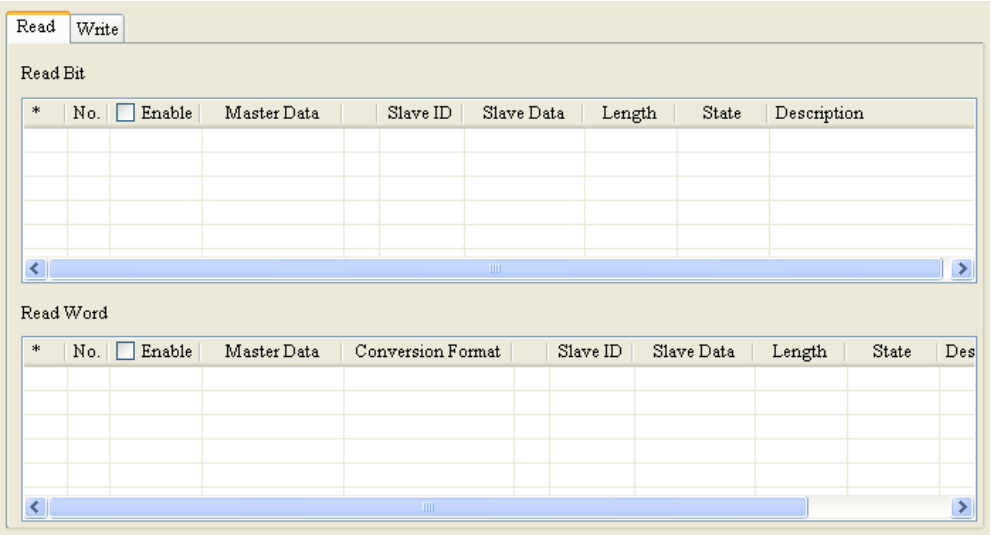
(2) SCM Setting

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

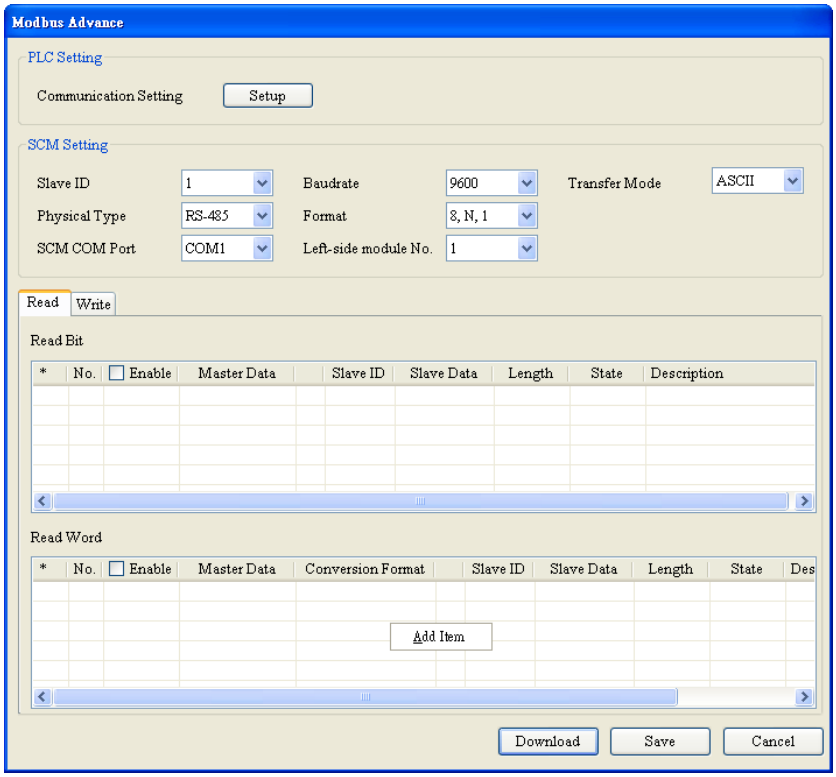



(3) Modbus Advance－Read/Write

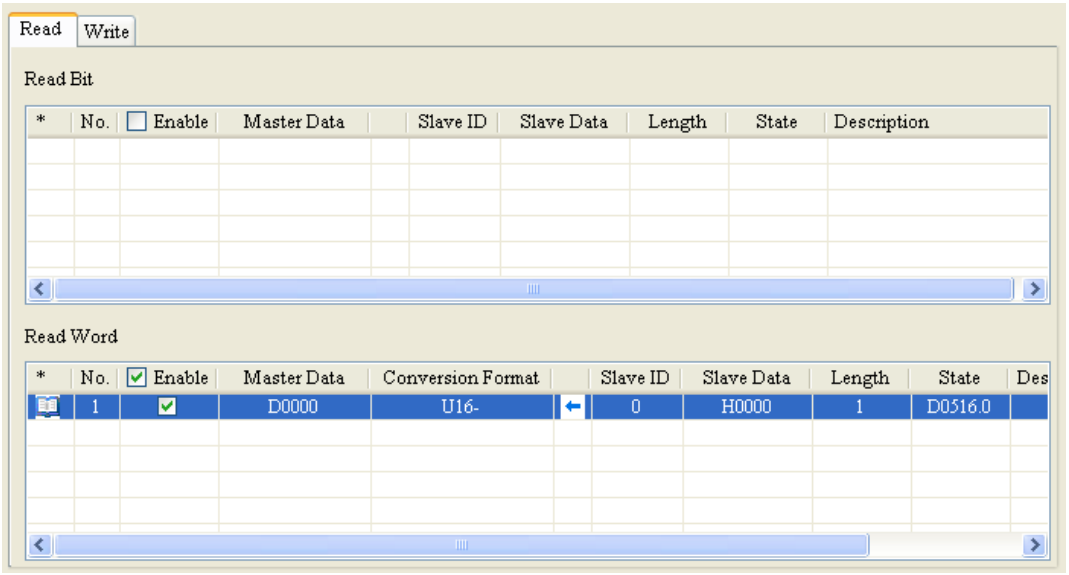
Set “Read Bit”/“Read Word” and “Write Bit”/ “Write Word”.

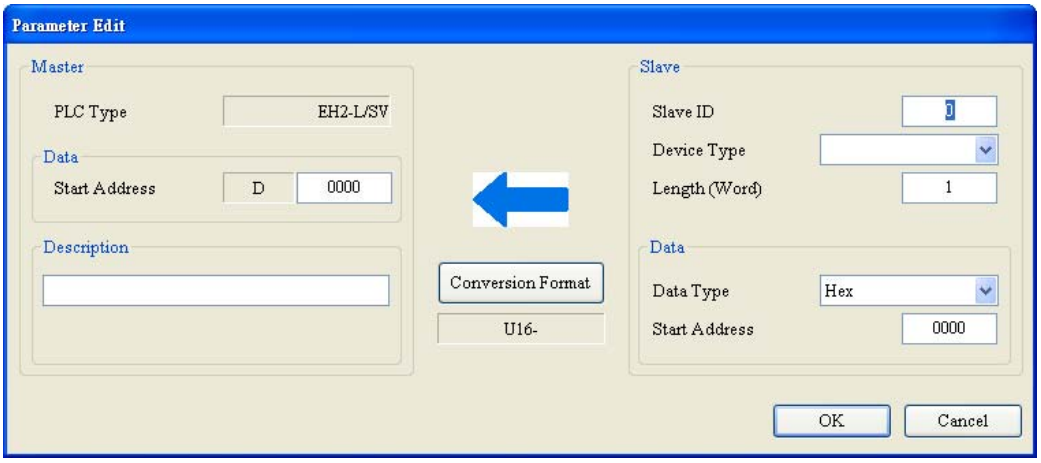



Press the right key of the mouse, and click “Add Item” to increase bits and words. The bits are listed in the upper column, and the words are listed in the lower column.




After clicking twice the added item, the user can edit the parameter.



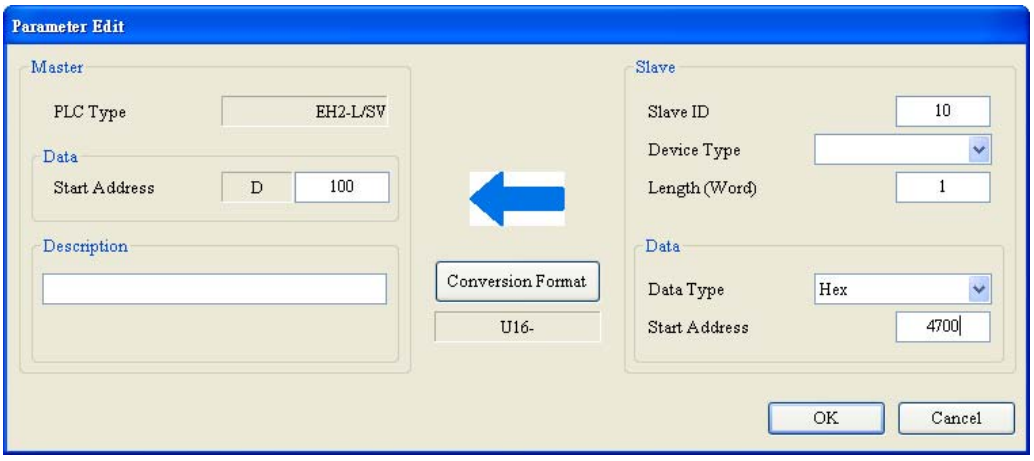

Master:

| | |
|---|---|
| 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 D 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:

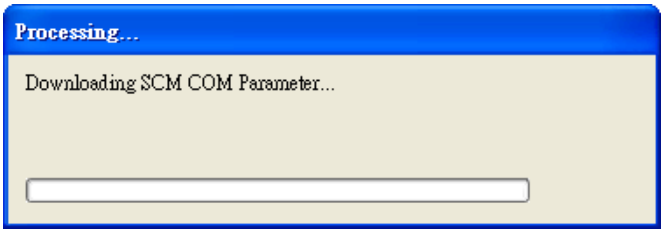
| | |
|---|---|
| Slave ID: | The number of the salve device from which the data is read |
| Device Type: | The user can choose the Delta PLC type. If the PLC used is not a Delta PLC, please leave the column blank. |
| Length (bit): | It indicates the length of the data being read. The maximum length 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 D register. |
| Start Address | The start address of the data |

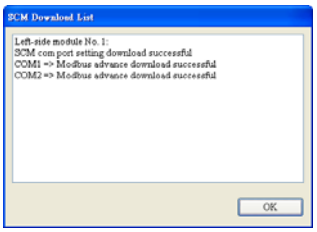
If the absolute position of the present value of the Delta DTA temperature controller is the hexadecimal value, 4700 (H’4700), and the station number is10, the present value can be read and stored in D100 in the MPU of the PLC through COM1. The settings are as follows:




**【Download】**

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



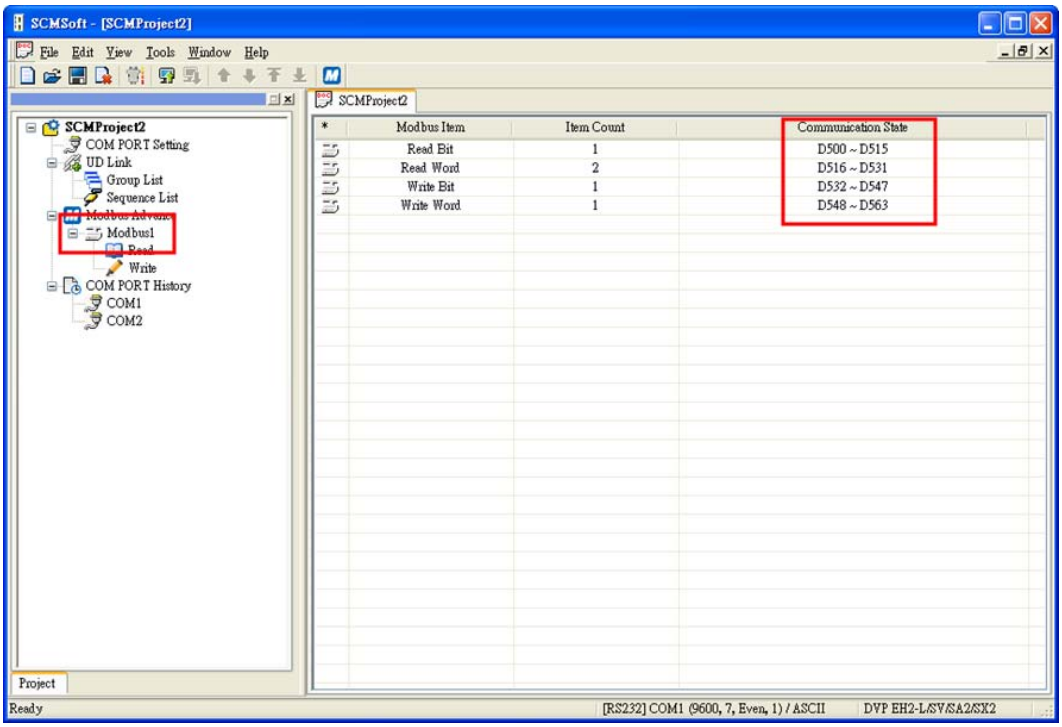

**【Communication state】**

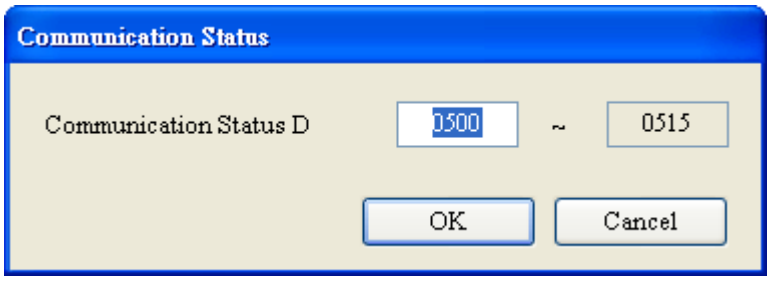
SCM module provides the communication state of Modbus Advance. There are four sections ─ Read Bit, Read Word, Write Bit, and Write Word. The execution status in each line is stored in the bits in the register of each section. If D100 is entered into No.1, the execution status of the data exchange in 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.



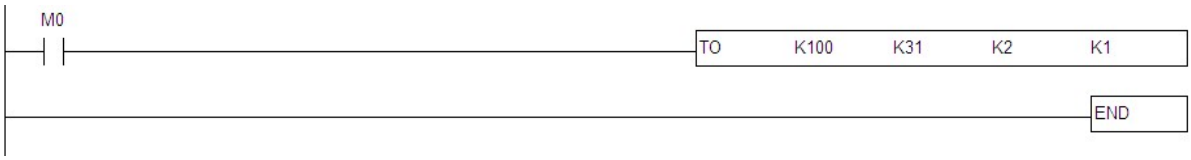




**【Enable】**

Control the data exchange through “TO instruction” 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 | Trigger the data exchange through COM1 to read bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 32 | R/W | Trigger the data exchange through COM2 to read bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 33 | R/W | Trigger the data exchange through COM1 to write bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 34 | R/W | Trigger the data exchange through COM2 to write bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |

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

| 裝置名稱 | 註解 | 狀態 | 設定值 | 現在值(16bits) | 現在值(32bits) | 浮點數 | 型態 | T/C裝置設定狀態 |
|---|---|---|---|---|---|---|---|---|
| D100 | | | | K286 | K286 | F4.007E-43 | 有號10進制 | |
| D0 | | | | K1 | K1 | F1.401E-45 | 有號10進制 | |

**MEMO**

# 6. Introduction of SCMSoft

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

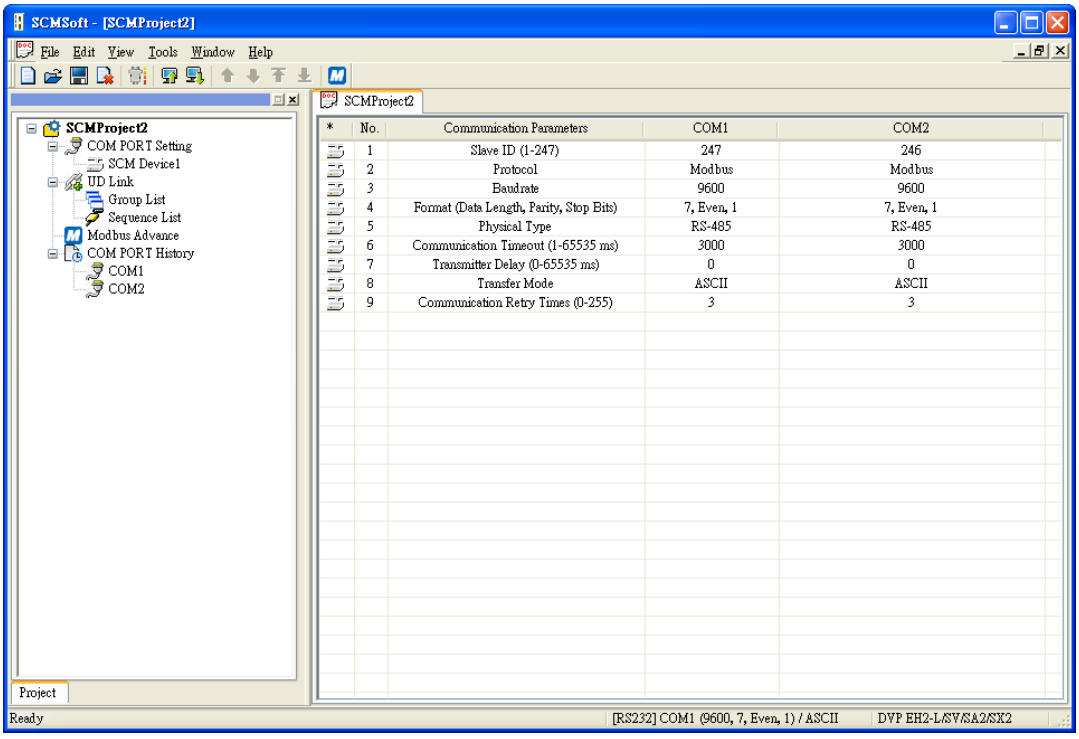
## 6.1 SCM project

Through establishing a SCM project, SCM module makes the execution plan for COM1 and COM2. A 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 SCM module execute (Ch 6.2). |
| UD Link: | It can connect to RS-485 communication, and the user can define the content of the RS-485 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 portion on SCM module (Ch 6.5). |

## 6.2 COM PORT setting

Set the serial communication format:



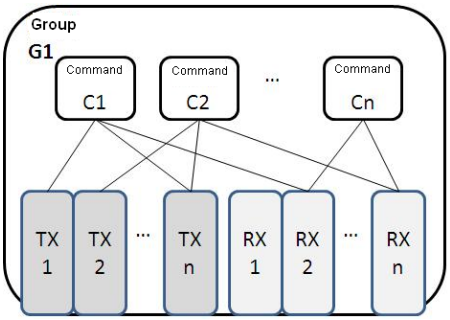

| | |
|---|---|
| Slave ID: | The user can set the slave IDs of COM1 and COM2. The superior device connects to SCM module through the slave ID. The default slave ID of COM1 is 247, and that of COM2 is 246. |
| Protocol: | When the standard Modbus is used, thee user can choose Modbus. When the user-defined format of RS-485 is used, the user can choose UD Link. |

| | |
|---|---|
| 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. |

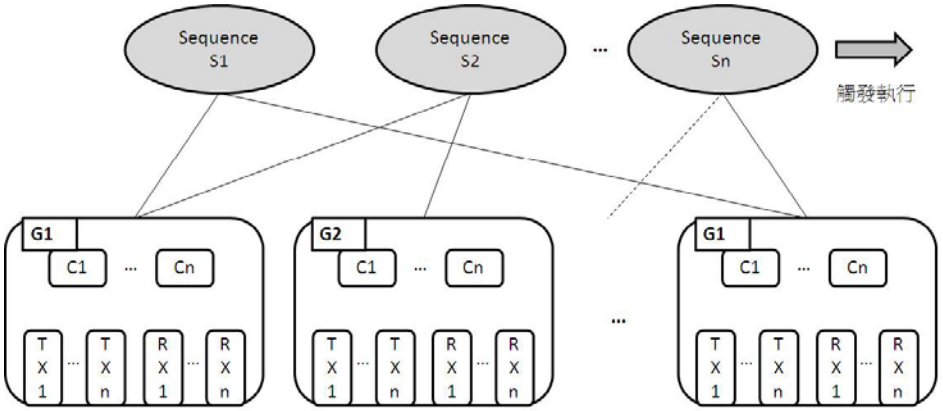
## 6.3 UD Link (user-defined link)

UD Link provides non-Modbus RS-485 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

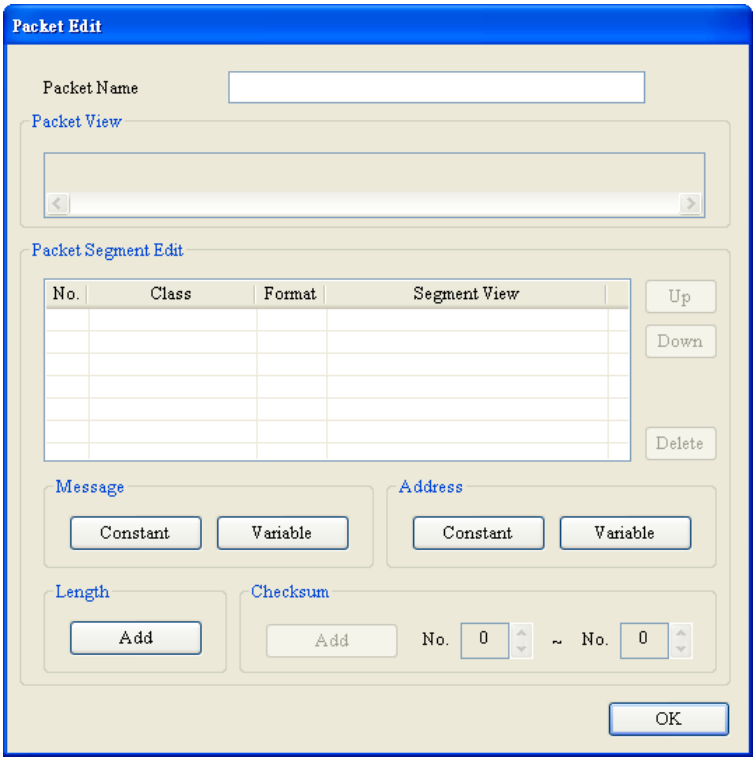



First of all, establish the transmission instructions (TXs) and the reception instructions (RXs) in the group. Then, set the execution sequence and the number of times for TXs and RXs through the commands.

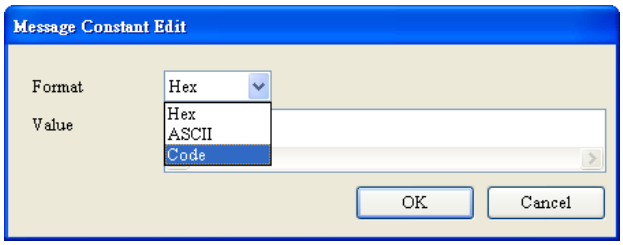
Finally, trigger and execute the instructions as a group. In addition, if various groups of group packets are required in a large system, the user can create the groups in the sequences, and set the execution sequence.

### 6.3.1 TX Packets and RX Packets

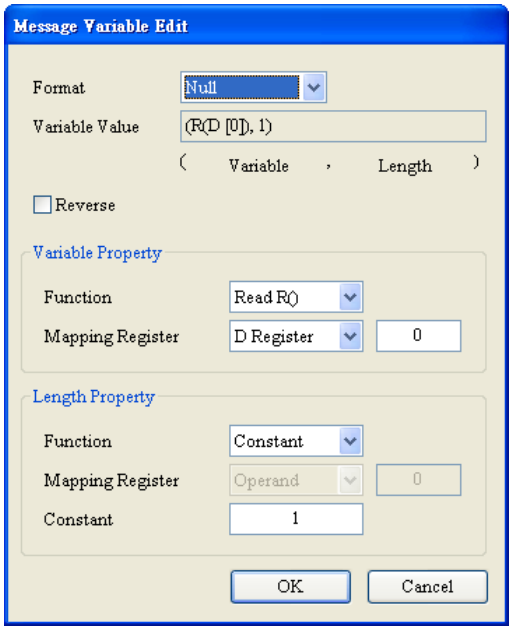
The user can create various TX packets and RX packets in a group. The contents of TX packets and RX packets may include several messages, one address, one length, and one checksum.




- 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.




- 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.




- Variable: The data is a variable whose mapping register can be the internal register in DVPSCM12-SL 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 DVPSCM12-SL or the register in the MPU of the PLC. The internal registers in DVPSCM12-SL include I1, I2, O1, and O2. The registers in the PLC include the D registers and the operands.

| Register | Definition | Register | Definition |
|---|---|---|---|
| D | Internal D register in the PLC | Operand | 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.

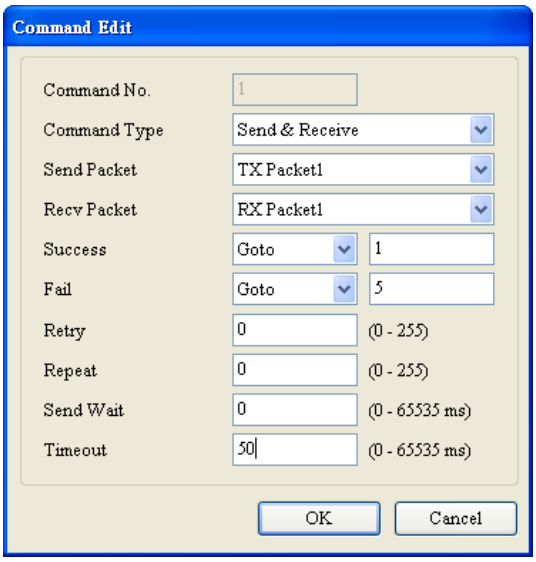
  Format: The user can choose the format of the checksum segment.

  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.: 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 “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.

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

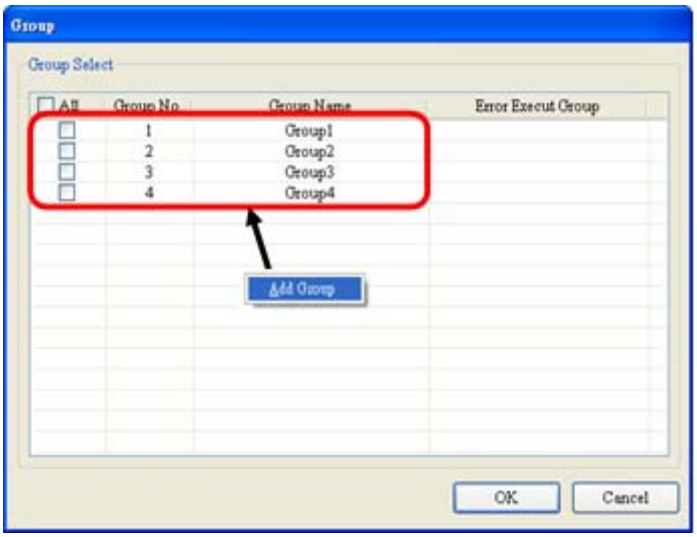
Repeat: It means the number of times the sending of a command has been repeated after the command has been executed successfully.

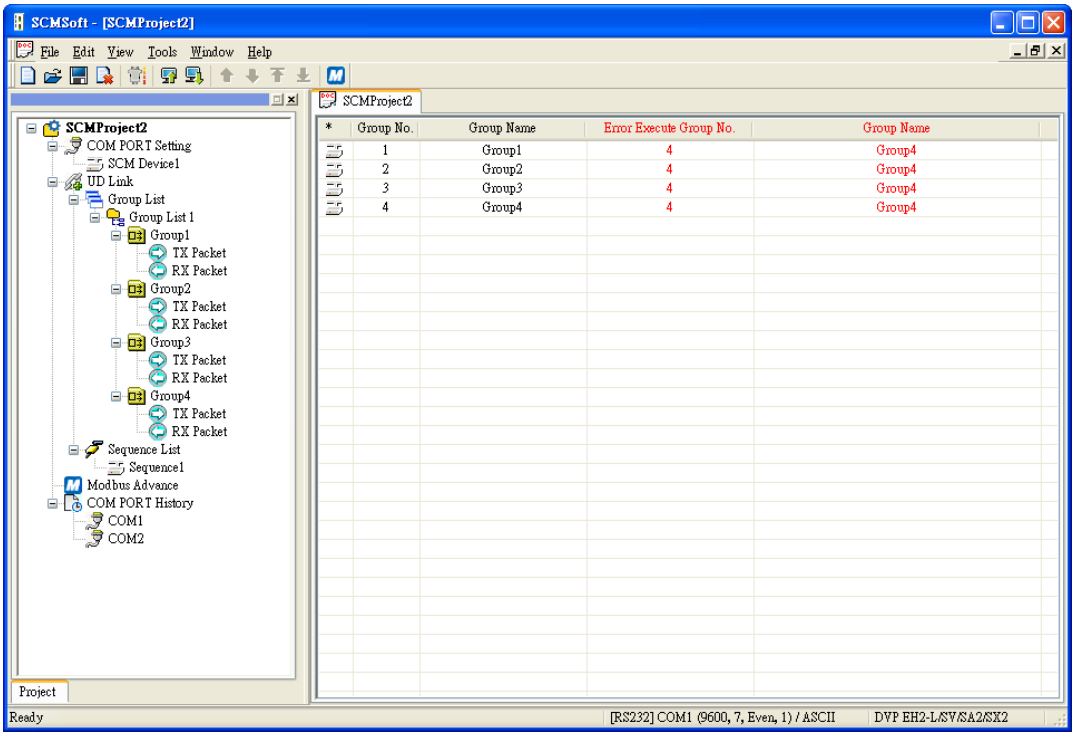
Send Wait: The default time interval between the instructions is 0 ms, that is, the next instruction is transmitted immediately after the reply is received.

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.




## 6.4 Modbus Advance

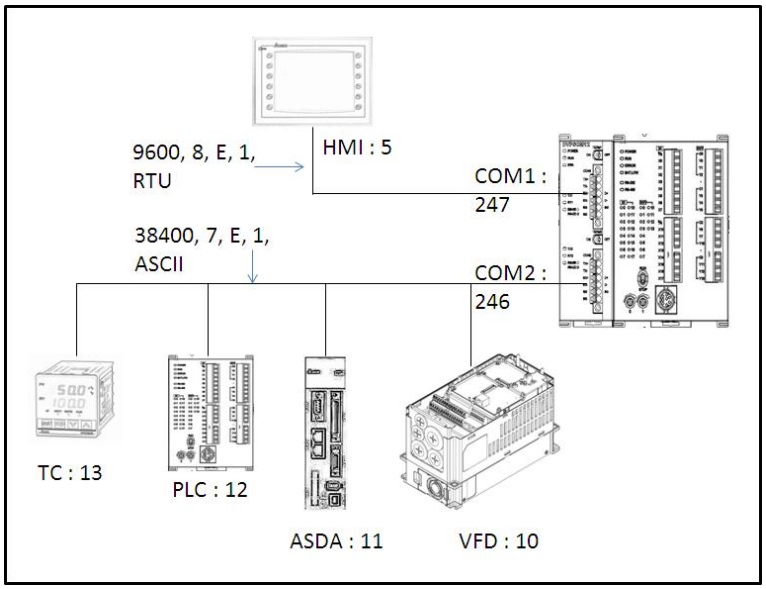
Please refer to Chapter 5 for more related introduction.

**MEMO**

# 7. Application

## 7.1 Modbus

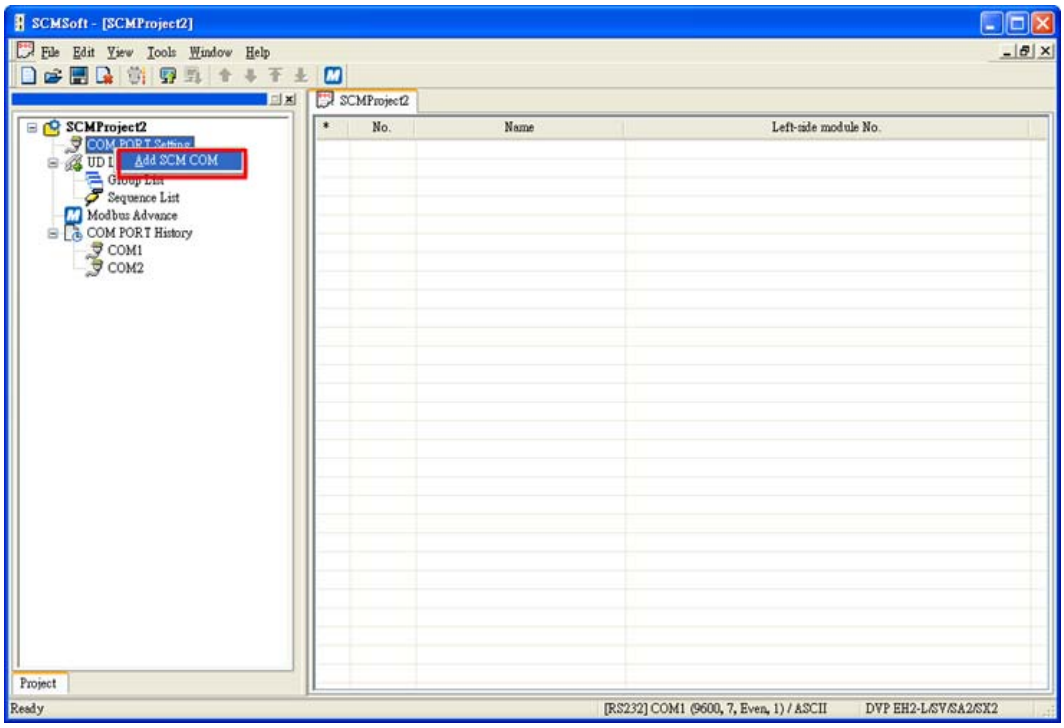
This chapter introduces how DVPSCM12-SL connects to other Delta industrial products such as the human-machine interfaces, the text panels, the PLCs, the motor drives, and the servo motors through the standard Modbus. The connection diagram is as below:

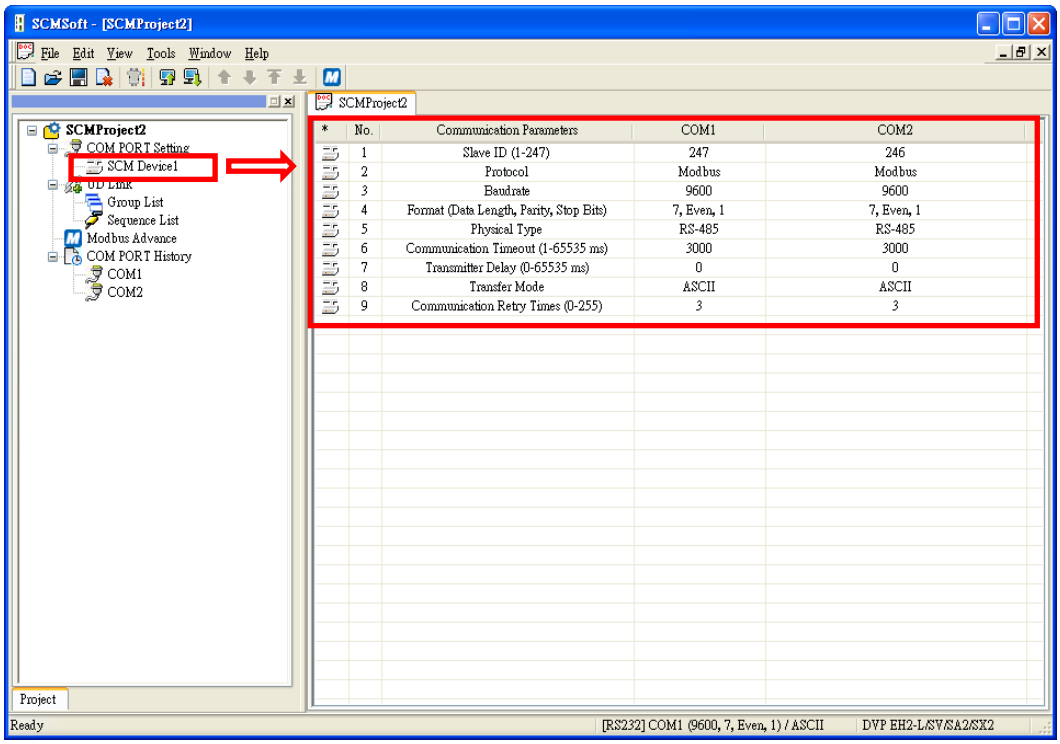



| Product | Station number | Communication protocol | Address from which the data is read | Register in the MPU | Address into which the data is written | Register in the MPU |
|---|---|---|---|---|---|---|
| HMI | 5 | 9600, RTU, 8, E, 1 | - | | - | |
| VFD | 10 | 38400, ASCII, 7, E, 1 | 2103H | D100 | 2000H<br>2001H | D150~D151 |
| ASDA | 11 | 38400, ASCII, 7, E, 1 | 0101H<br>020AH | D200,D201 | 0101H<br>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 The connection between the Modbus slave and the Delta product

(1) For DVPSCM12-SL as the Modbus slave, the user only has to set the parameters such as the station number and the baudrate to allow the connection with the master.
Open SCMSoft → 『New Project』 → COM PORT setting: 『Add SCM COM』 → Set the communication parameters.




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

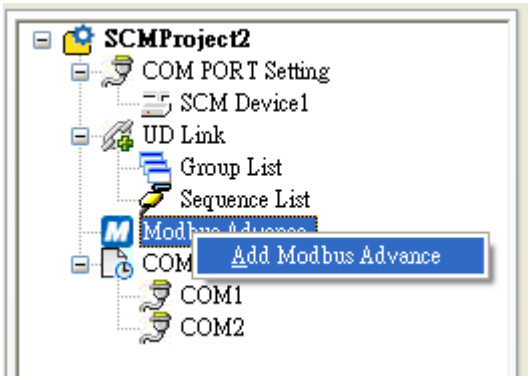
| * | No. | Communication Parameters | COM1 |
|---|---|---|---|
| | 1 | Slave ID (1-247) | 247 |
| | 2 | Protocol | Modbus |
| | 3 | Baudrate | 9600 |
| | 4 | Format (Data Length, Parity, Stop Bits) | 8, Even, 1 |
| | 5 | Physical Type | RS-485 |
| | 6 | Communication Timeout (1-65535 ms) | 3000 |
| | 7 | Transmitter Delay (0-65535 ms) | 0 |
| | 8 | Transfer Mode | RTU |
| | 9 | Communication Retry Times (0-255) | 3 |

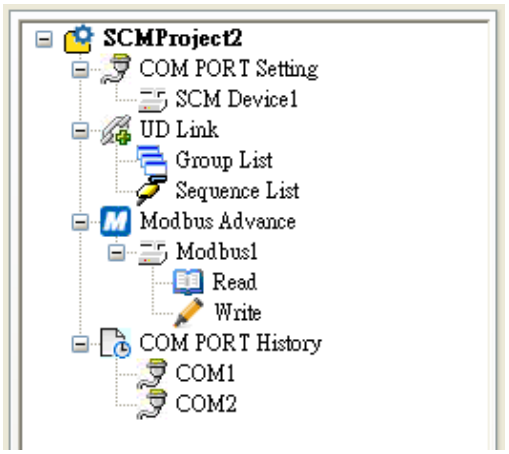
### 7.1.2 The connection between the Modbus master and the Delta product

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

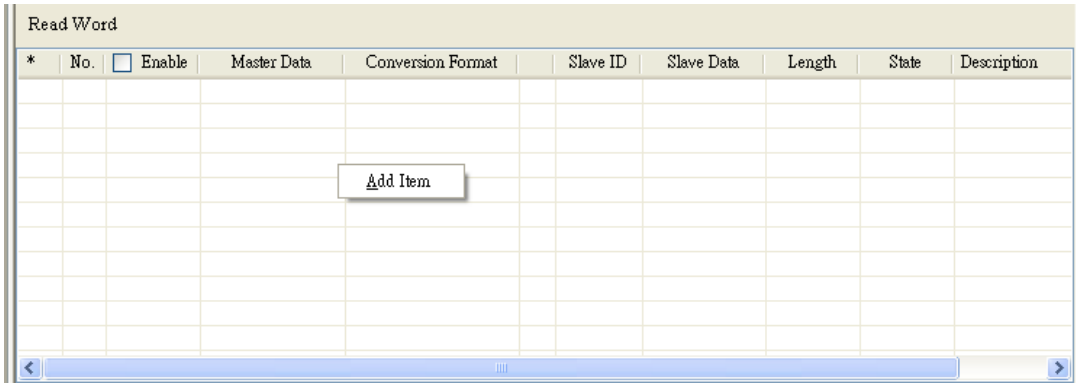
| * | No. | Communication Parameters | COM1 | COM2 |
|---|---|---|---|---|
| | 1 | Slave ID (1-247) | 247 | 246 |
| | 2 | Protocol | Modbus | Modbus |
| | 3 | Baudrate | 9600 | 9600 |
| | 4 | Format (Data Length, Parity, Stop Bits) | 8, Even, 1 | 7, Even, 1 |
| | 5 | Physical Type | RS-485 | RS-485 |
| | 6 | Communication Timeout (1-65535 ms) | 3000 | 3000 |
| | 7 | Transmitter Delay (0-65535 ms) | 0 | 0 |
| | 8 | Transfer Mode | RTU | ASCII |
| | 9 | Communication Retry Times (0-255) | 3 | 3 |

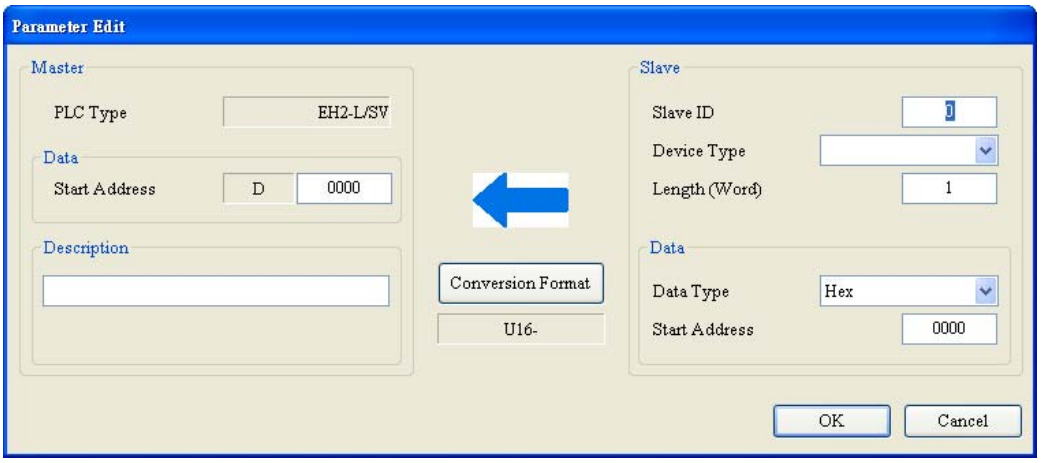
(2) Add Modbus Advance



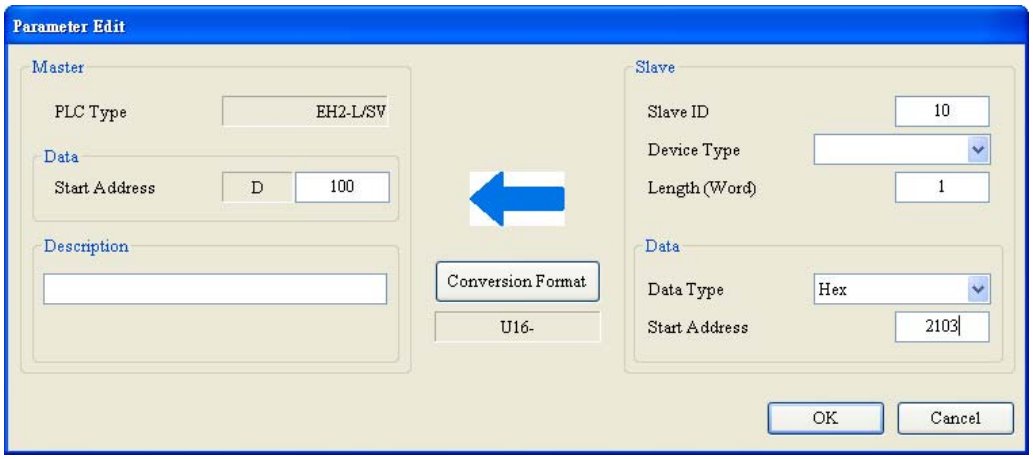



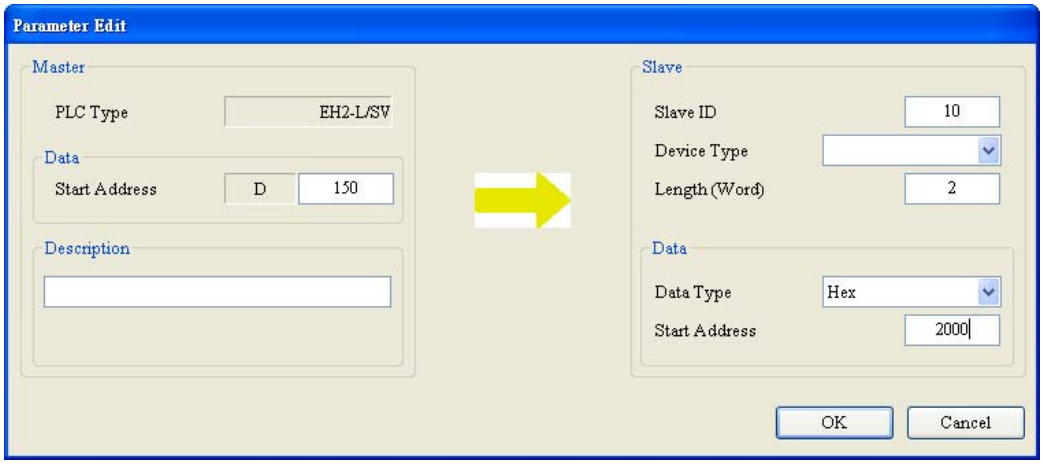

(3) Set the data exchange in the slave: Add Item → Click the added item twice to set the reading/writing information in the slave.



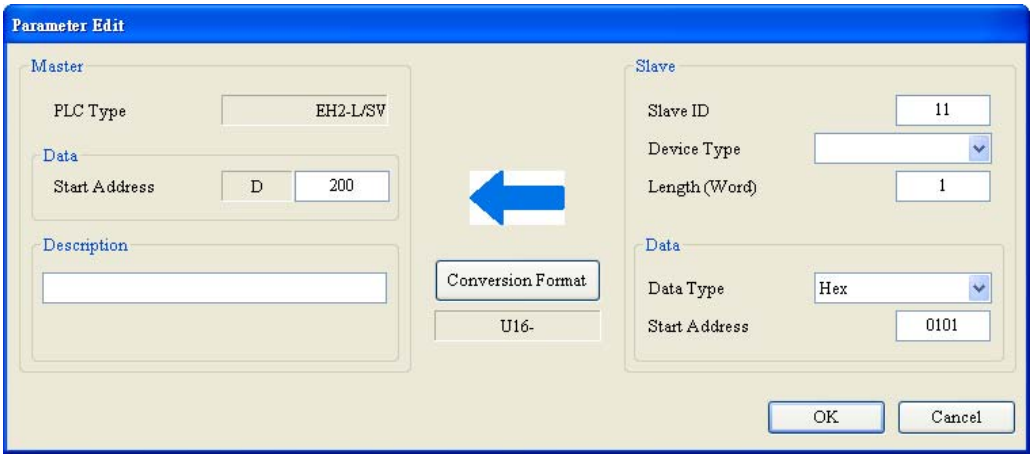


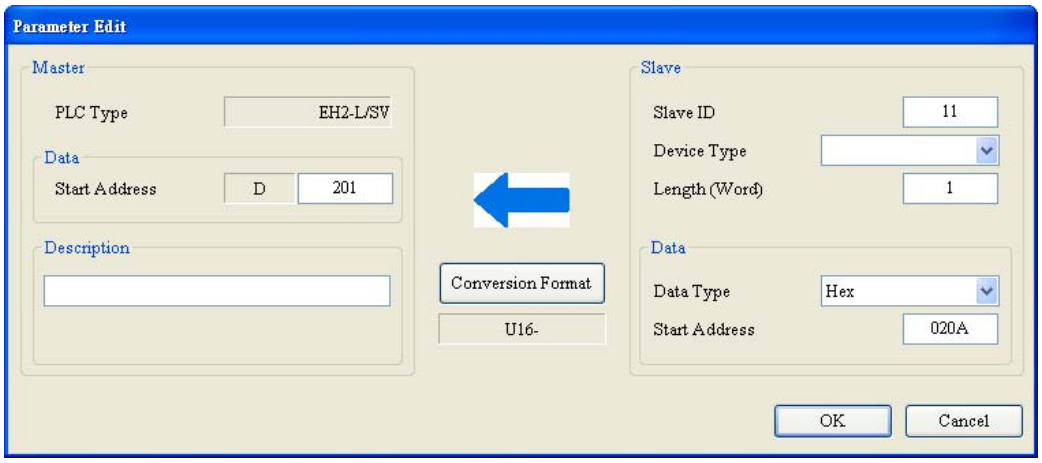

VFD (D100←2103H), (D150, D151→H2000, H2001)



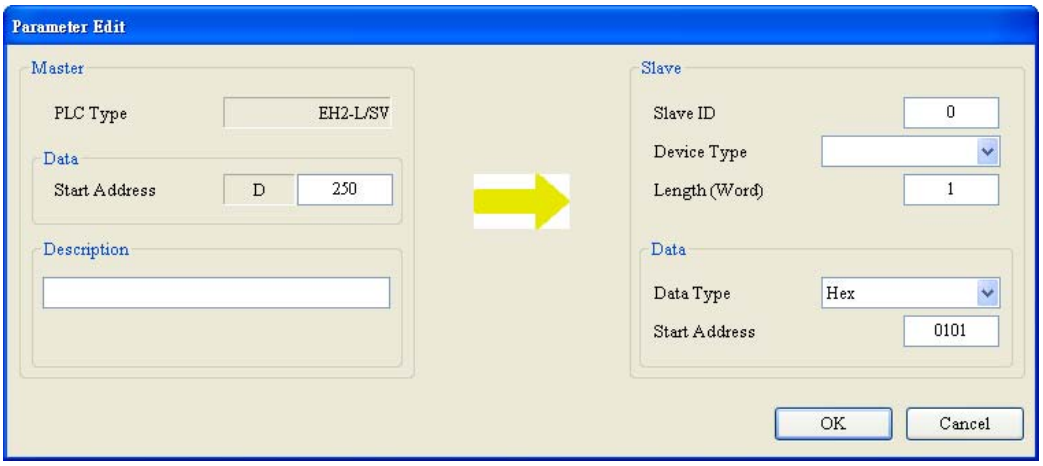



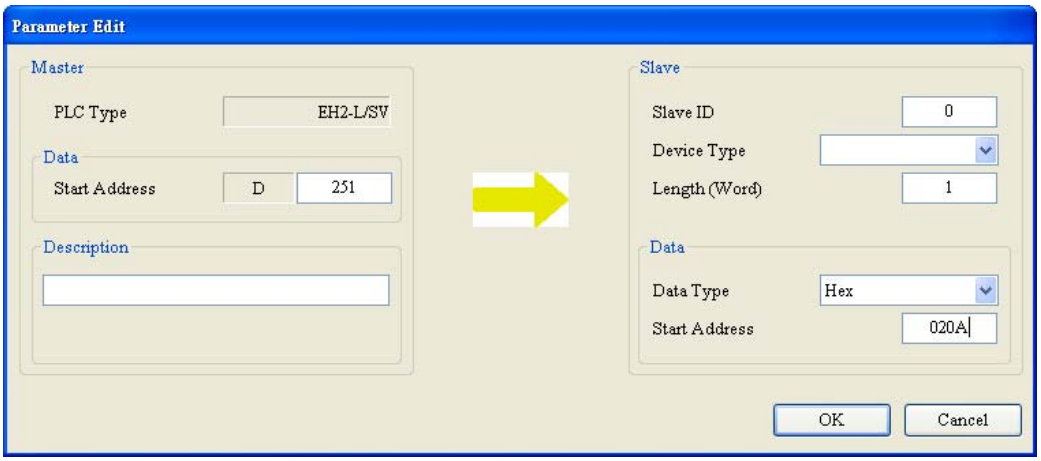
ASDA (D200←0101H, D201←020AH)



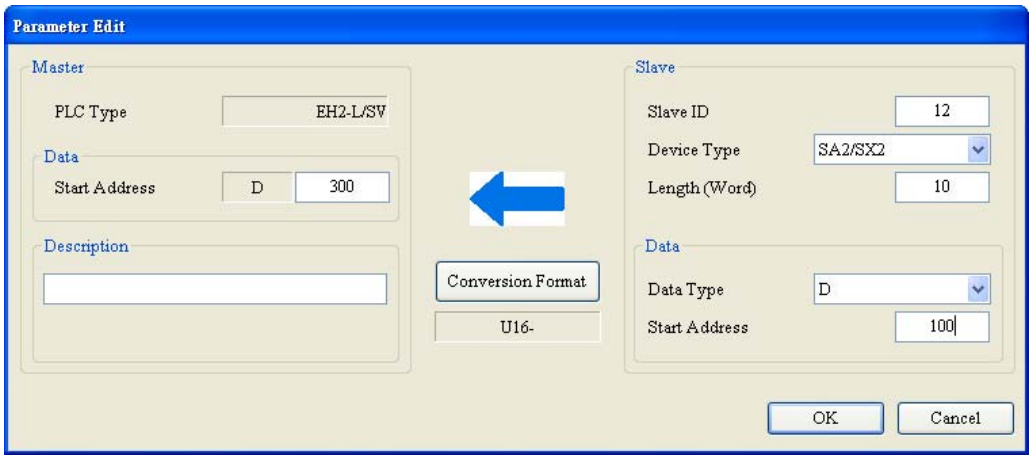



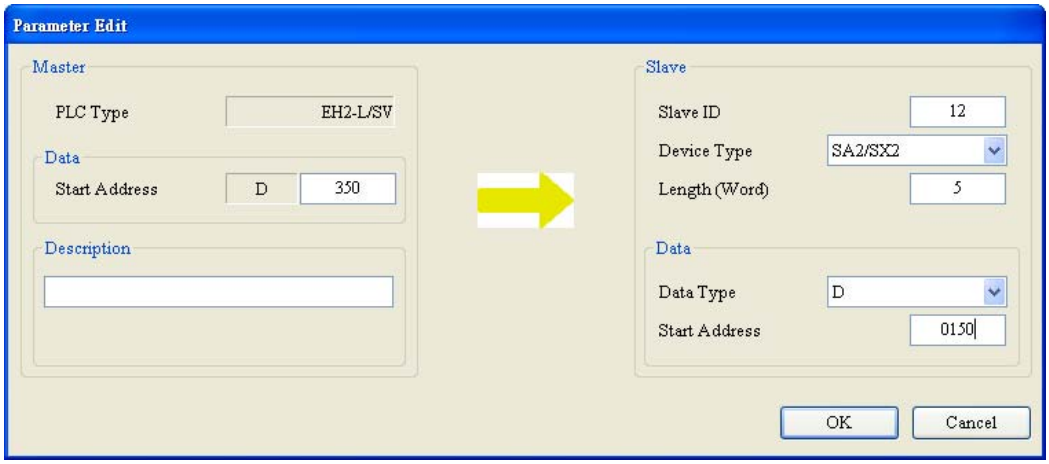

(D250→0101H, D251→020AH)



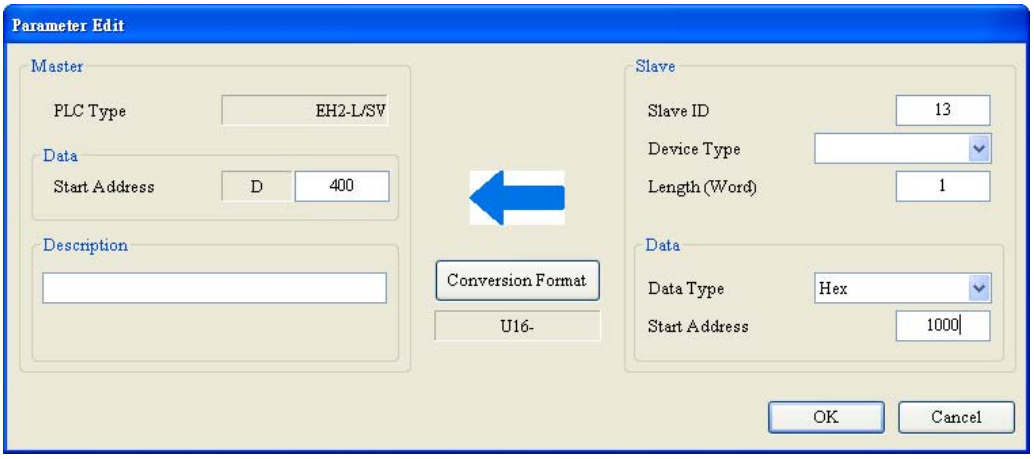


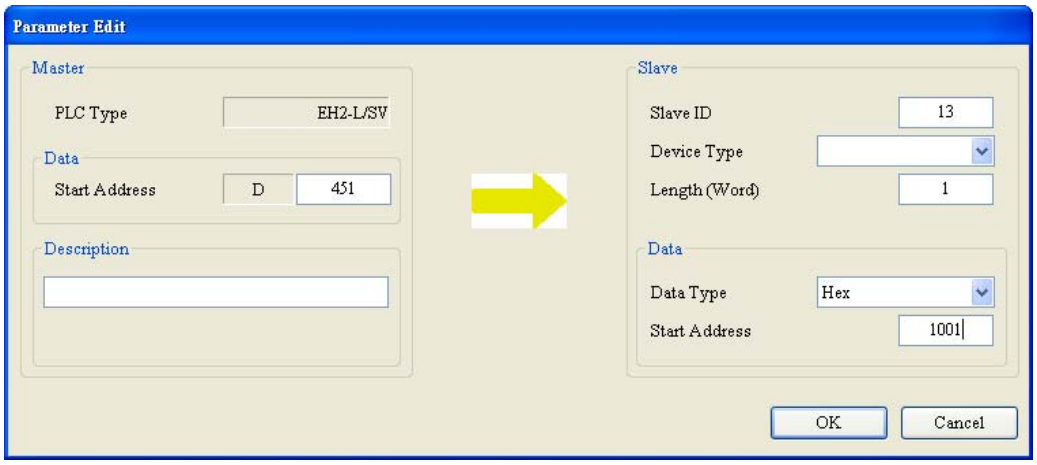

PLC (master D300~D309←slave D100~D109), (master D350~D354→slave D150~D154)



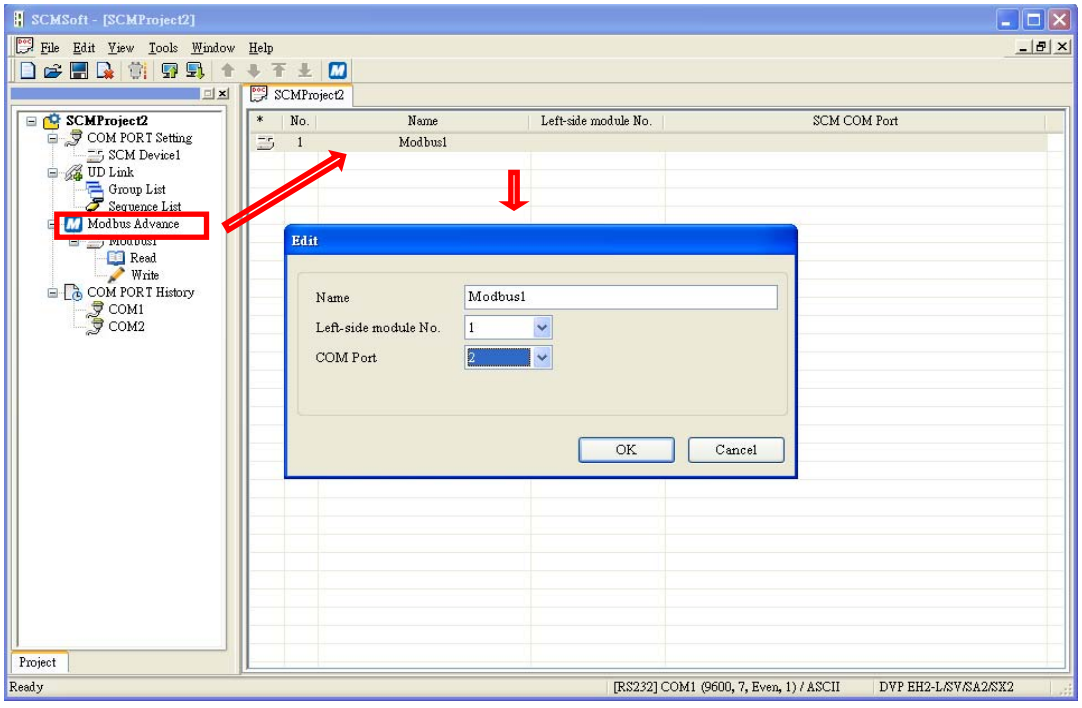



TC (D400←1000H), (D451→1001H)



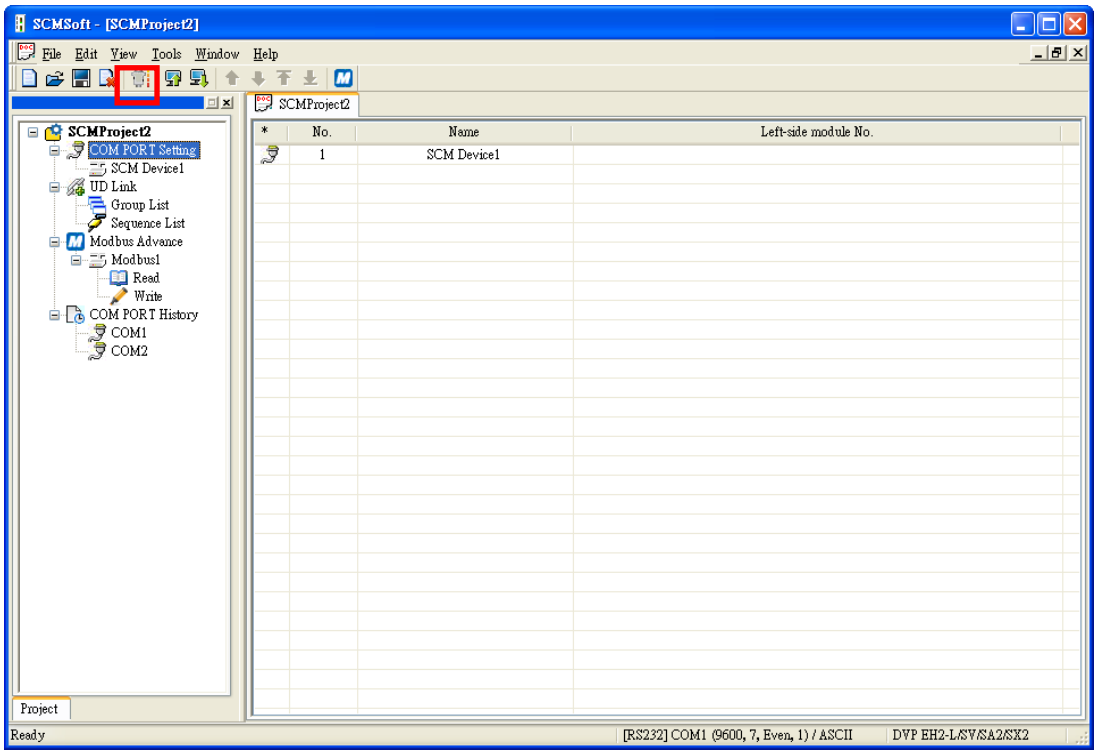



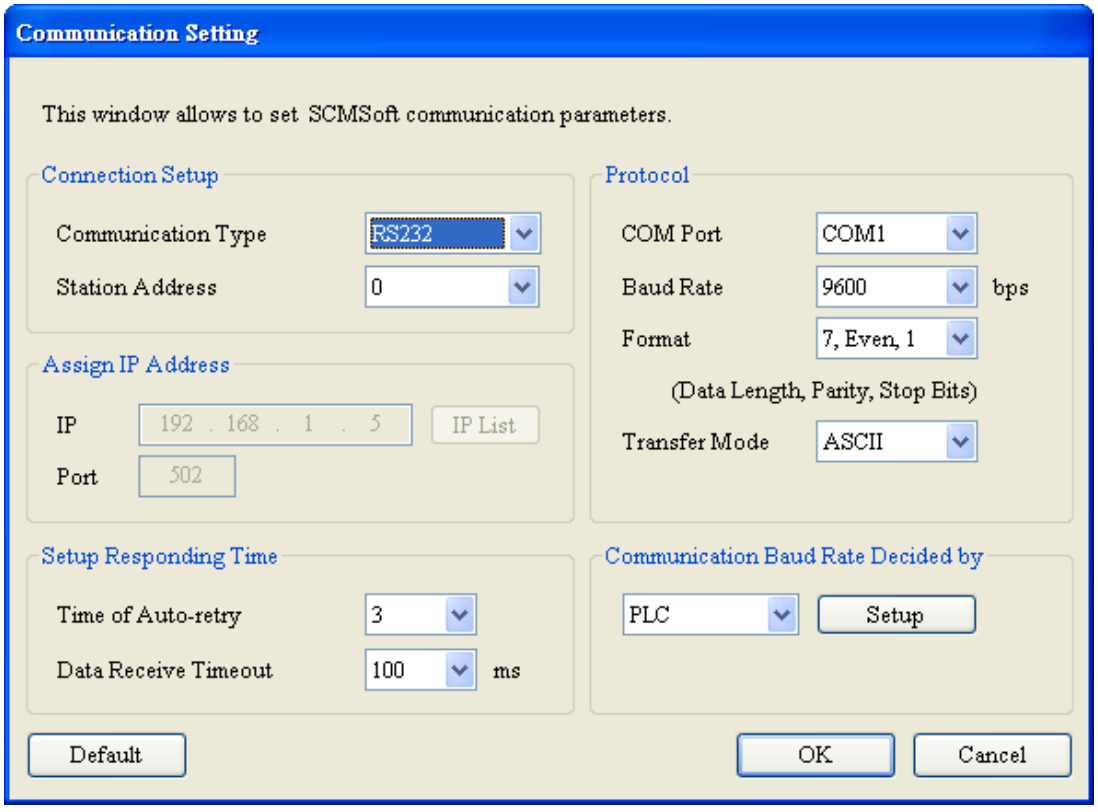

After the setting is complete, the user can designate the communication port used in Modbus Advance ─ COM port 2 on left-side module number 1.



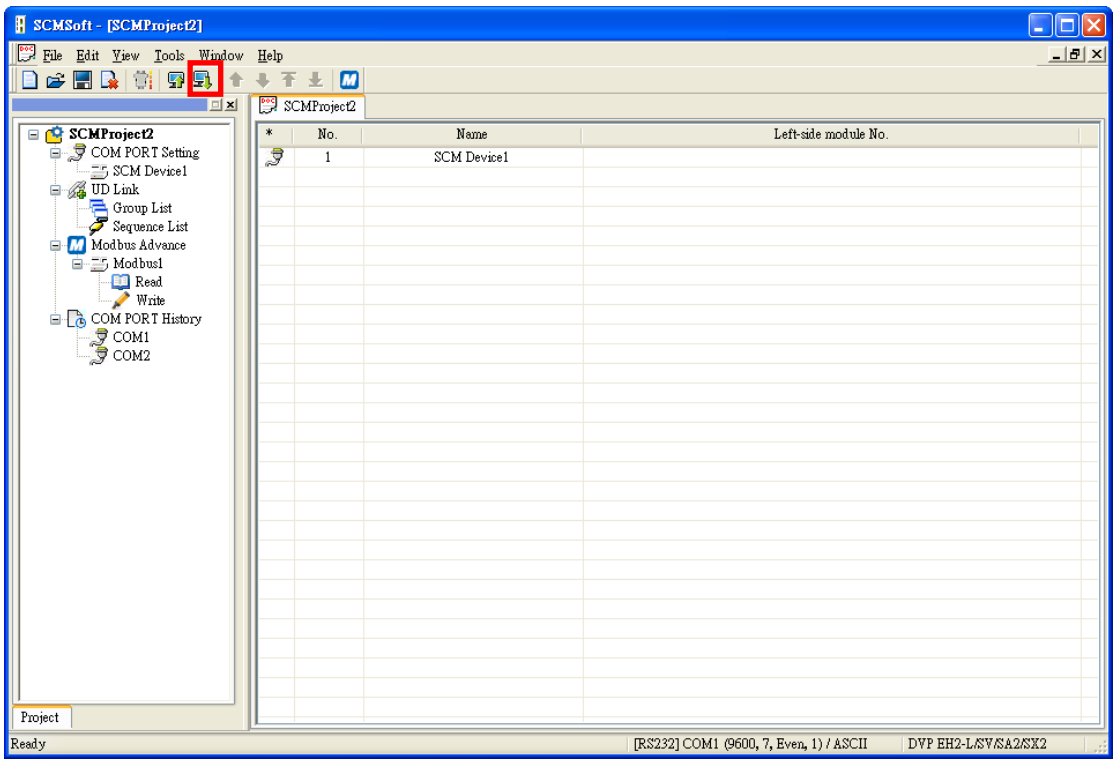
(4) Download

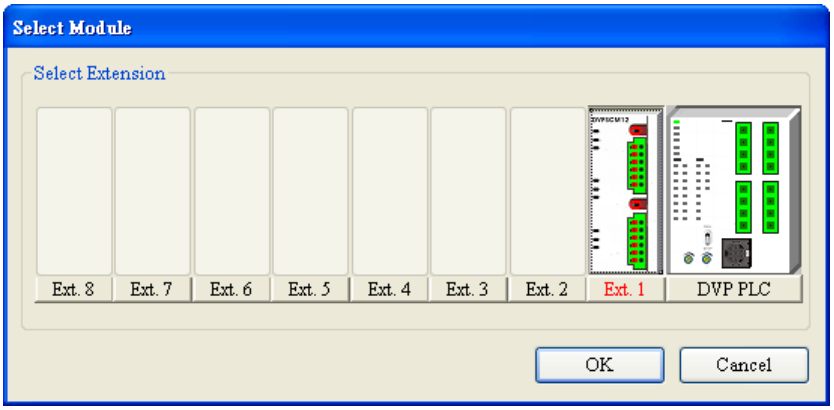
The user can set the communication. After the setting is complete, click “OK” to exit from the communication setting, and the parameters are set.

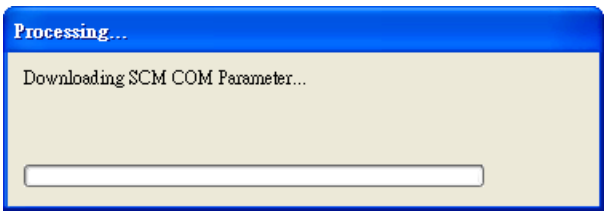


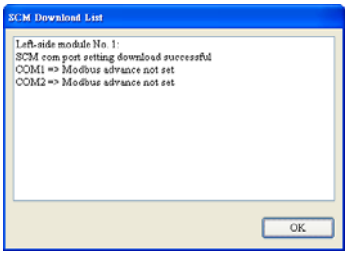




Click “Download Device”, choose the left-side module which will be downloaded, and click “OK”. If only one device is connected, click “OK” directly.




(5) Enable

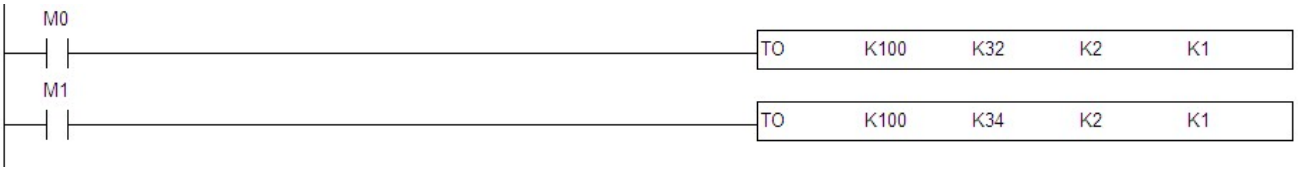
Control the data exchange through “TO instruction” in WPLSoft to read bits/read words/write bits/write words (CR#31~CR#34).

| | | | |
|---|---|---|---|
| 31 | R/W | Trigger the data exchange through COM1 to read bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 32 | R/W | Trigger the data exchange through COM2 to read bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |

| | | | |
|---|---|---|---|
| 33 | R/W | Trigger the data exchange through COM1 to write bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |
| 34 | R/W | Trigger the data exchange through COM2 to write bits or words. | High byte: bit; Low byte: word<br>0: Not trigger, 1: Trigger once, 2: Always trigger |

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 SCM module.

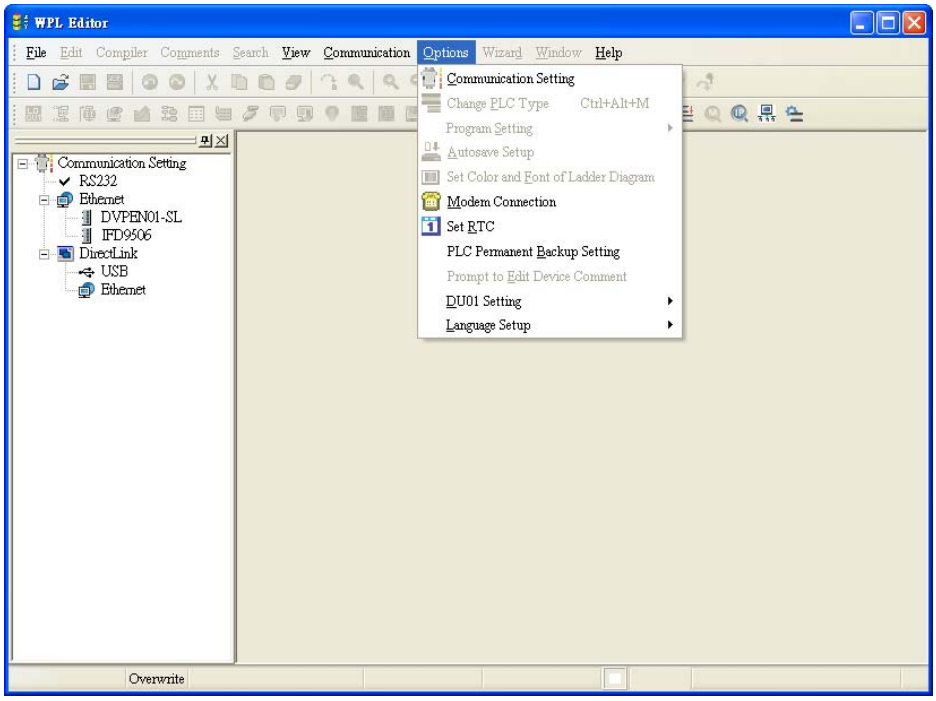
After M1 is triggered, the data will be written into the slave address which has been set through COM2 on SCM module.

## 7.2 Connecting to WPLSoft

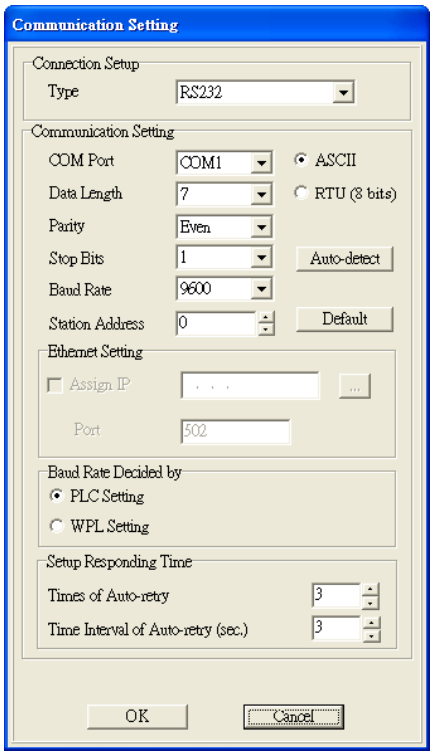
SCM module can be used as the additional communication port of the PLC master. When RS-485 communication of the PLC master is executed, the user can use WPLSoft to monitor the master through SCM module. The default communication format of COM1 on SCM module is 9600, 7, Even, 1, and station number 247.

(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 SCM module. If other communication parameters are used, they need to be modified in COM PORT Setting of SCM module. In addition, the setting of “Station Address” should conform to COM1 on SCM module rather than the station number of the MPU of the PLC.




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

## 7.3 RS-485

This section introduces how DVPSCM12-SL 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 SCM module connects to the electricity meter through RS-485 in UD Link.

(1) The record type

Set the station number 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 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<br>Abbreviated record | N/A |
| 2 | Query about the status of the device: abbreviated record | Abbreviated record |
| 3 | Measured value and error (cyclic data)<br>Abbreviated record | Full record |
| 4 | Event data analyzed erroneously<br>Abbreviated record | Full record |
| 5 | Measured value<br>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 UD Link

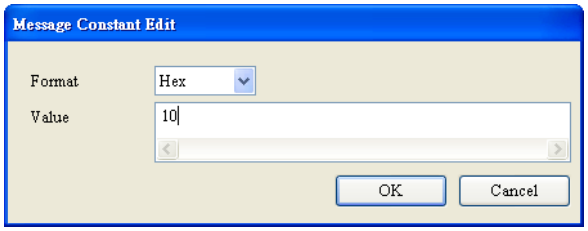
**【Type 1】**

Only send the abbreviated record (abbreviated record):

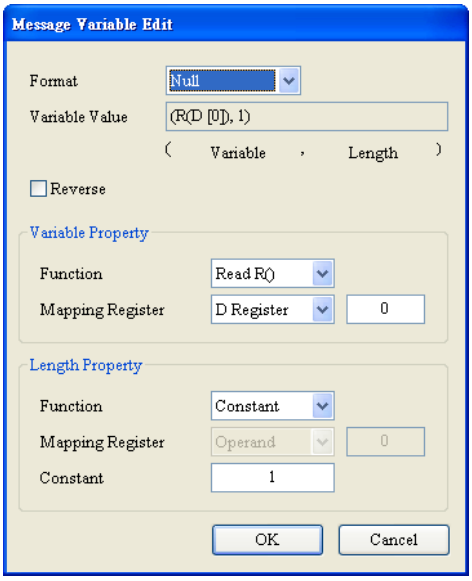
『Start word』+『device address (IA)』+『Function code (FF)』+『Checksum (CS)』+『End marker』

→ 10h + D0 + 09h + (IA+FF) + 16h

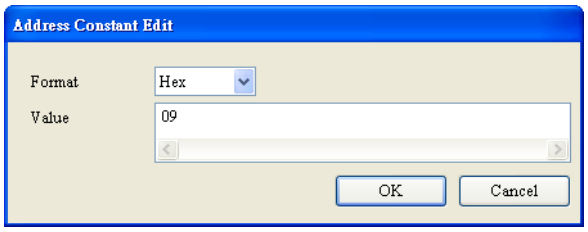
- Start word: 10h



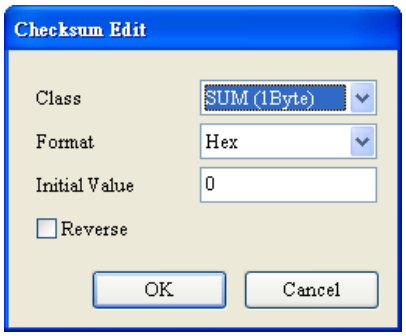

- Read the device address from D0 (IA).




- Function code (FF): 09h

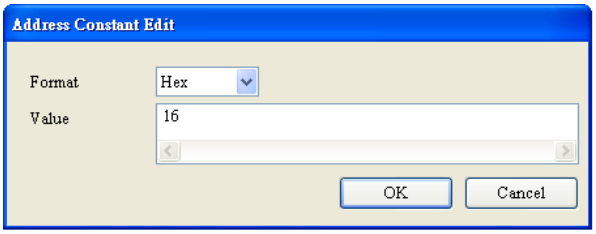



- Checksum (1byte; adding the previous two items up):

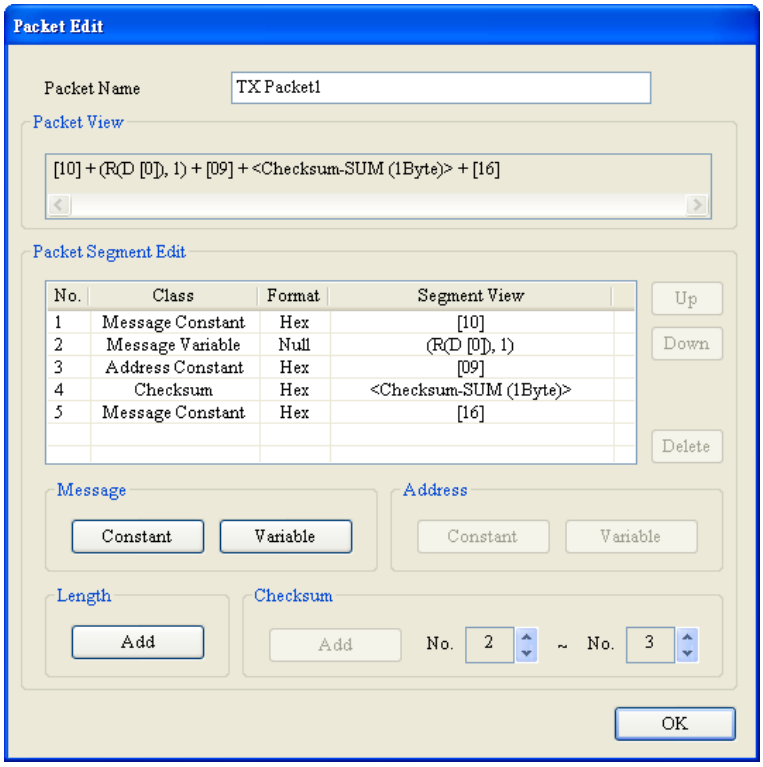




- End word: 16h

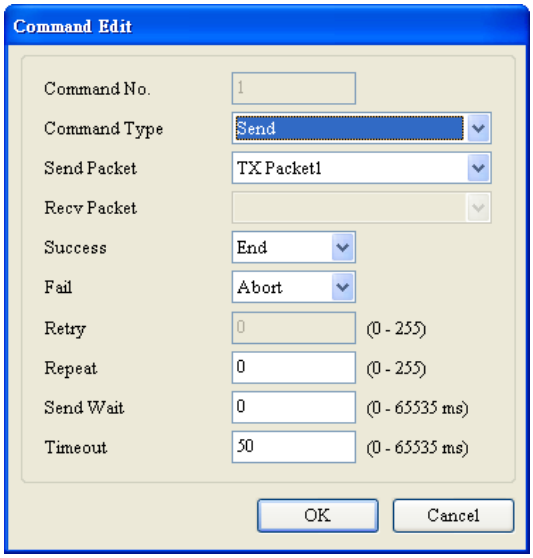


- The editing is complete:




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

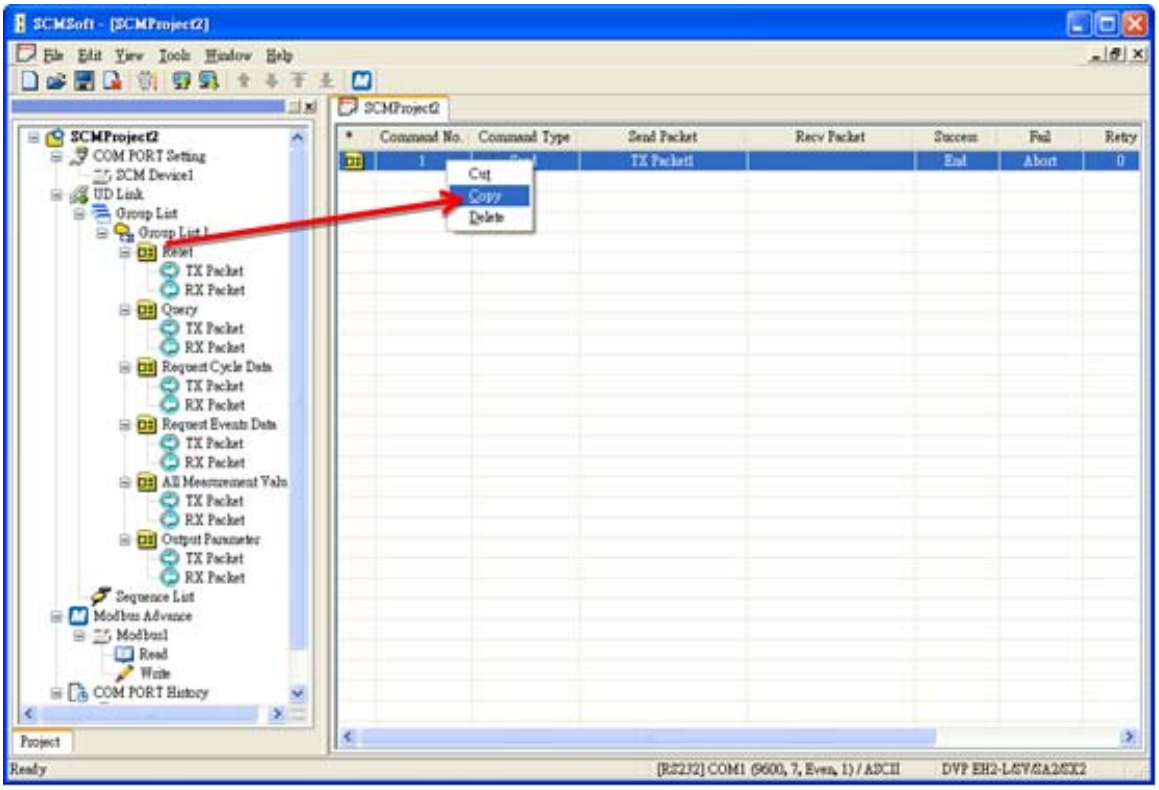
Edit the command: Sending Tx Packet1; no response address



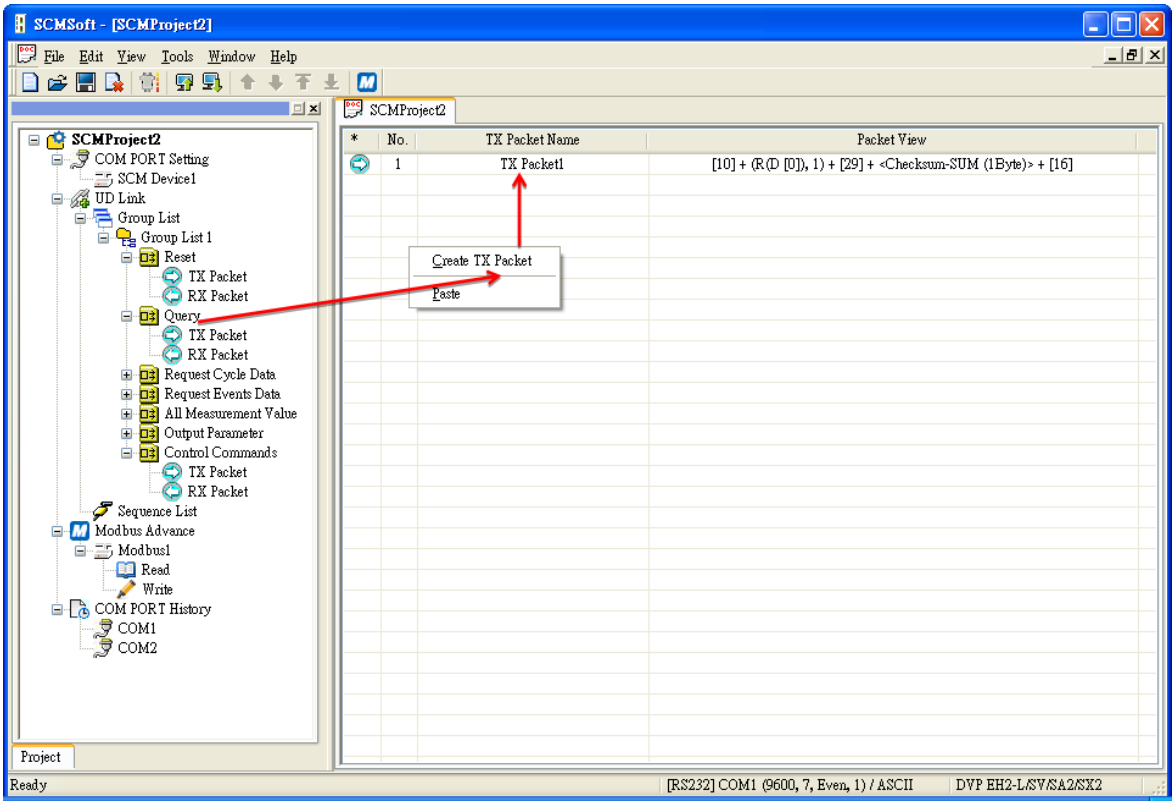
**【Type 2】**

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.




- Paste the setting to TX Packet in Query group.

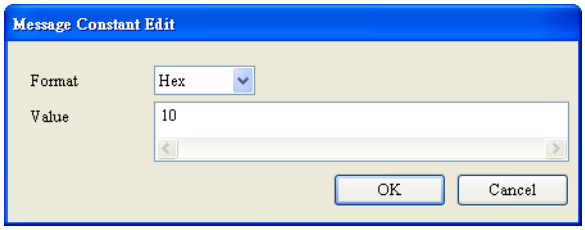



Respond with the abbreviated record.

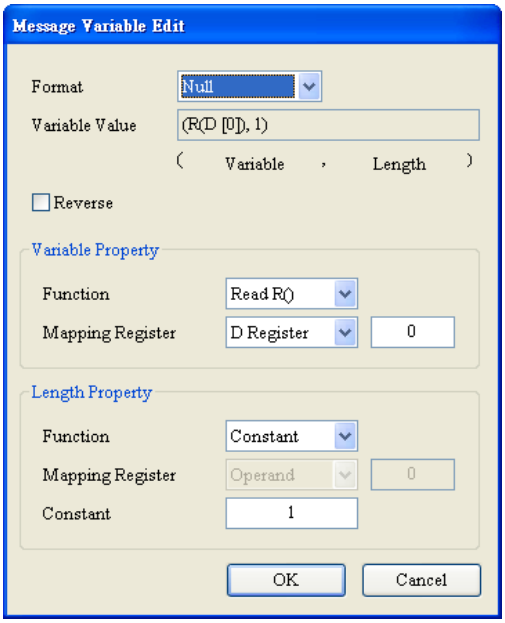
『Start word』+『Device address (IA)』+『Function code (FF)』+『Checksum (CS)』+『End marker』

→ 10h + D0 + 09h + (IA+FF) + 16h

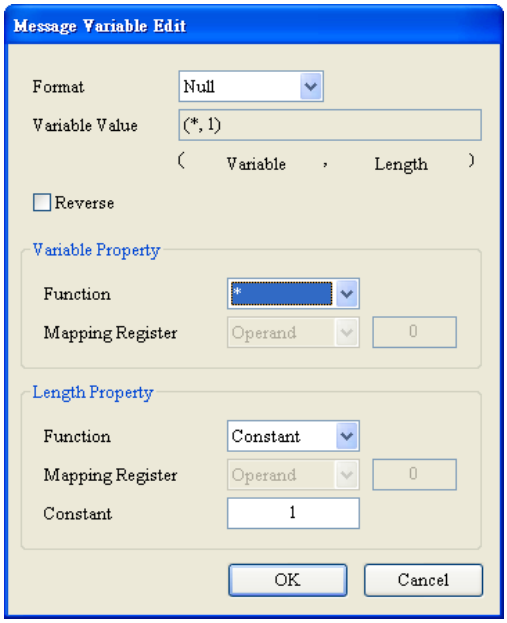
- Start word: 10h



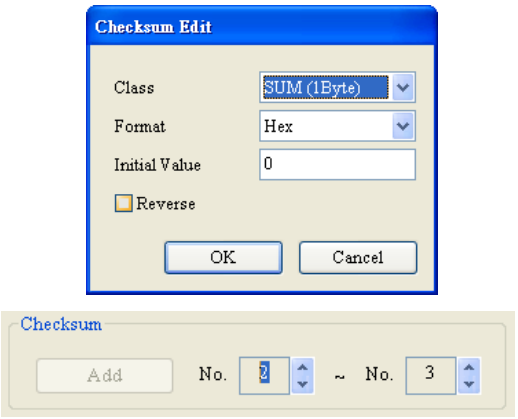

- Check whether the response address and the device address previously read from D0 (IA) are the same.



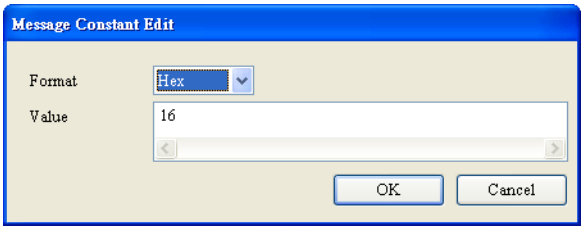

- Ignore the function code (FF) of the response: (*, 1): Ignore the word whose length is 1. If the user wants to store the function code, the user can refer to the setting of the device address (IA) to store the function code in the D register.



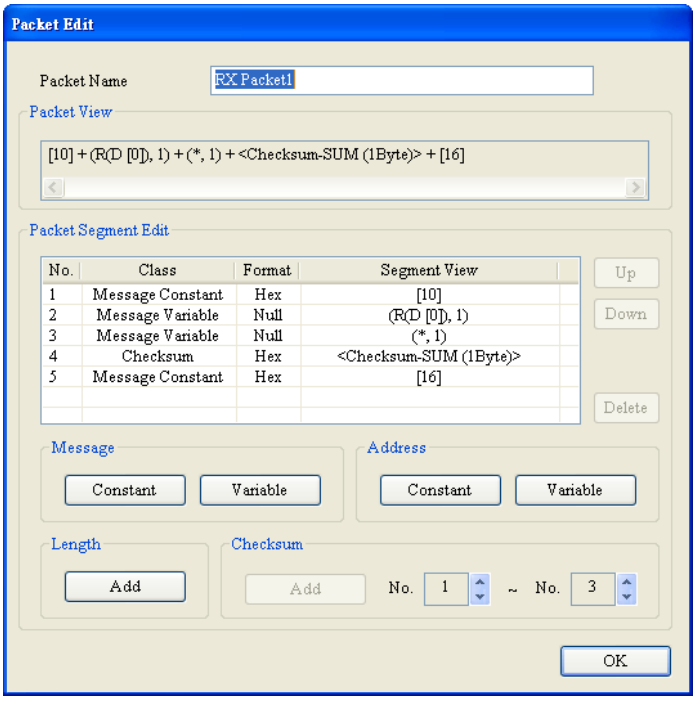

- Checksum (1byte, adding the previous two items up):



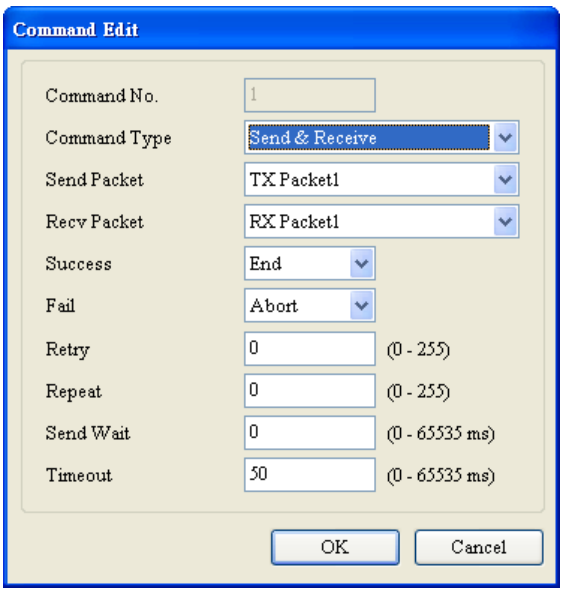
- End word: 16h




- The editing is complete:




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

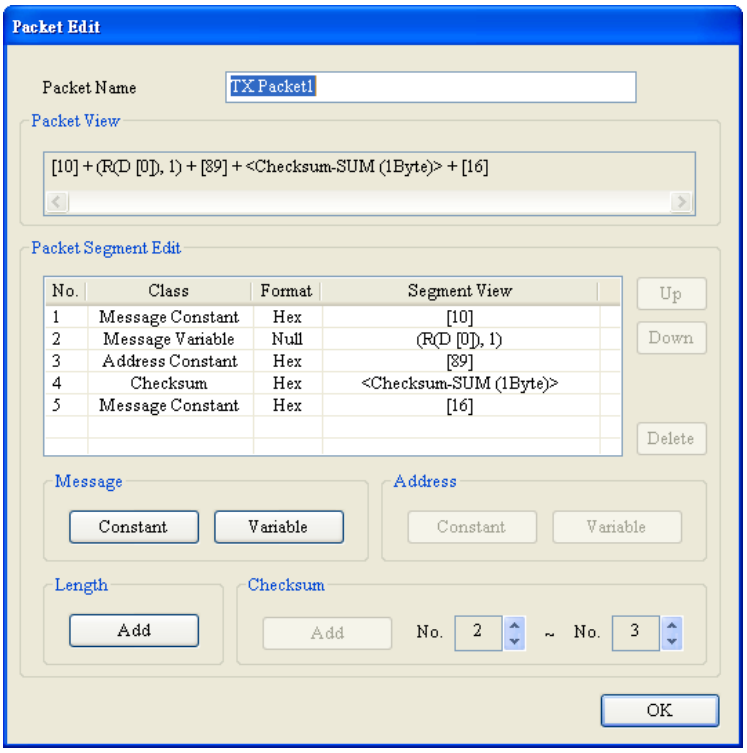


**【Type 3】**

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.

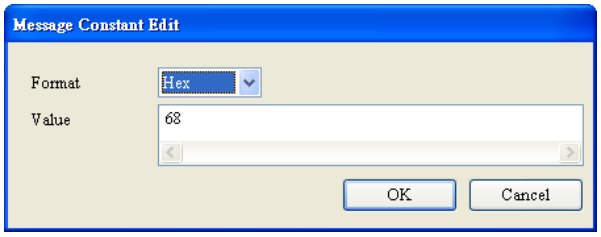



Respond with the full record.

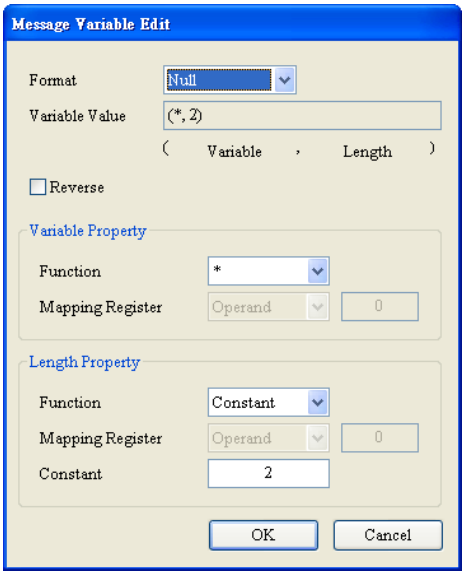
『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Data block (DB)』+『Checksum (CS)』+『End marker』

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

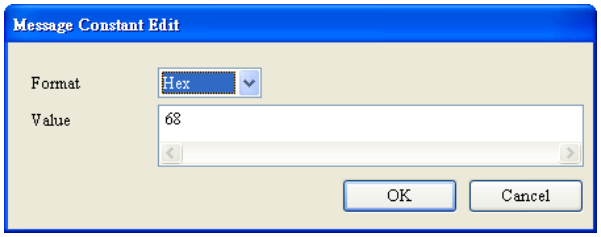
■ Start word: 68h



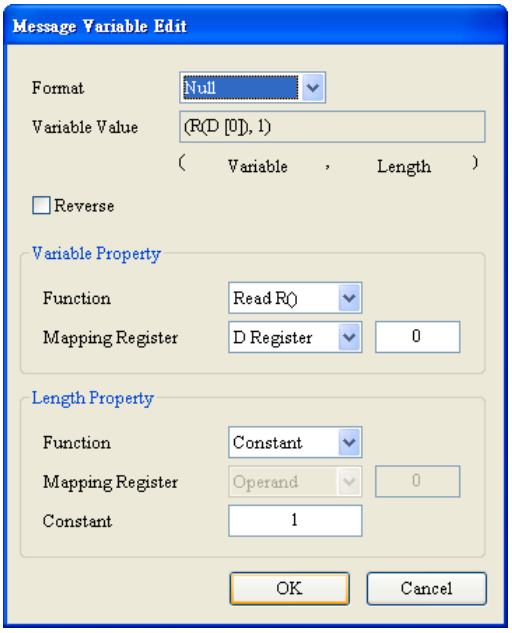

■ Length + Length (repeat): Ignore these two words; They can be ignored or stored.



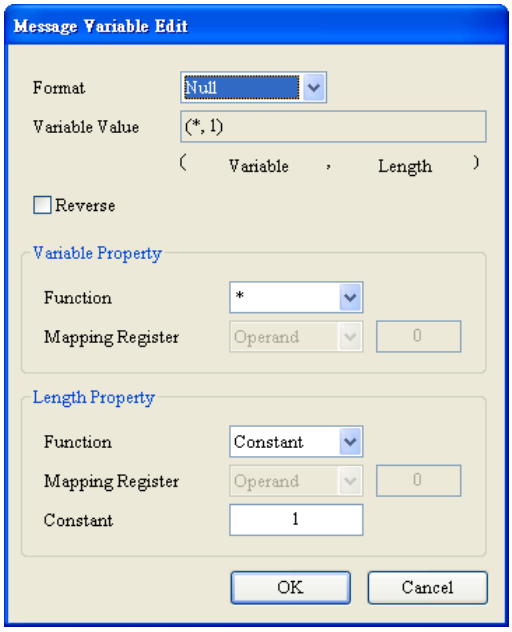

■ Start word: 68h



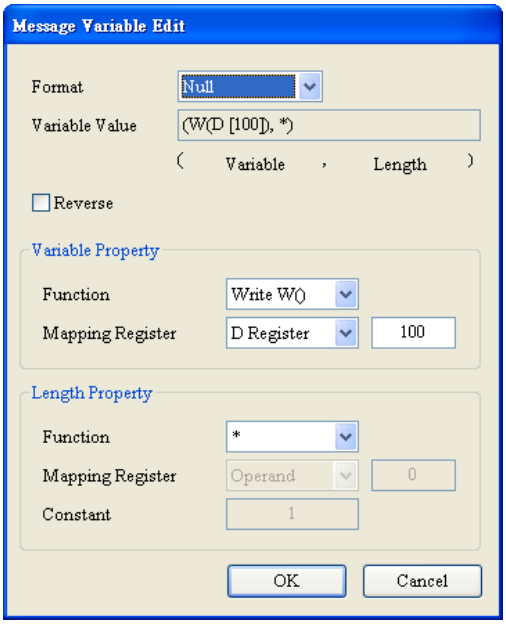

■ Device address (IA): Check whether the response address and the device address previously read from D0 (IA) are the same.

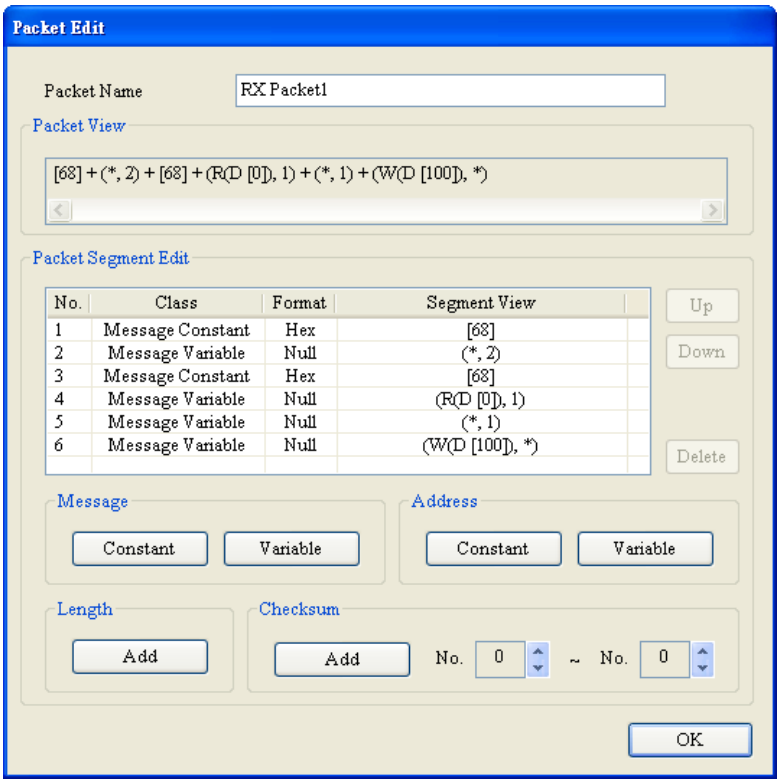


- Function code: Ignore the word.



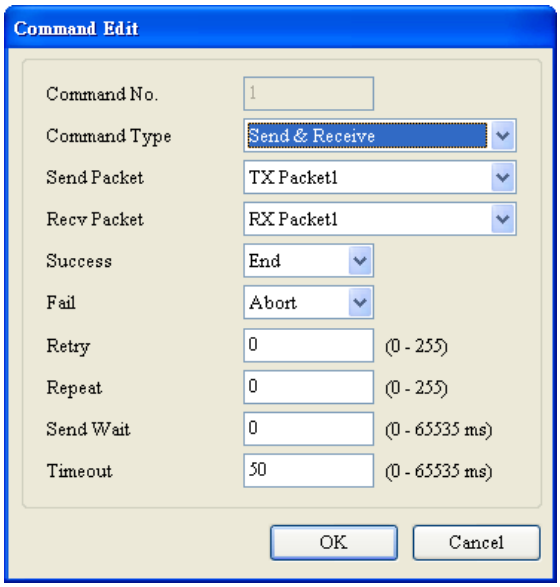

- The data after the function code is stored in the registers starting from D100. (Note)




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.

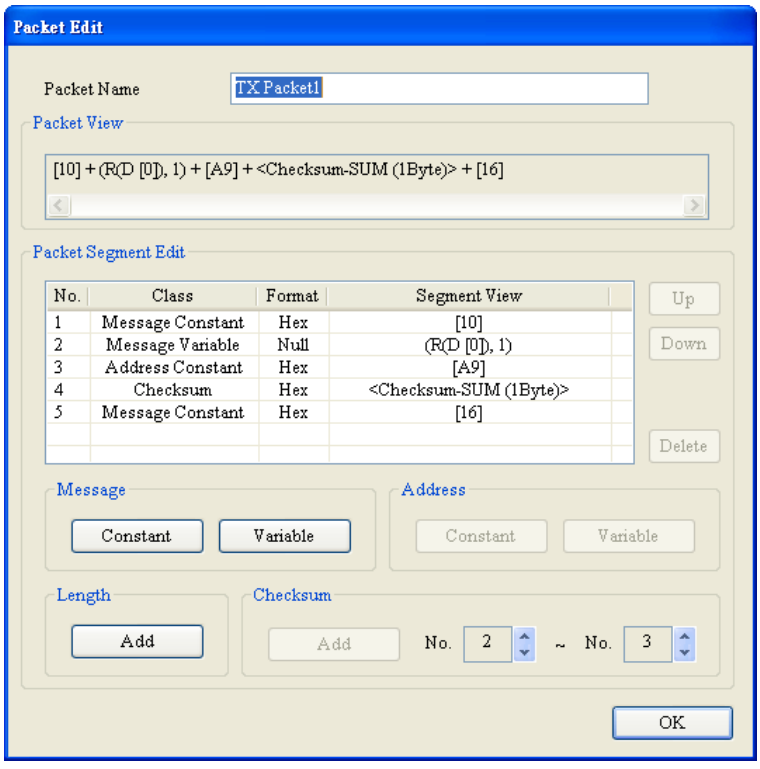
Edit the command: Sending Tx Packet1, and receiving Rx Packet1




**【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.

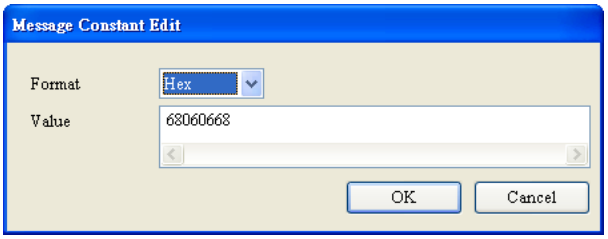



Respond with the full record.

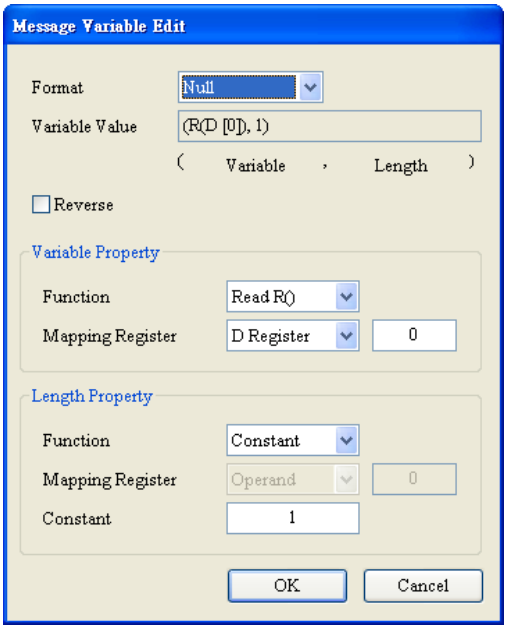
『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Data block (DB)』+『Checksum (CS)』+『End marker』

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

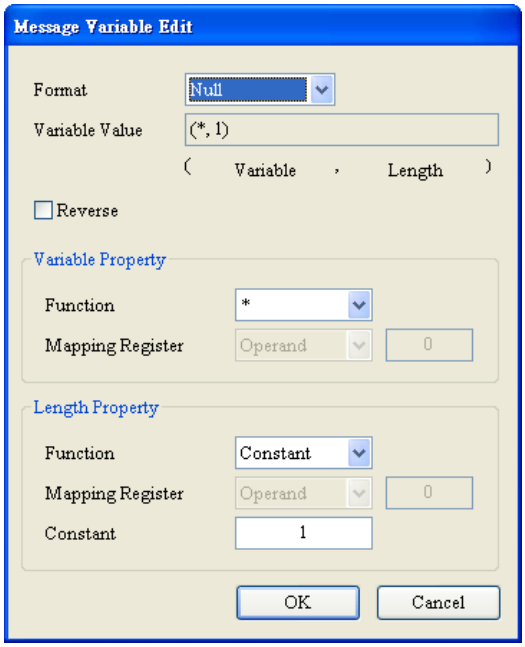
■ Star word-Length-Length-Star word



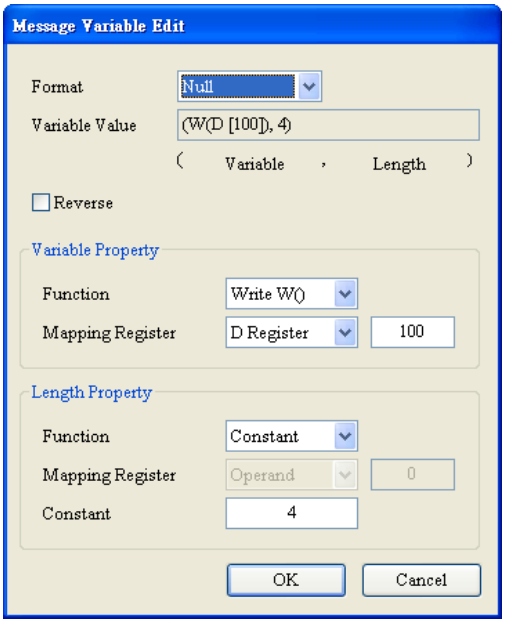

■ Check whether the response address and the device address previously read from D0 (IA) are the same.



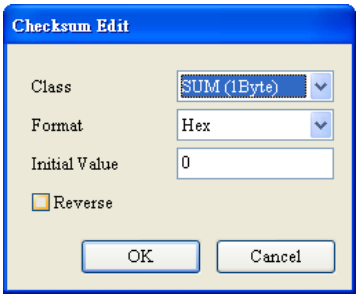

■ FF：Ignore the function code.

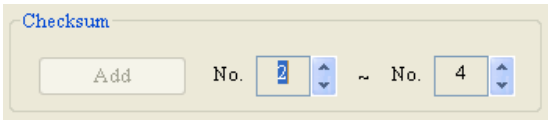



■ Store PI+DB in D100.

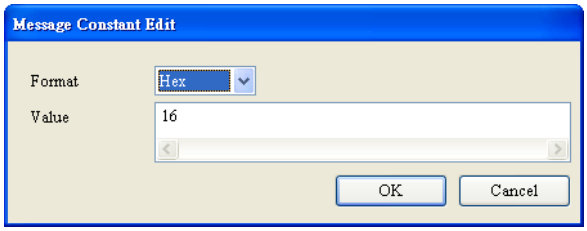


■ Checksum:

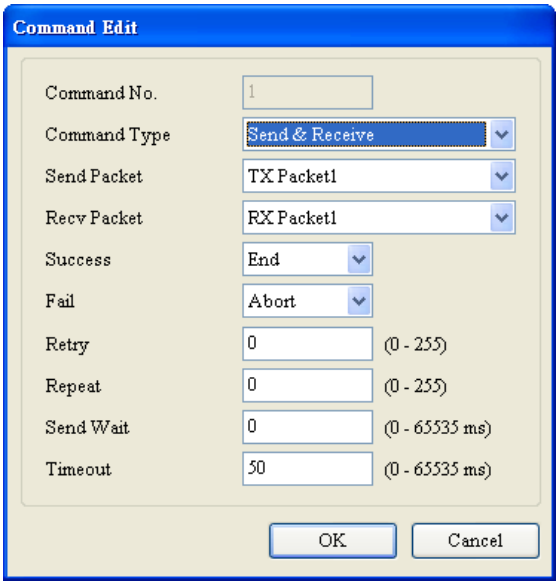



■ End word:




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




**【Type 5】**

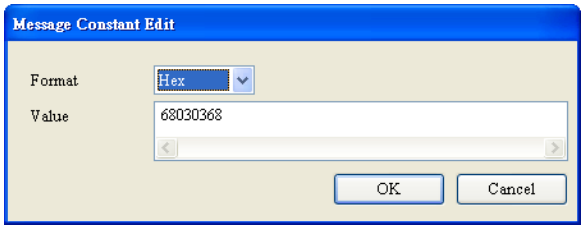
Send the abbreviated record, and respond with the full record.

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

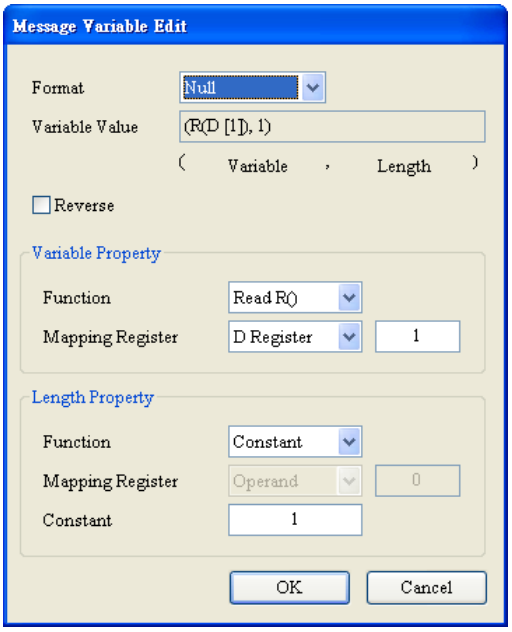
『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Checksum (CS)』+『End marker』

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

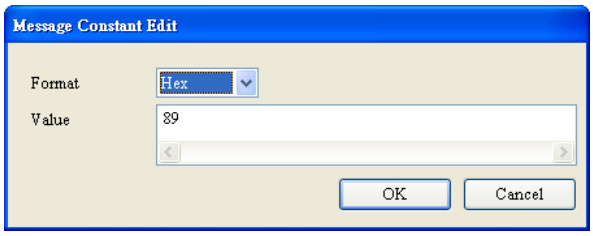
■ Start word-Length-Length-Start word



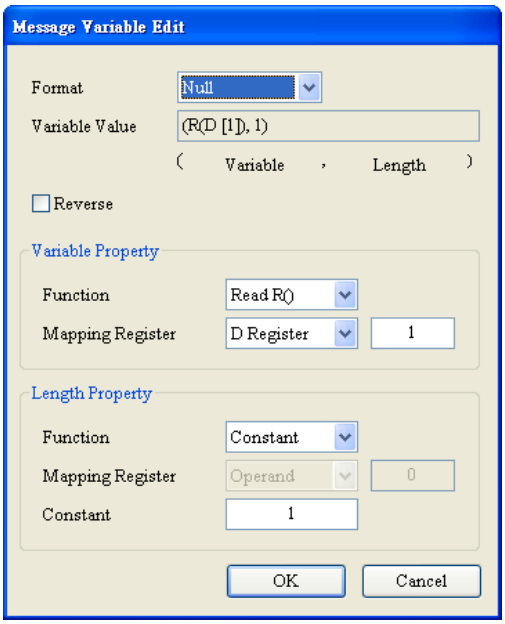

■ The device address is read from D0.



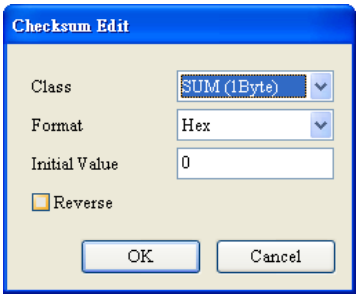

■ Function code: 89h

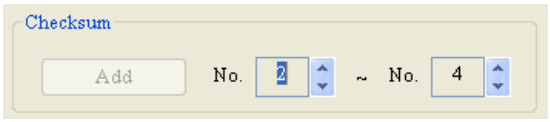



■ The parameter index is read from D1.

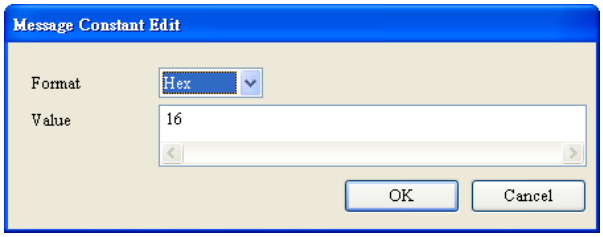


■ Checksum:




■ End word:

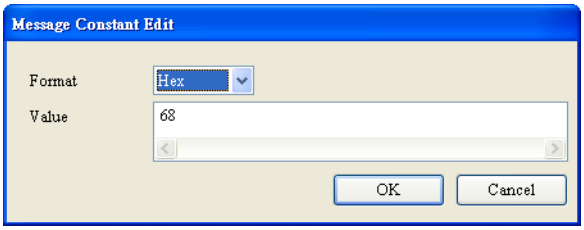



Respond with the full record.

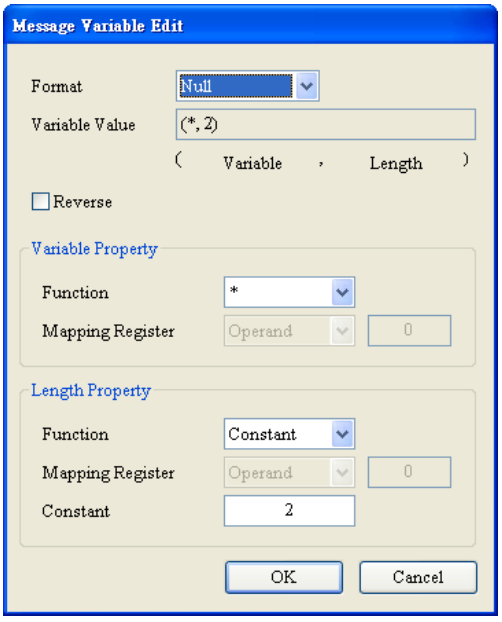
『Start word』+『Length』+『Length (repeat)』+『Start word』+『Device address (IA)』+『Function code (FF)』+『Parameter index (PI)』+『Data block (DB)』+『Checksum (CS)』+『End marker』

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

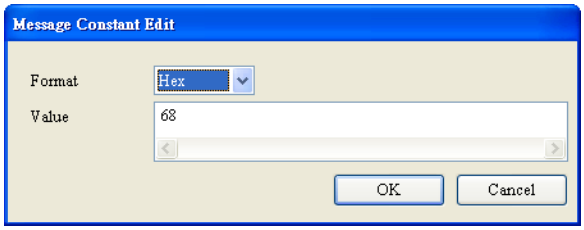
■ Start word:



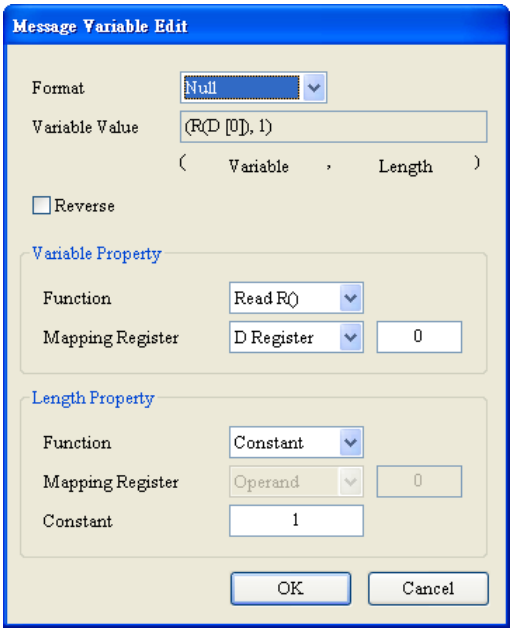

■ Length-Length (two words): Ignore the two words.



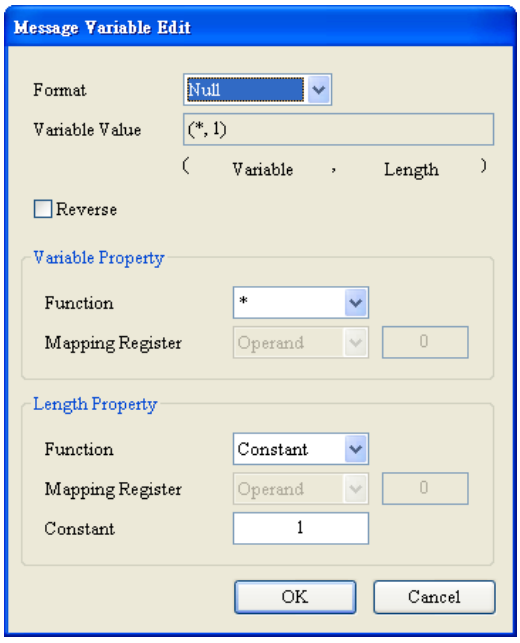
■ Start word: 68h



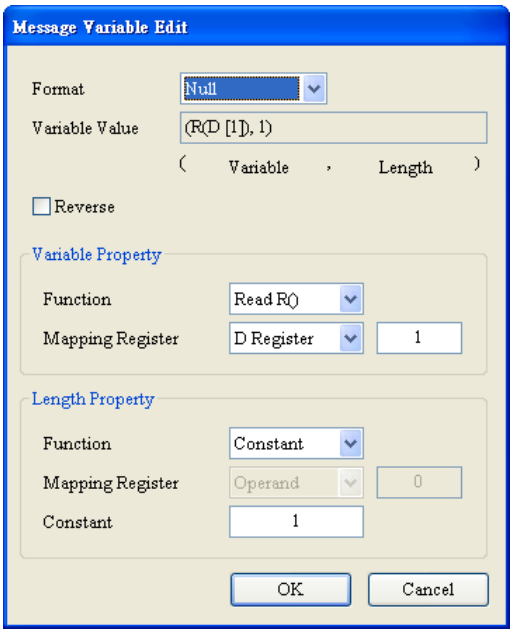

■ Check whether the response address and the device address previously read from D0 (IA) are the same.



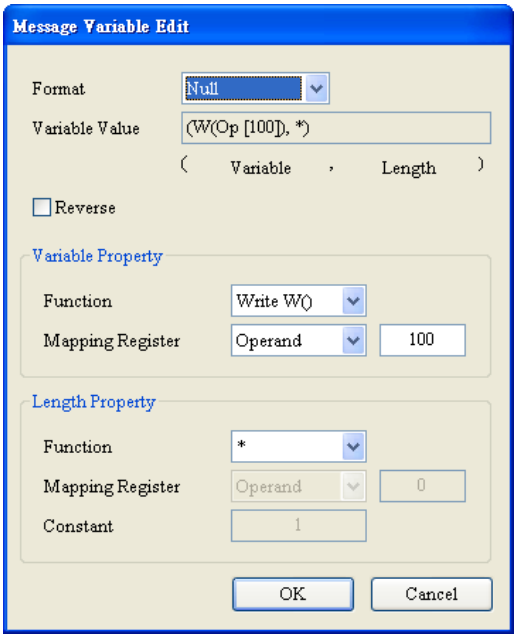

■ Function code:




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

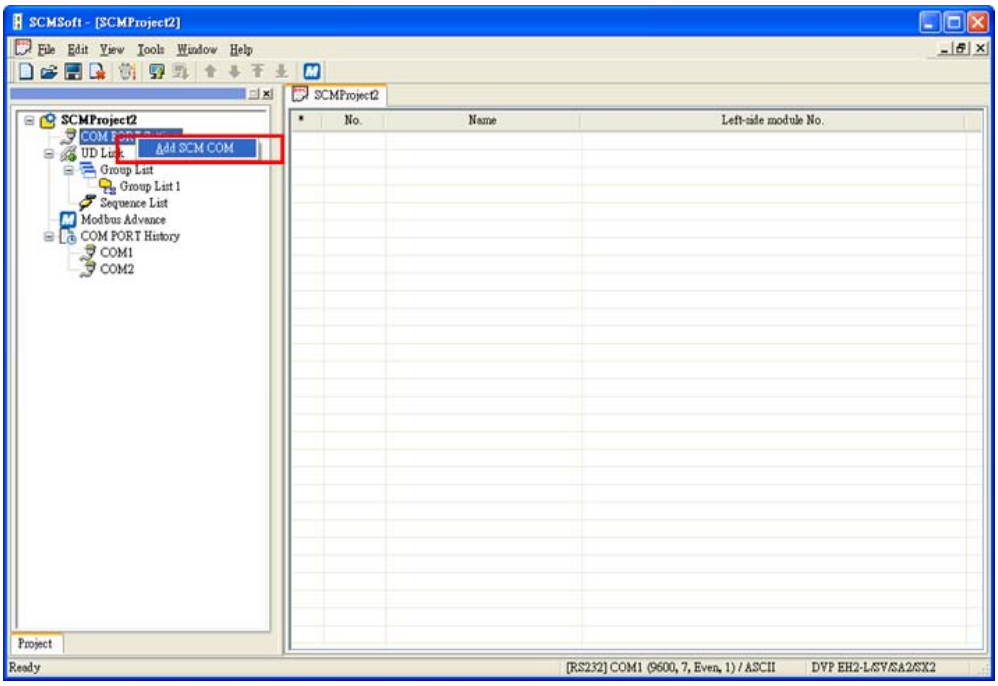


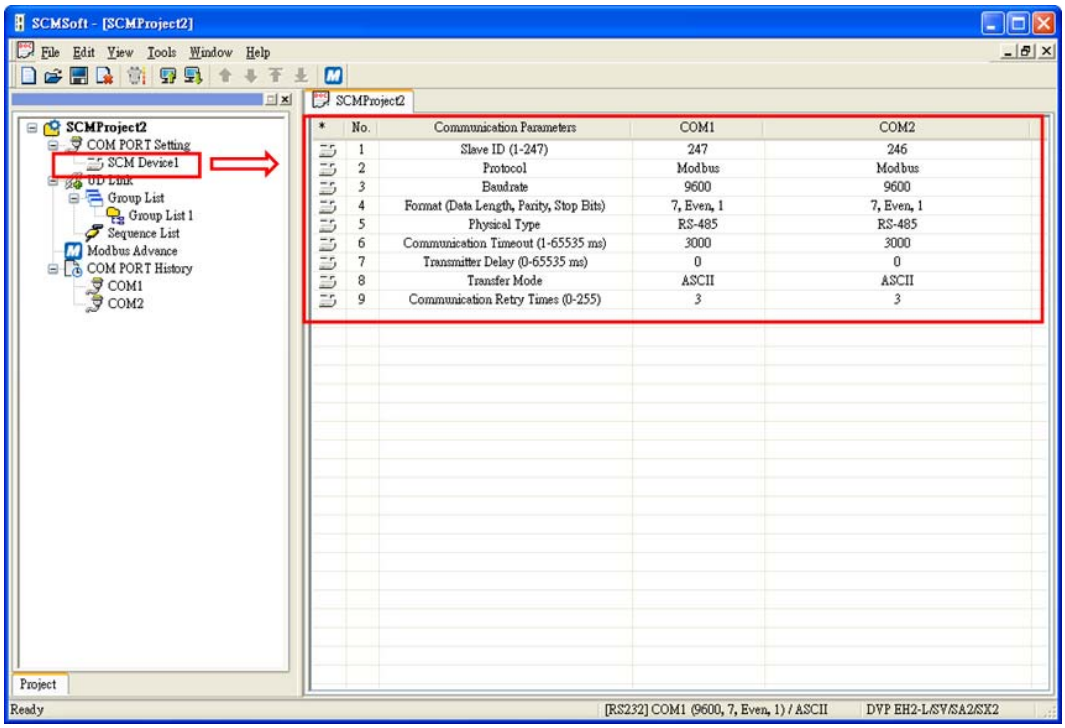
- Data block: The response data is stored in the registers starting from D100.




(4) Download

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






| * | No. | Communication Parameters | COM1 | COM2 |
|---|---|---|---|---|
| | 1 | Slave ID (1-247) | 247 | 246 |
| | 2 | Protocol | Modbus | Modbus |
| | 3 | Baudrate | 9600 | 9600 |
| | 4 | Format (Data Length, Parity, Stop Bits) | 7, Even, 1 | 7, Even, 1 |
| | 5 | Physical Type | RS-485 | RS-485 |
| | 6 | Communication Timeout (1-65535 ms) | 3000 | 3000 |
| | 7 | Transmitter Delay (0-65535 ms) | 0 | 0 |
| | 8 | Transfer Mode | ASCII | ASCII |
| | 9 | Communication Retry Times (0-255) | 3 | 3 |

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

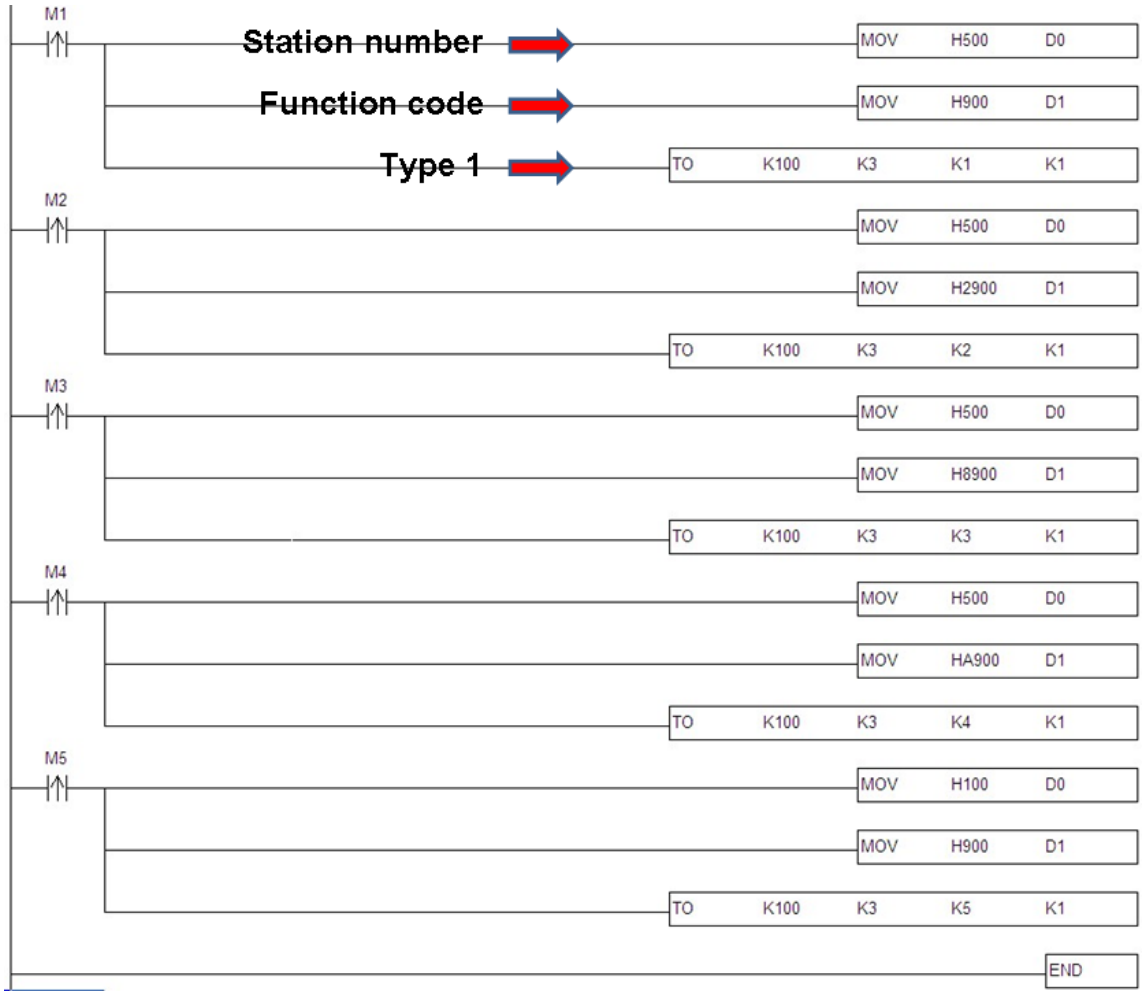
| * | No. | Communication Parameters | COM1 | COM2 |
|---|---|---|---|---|
| | 1 | Slave ID (1-247) | 247 | 246 |
| | 2 | Protocol | UD Link | Modbus |
| | 3 | Baudrate | 9600 | 9600 |
| | 4 | Format (Data Length, Parity, Stop Bits) | 7, Even, 1 | 7, Even, 1 |
| | 5 | Physical Type | RS-485 | RS-485 |
| | 6 | Communication Timeout (1-65535 ms) | 3000 | 3000 |
| | 7 | Transmitter Delay (0-65535 ms) | 0 | 0 |
| | 8 | Transfer Mode | ASCII | ASCII |
| | 9 | Communication Retry Times (0-255) | 3 | 3 |

(5) WPLSoft triggers UD Link

The command numbers set in each type are triggered by “To instruction” in WPLSoft which triggers the execution of UD Link. K1 is written into CR3. When the command number is 1 when the command number is 1, and by analogy, K2 is written into CR3 when the command 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 number 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 number is 5, and the same applies to the reading of the response address from D100.


M1
Station number
Function code
Type 1
MOV H500 D0
MOV H900 D1
TO K100 K3 K1 K1
M2
MOV H500 D0
MOV H2900 D1
TO K100 K3 K2 K1
M3
MOV H500 D0
MOV H8900 D1
TO K100 K3 K3 K1
M4
MOV H500 D0
MOV HA900 D1
TO K100 K3 K4 K1
M5
MOV H100 D0
MOV H900 D1
TO K100 K3 K5 K1
END

**MEMO**

## 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 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 |
| 0x0010 | Restore the factory default. |

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 | | | |