

ModView Message Editor 1.0

EKAN ME-100M



Table of Contents

CHAPTER 1 MODVIEW MESSAGE EDITOR INSTALLATION	4
1-1 MODVIEW MESSAGE SOFTWARE INSTALLATION	4
1-2 START MODVIEW MESSAGE EDITOR	6
1-3 MODBUS ADDRESS SETTING	8
CHAPTER 2 VARIABLE CONFIGURATION	9
2-1 BOOLEAN VARIABLE CONFIGURATION.....	10
2-2 FLOAT VARIABLE CONFIGURATION	12
CHAPTER 3 MESSAGE EDITING.....	14
3-1 MESSAGE WITH TEXT	14
3-2 MESSAGE WITH VARIABLE (I/O DATA, DATE/TIME).....	18
3-3 MESSAGE WITH GRAPHIC	21
CHAPTER 4 UPLOAD AND DOWNLOAD MESSAGE PROJECT	23
4-1 UPLOAD MESSAGE FILES	23
4-2 DOWNLOAD MESSAGE FILES.....	24
CHAPTER 5 TEST MESSAGE DISPLAY	25
CHAPTER 6 MODBUS PROTOCOL INTRODUCTION	27
6-1 MODBUS FRAME FORMAT (WRITE SINGLE COIL)	27
6-2 MODBUS FRAME FORMAT (WRITE MULTIPLE REGISTERS)	29
CHAPTER 7 CONNECTION WITH VISUAL BASIC CLIENT (MODBUS TCP)	32
7-1 TRIGGER REGULAR MESSAGE	32
7-2 UPDATE VARIABLES	32
<i>Update float variable</i>	33
<i>Update Boolean variable</i>	34
7-3 TRIGGER EMERGENCY MESSAGE.....	35

CHAPTER 8 CONNECT WITH OPC SERVER	36
8-1 TAG CONFIGURATION	36
8-2 SET REGULAR MESSAGE COIL	37
8-3 SET FLOAT VARIABLE REGISTER	38
8-4 SET BOOLEAN VARIABLE COIL	39
8-5 SET EMERGENCY MESSAGE COIL	40
8-6 CREATE OPC CLIENT	41
CHAPTER 9 CONNECT WITH INDUSOFT (HMI/SCADA).....	42
9-1 CREATE INDUSOFT PROJECT AND CREATE TAGS.....	43
9-2 MAPPING VARIABLES TO BUTTON	44
9-3 MAPPING VARIABLES WITH MODBUS TCP	45
CHAPTER 10 CONNECT WITH ISAGRAF (SOFTPLC).....	47
10-1 SET 7188EG AS MODBUS MASTER DEVICE	48
10-2 USING FUNCTION BLOCK	49
10-3 MODBUS FUNCTION BLOCK ON ISAGRAF.....	50
ABOUT ESOFTSYSTEM CORP.....	51

Chapter 1 ModView Message Editor Installation

EKAN ME-100M use ModView script as display control language, ModView message editor could help you generate these scripts, and upload/download these script into EKAN ME-100M. Following chapter will teach you to use message editor to edit messages. And how to use third party client like Visual Basic, InduSoft, OPC...to control EKAN ME-100M message display.

1-1 ModView message software installation

Your computer is required to fit the minimum requirement in order to install ModView message software.

Operation system version: Windows 2000 or higher

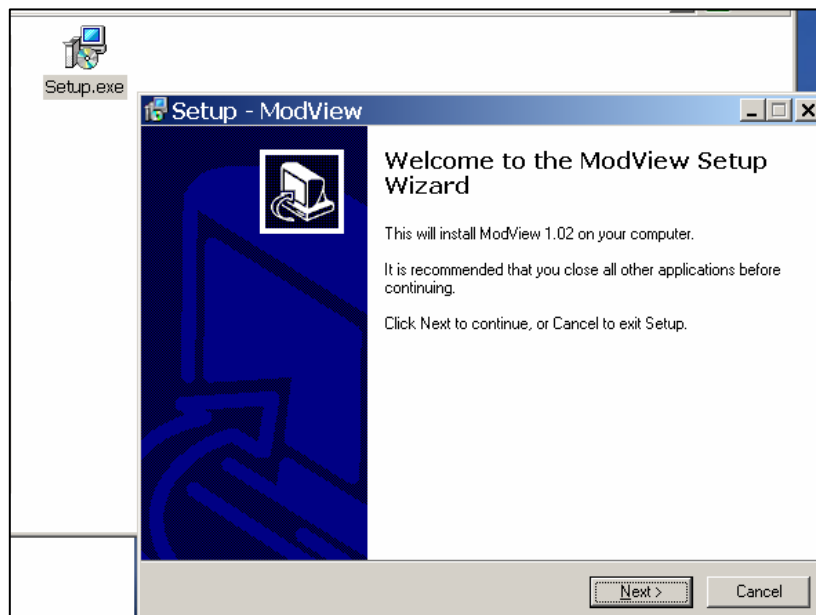
CPU: Pentium II 350 MHz or higher

RAM: 128MB or higher

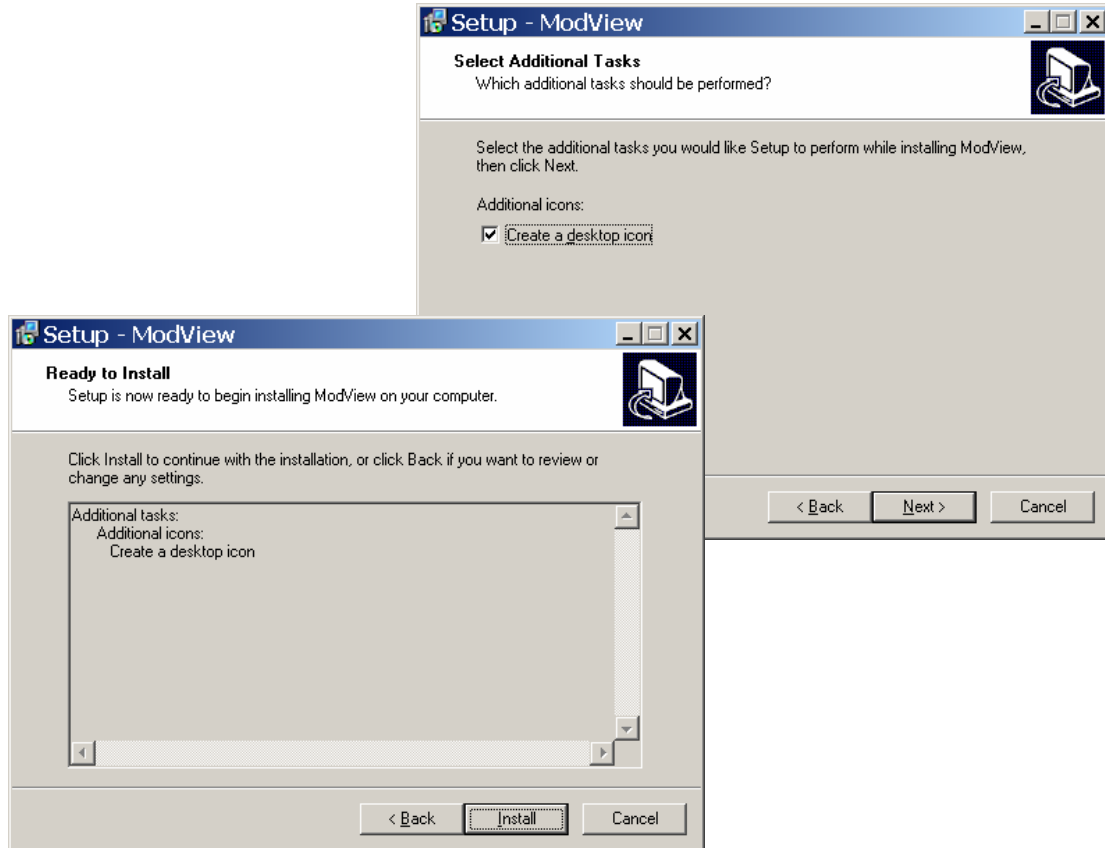
Display: Standard display

Disk space needed: 5 MB or Higher

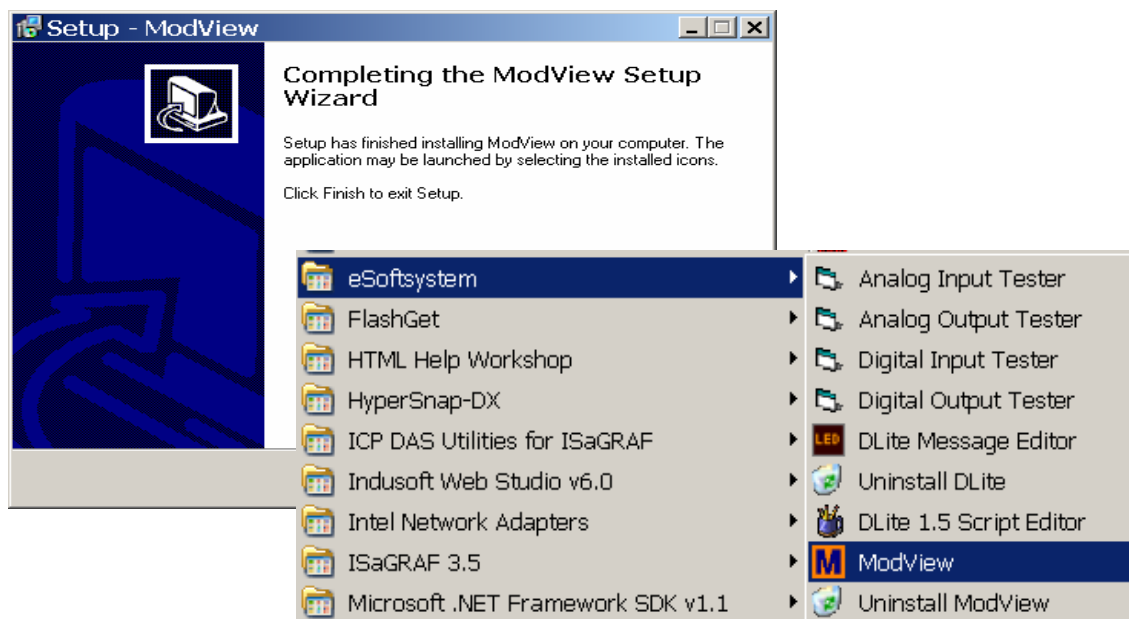
Click on **Setup** icon in your EKAN CD root directory, press **[Next]** to continue:



Please select create desktop icon (if needed), Press **[Next]** and the **[Install]** to install ModView program.



After install, You could use **[Start/Program/eSoftsystem/ModView]** to start ModView program:

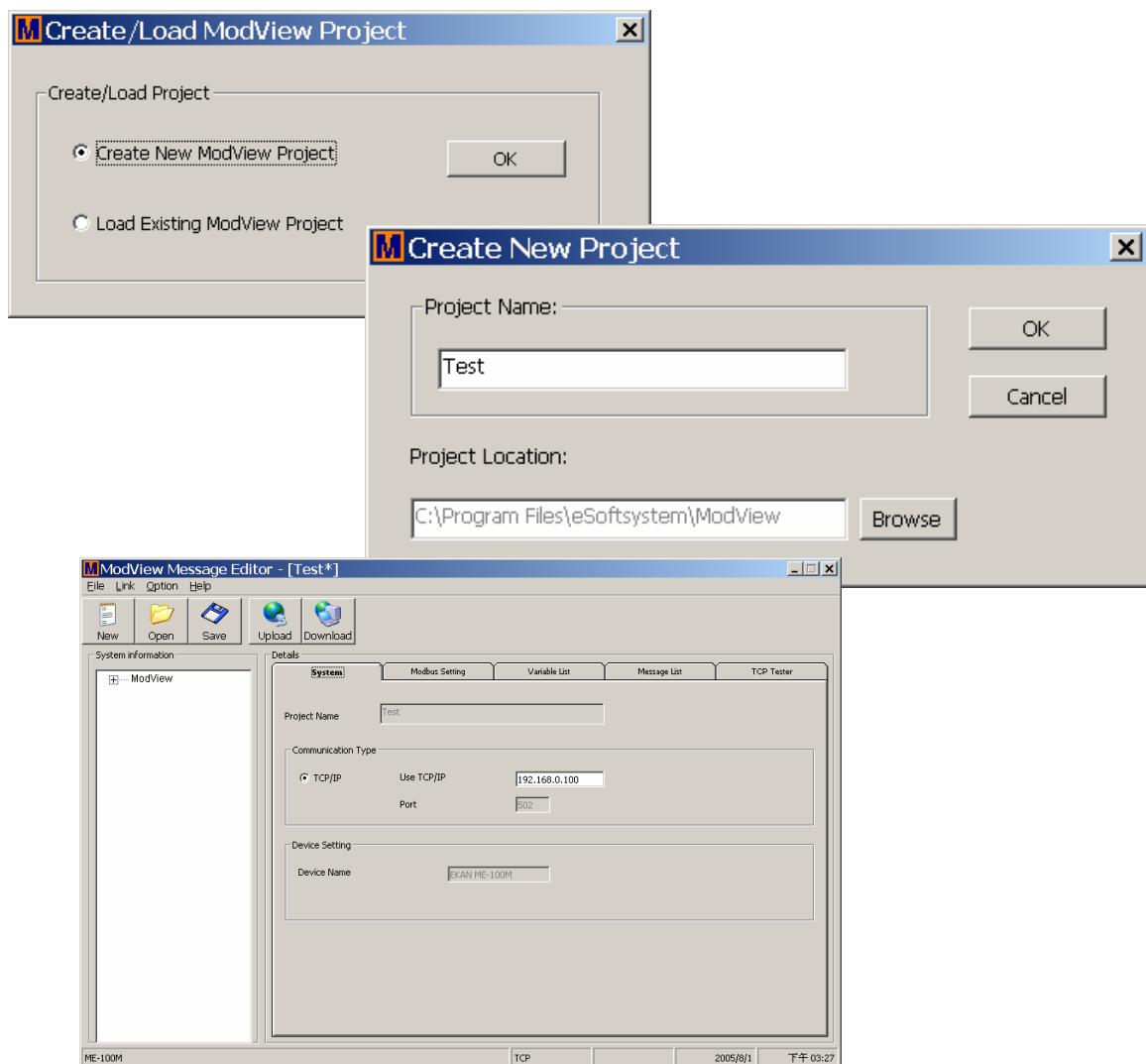


1-2 Start ModView message editor

After successfully install ModView message editor, you could use **[Start/Program/eSoftsystem/ModView]** to start ModView program.

Click on the message editor icon, select create a new project and give project name.

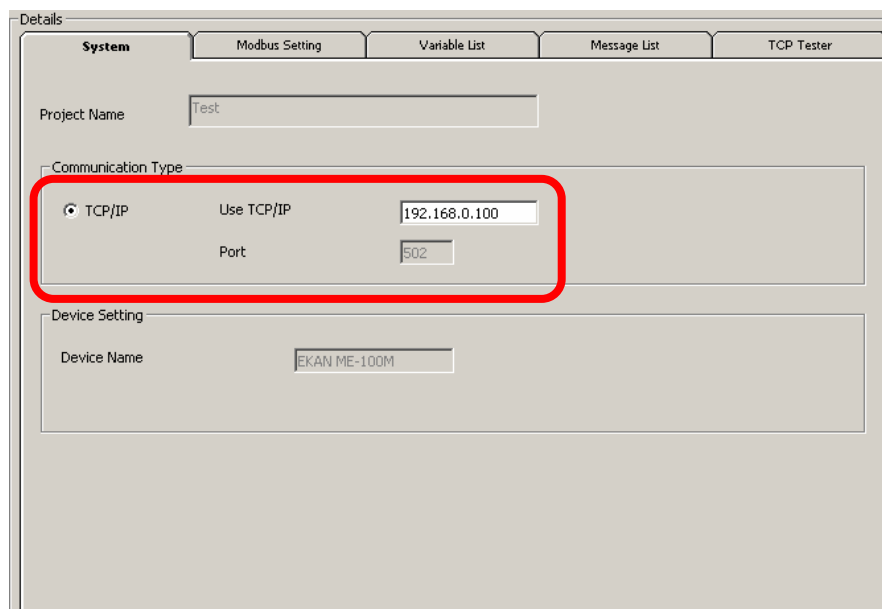
1. Select create a new project
2. Give the project name "**Test**"
3. Change the project location (if needed)
4. Enter the ModView main manual



After project created, Press **[Save]** button, the project file will be store at project directory and with the file name XXXX.mvp. XXXX is file name. And the upload message files will be generated at same time.



Input the ModView IP address in the TCP/IP text box, when the project files upload, **ModView Tester** will be use this IP setting for system configuration, **Make sure the IP Address is CORRECT here:**



1-3 MODBUS address setting

ModView could have 40 regular message groups and 20 emergency message groups, each group could be triggered individually and have their own message scripts. Each message group could have one line or multi lines of message script.

All messages using 60 continuous addresses, default start from 00300. First 20 messages is Emergency messages (00300-00319), other 40 messages is regular messages (00320-00359).

- Float Register:** 40100-40124 (64 float variables)
- Boolean Coil:** 00100-00131 (32 Boolean variables)
- Emergency message:** 00300-00319 (20 message blocks)
- Regular message:** 00320-00359 (40 message blocks)

In some systems, must use 00301-00320 and 00321-00360 to connect systems. You may change these MODBUS setting to fit your software environment setting. You must enter all address required, to let **[APPLY]** take effect.

The screenshot shows the 'Modbus Setting' dialog box with the following fields and values:

Setting Name	Current Address	New Address
Emergency Message Start Address (20 Continuous 0xxxx Coil Address)	300	
Regular Message Start Address (40 Continuous 0xxxx Coil Address)	320	
Float Variable Start Address (64 Continuous 4xxxx Register Address)	100	
Boolean Variable Start Address (32 Continuous 0xxxx Boolean Address)	100	

Buttons at the bottom: Restore Default, Apply

Chapter 2 Variable configuration

ModView support 2 different types of variable: Boolean and Float. Boolean mapped to MODBUS address 0xxxx, user could use MODBUS function 5 to update the Boolean variable status.

Float variable mapped to MODBUS address 4xxxx, user could use function 16 to update Float variable status.

All MODBUS address setting will be saved in upload file: "modbus.adr"

File name	Function
modbus.adr	MODBUS address setting file
default.pls	Play List file
stdmsg.ilm	Standard message configuration file
altmsg.ilm	Emergency message configuration
ilmvar.def	Boolean variable and float variable address and format definition file

EKAN ME-100M supports 64 float variables and 32 Boolean variables. Default start address for float variables is 40100, and for Boolean variables is 00100.

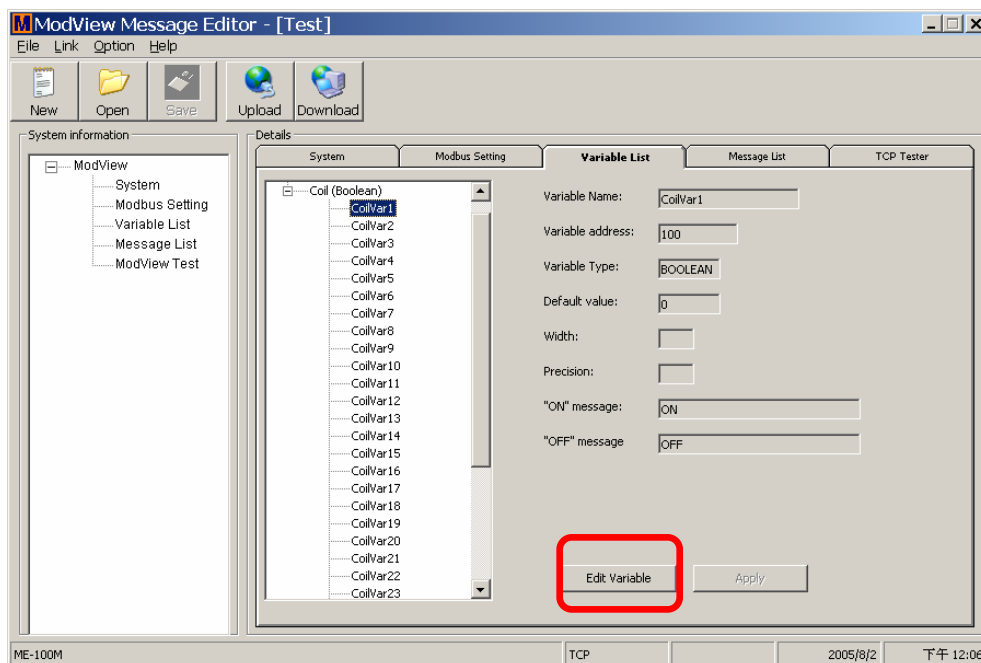
EKAN ME-100M need following configuration files in order to make message play correctly. ModView will automatically generate these files when project is saved or uploaded.

In the following section, we will explain details of those files, and the syntax of message script. You could find message configuration sample under /Sample/ModView_Sample folder.

2-1 Boolean variable configuration

Boolean type variable only have 2 different states: ON and OFF (0 or 1), you could set different message on ON state and OFF state.

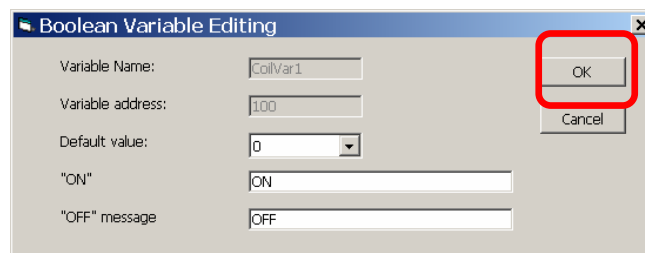
1. Switch to **[Variable List]** page
2. Select the coil variable you want edit.
3. Press **[Edit Variable]** button



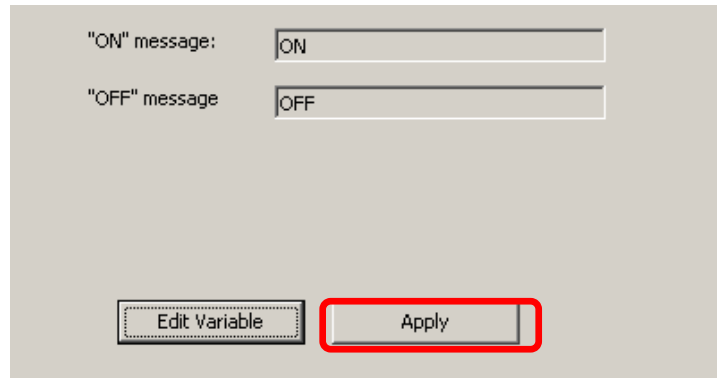
Select default value for the Boolean variable, default is "0"

Input the Boolean "ON" message, and Boolean "OFF" message. The input message length is limited into 48 chars (8x16). If the characters is too long, then the EKAN will cut the extra length.

Press **[OK]** to set the variable, then **[Apply]** button to apply changes.



After you **[Apply]** the changes, the coil variable setting will be changed.
Please repeat previous steps to add more variables.



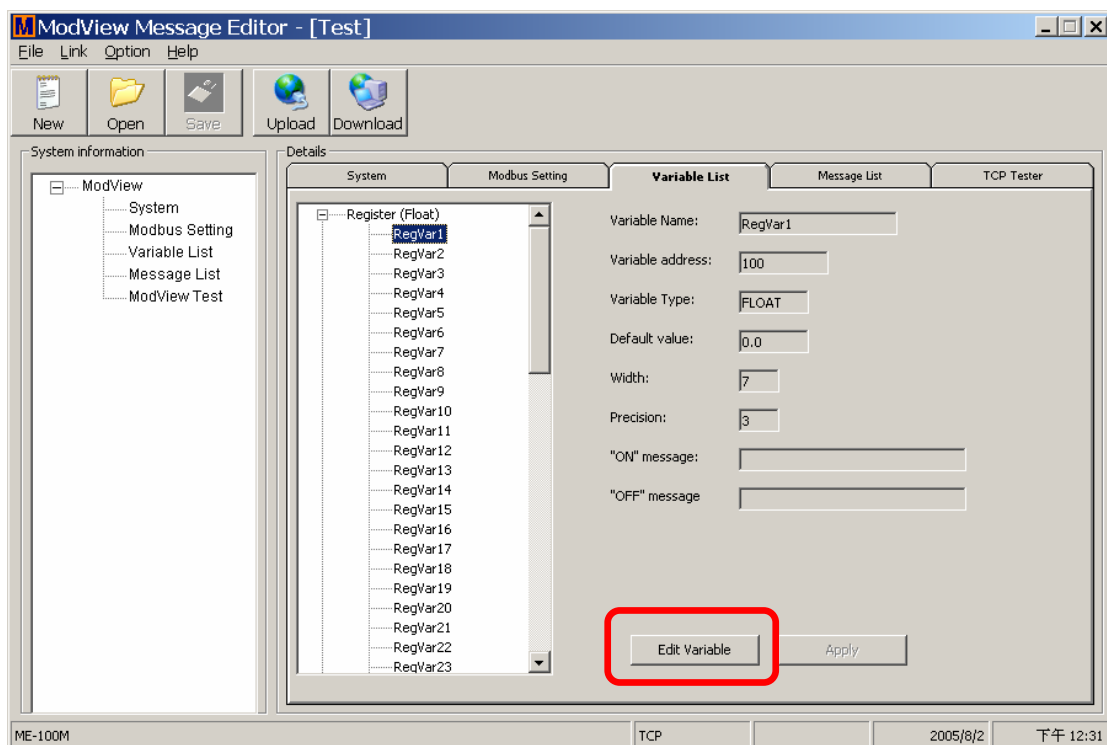
The screenshot shows a user interface for editing messages. It features two input fields: the first is labeled '"ON" message:' and contains the text 'ON'; the second is labeled '"OFF" message' and contains the text 'OFF'. Below these fields are two buttons: 'Edit Variable' and 'Apply'. The 'Apply' button is highlighted with a red rectangular border.

2-2 Float variable configuration

Float variable have 3 parameters need to set, first one is the length of the float variable, 2nd is the precision of variable, and 3rd is the default value of the float variable.

For example, a float variable "140.233" total length is 7, and the precision is 3, and the default value might be "10.101". You could use MODBUS protocol function 16 (10 hex) to update the float variable later.

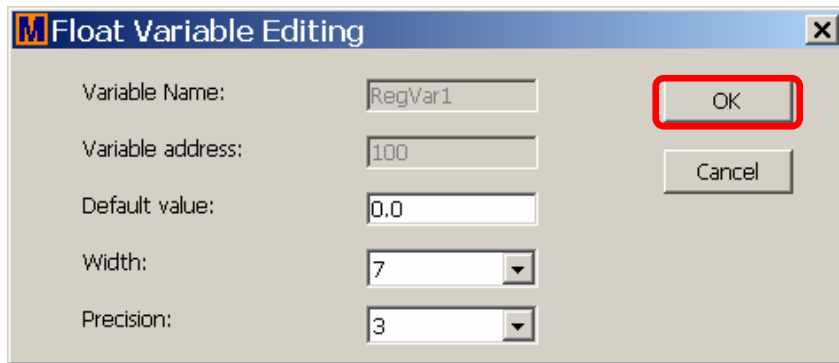
1. Switch to **[Variable List]** page
2. Select the float variable you want edit
3. Press **[Edit Variable]** button



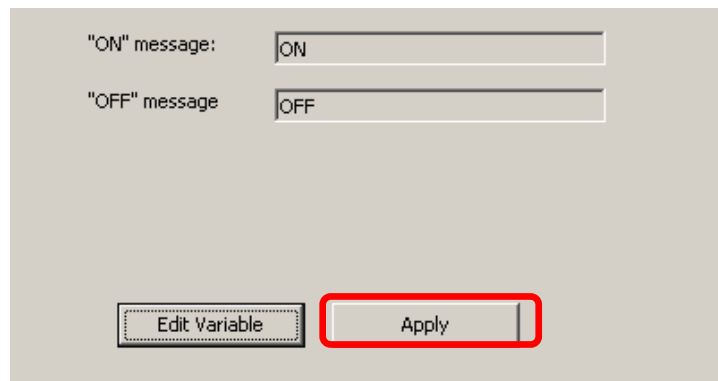
Set value for the float variable, default value is "0". The value set here will be used while EKAN ME-100M power up.

The width of the variable, 7 means the total character length including floating point is 7 characters. And the 3 means there will be 3 characters on the right side of floating point.

Press **[OK]** to set the variable, then press **[Apply]** button to apply changes.



After you **[Apply]** the changes, the coil variable setting will be changed. Please previous steps to add more variable.



Chapter 3 Message editing

ModView uses ModView script for message display. ModView message editor could help you to generate these scripts. And you could modify these scripts that generated by ModView message editor manually. We will show how to use message editor to help you generate these scripts in this chapter.

ModView have 40 regular message groups and 20 emergency message groups, each group could be triggered individually and have own message script blocks. Each message group could have one line or multi lines of message script.

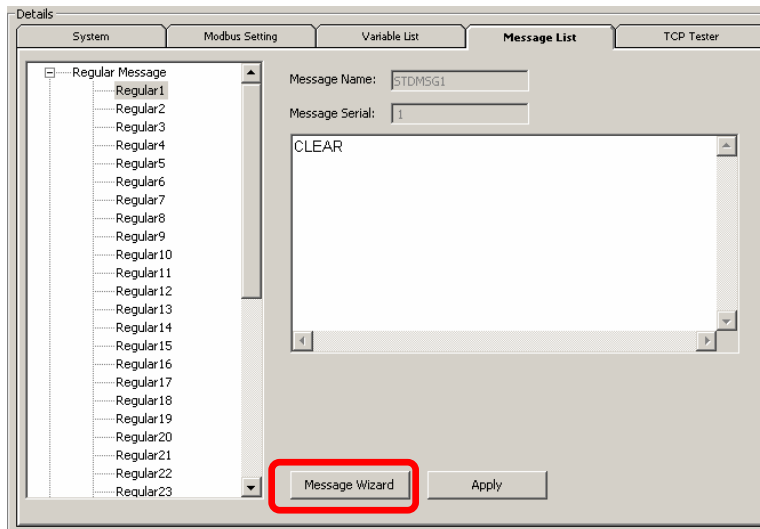
All message using 60 continuous addresses, default start from 00300. First 20 messages is Emergency messages (00300-00319), other 40 messages is regular messages (00320-00359).

In some systems, you should use 00301 to trigger first message instead 00300.

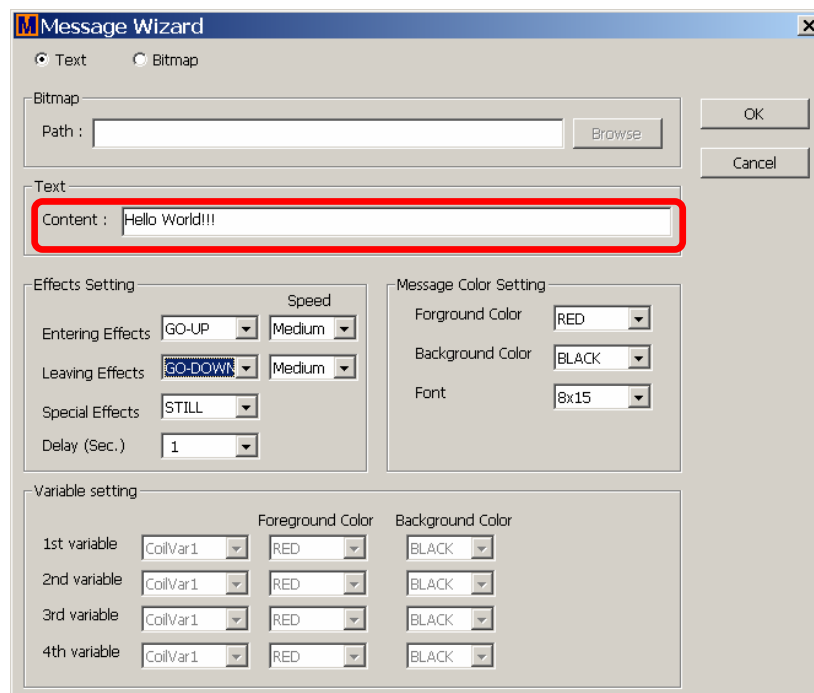
3-1 Message with text

The text based messages are plain text message without any variable inside. I could have special effect like scroll in, scroll-up, blink. Following is a regular meesage sample with using scroll-in effects:

1. Switch to [**Message List**] page
2. Select the regular message number that you want edit.
3. Press [**Message Wizard**] button

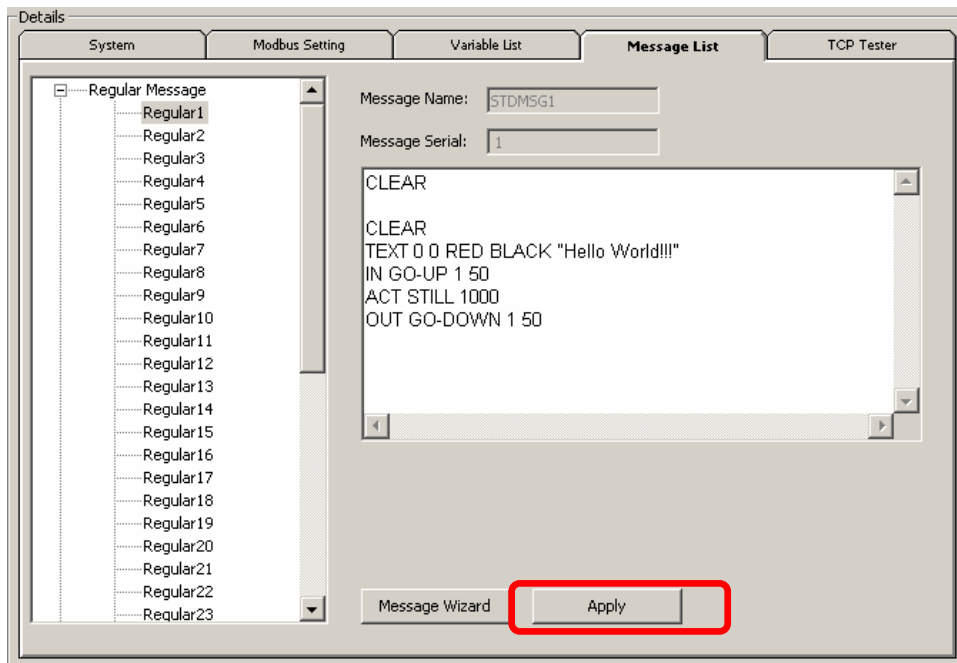


4. Select **[Text]** as message format
5. Input the message text "**Hello World!!!**" in the content field
6. Select **[Effect Setting]** and select **[GO-UP]** and **[GO-DOWN]** as entering and leaving effects
7. Select proper message color in the message color setting
8. Set the message font to "**8x15**"
9. Press **[OK]** button



Then, when you back to main menu, you will see following script in the text box:

```
CLEAR  
CLEAR  
TEXT 0 0 RED BLACK "Hello World!!!"  
IN GO-UP 1 50  
ACT STILL 1000  
OUT GO-DOWN 1 50
```



10. Press **[Apply]** button to save this message.

After you save the message, you could save all projects files by click **[Save]** button at control bar.

Press **[Save]** button at control bar



Then, You could upload the message by click **[Upload]** button

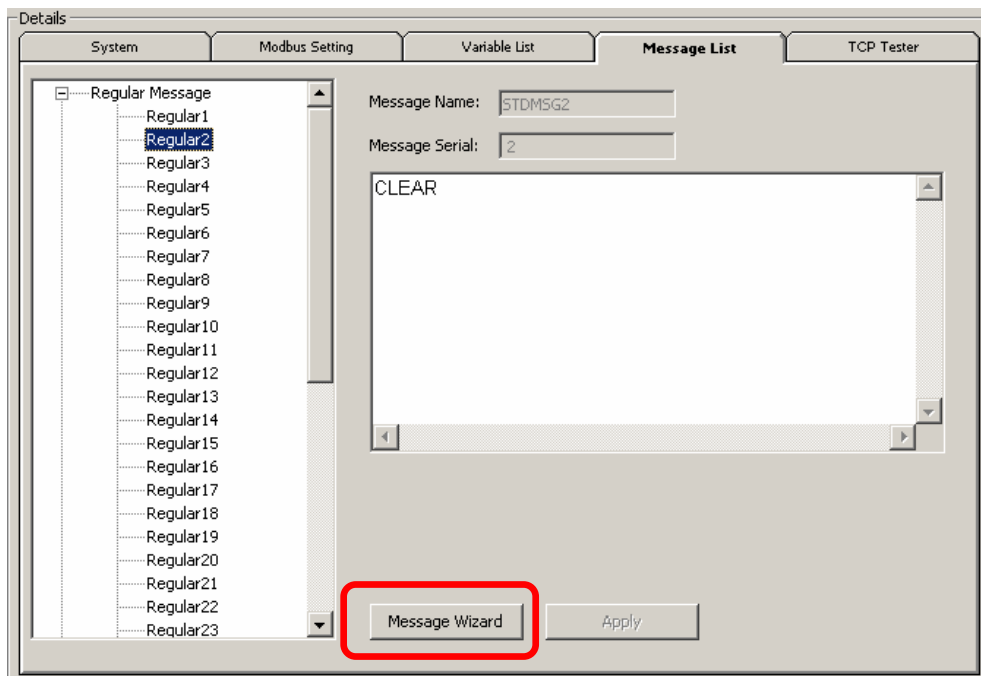


3-2 Message with variable (I/O Data, Date/Time)

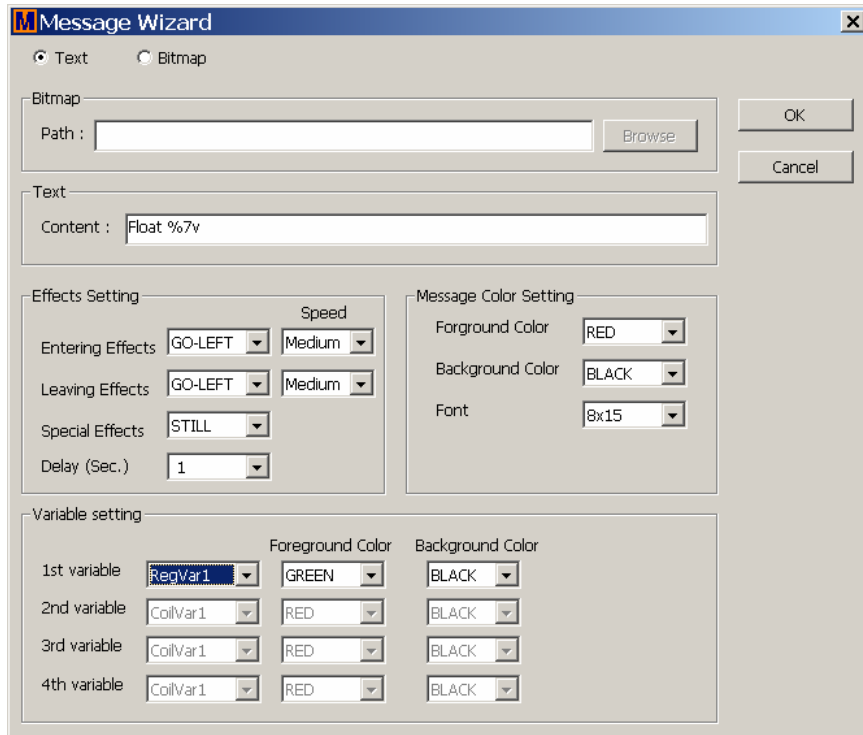
The most important function is the message could have variable inside. For example, a PLC's temperature information could send to ModView to display via MODBUS protocol. Following is a message with float variable inside at MODBSU address 40100.

When you insert variable, **you MUST use %Xv (ex: %7v)** as variable symbol to insert variable. The **X** is the character space for variable.

1. Switch to **[Message List]** page
2. Select the regular message you want edit. We may use regular message 2 now.
3. Press **[Message Wizard]** button



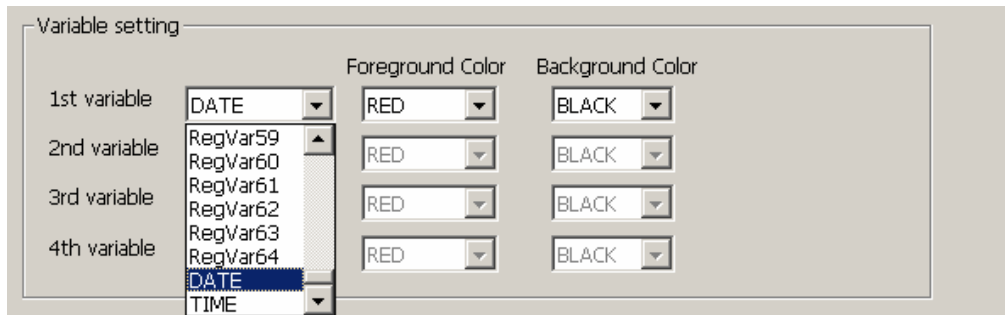
4. Select **[Text]** as message format
5. Input the message text **"Float %7v"** in the content field



6. Select variable name on the 1st variable field, here we use **"RegVar1"** for the Float variable we used in previous section. And select proper color for the variable text.

You could use "Float %7v Float 2 %7v Float 3 %7v Float 4 %7v to load multiple variable"

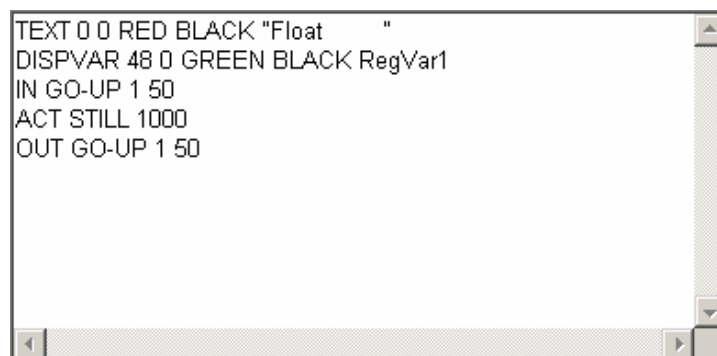
DATE/TIME is a special type of variable, you could use %8v for Time and %10v for Date.



7. Select **[Effect Setting]** and select **[GO-UP]** and **[GO-DOWN]** as entering and leaving effects

8. Select proper message color in the message color setting, we use Green in the sample. 9. Set the message font to **"8x15"**

10. Press **[OK]** button

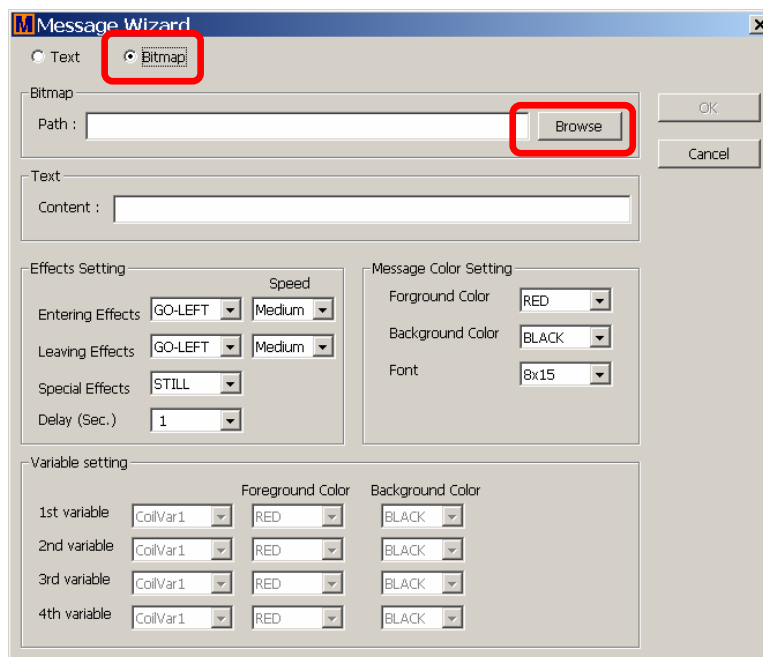


11. Press **[Apply]** button to save this message.

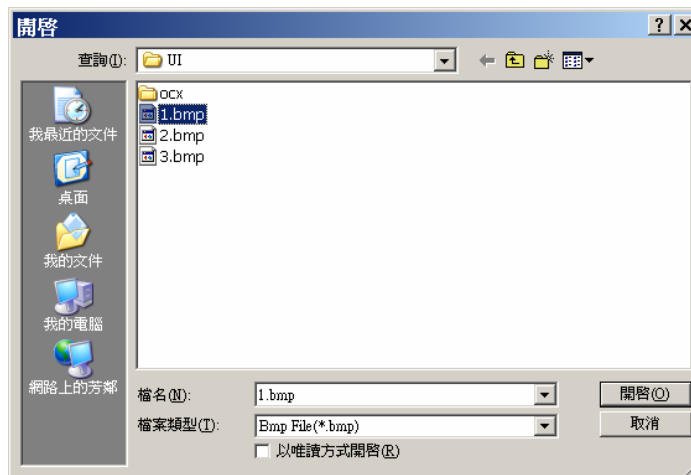
3-3 Message with graphic

You could use 1 bit BMP as message, and upload the BMP file by message editor.

1. Switch to **[Message List]** page
2. Select the regular message you want edit.
3. Press **[Message Wizard]** button

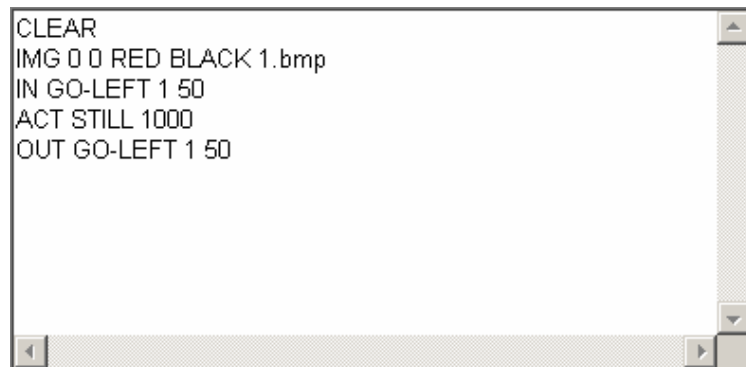


4. Select **[BMP]** as message format
5. Press **[Browse]** to add message



After insert the message, you could see inserted BMP's name, and select the effect want to use.

6. Select proper message color in the message color setting
7. Press **[OK]** button, and the **[Apply]** button to apply message setting.



ModView will upload all BMP files in the project directory. **MAKE SURE YOU PUT ALL BMP FILES IN THE PROJECT DIRECTORY BEFORE UPLOAD.**



Please make sure you put bmp in the project directory

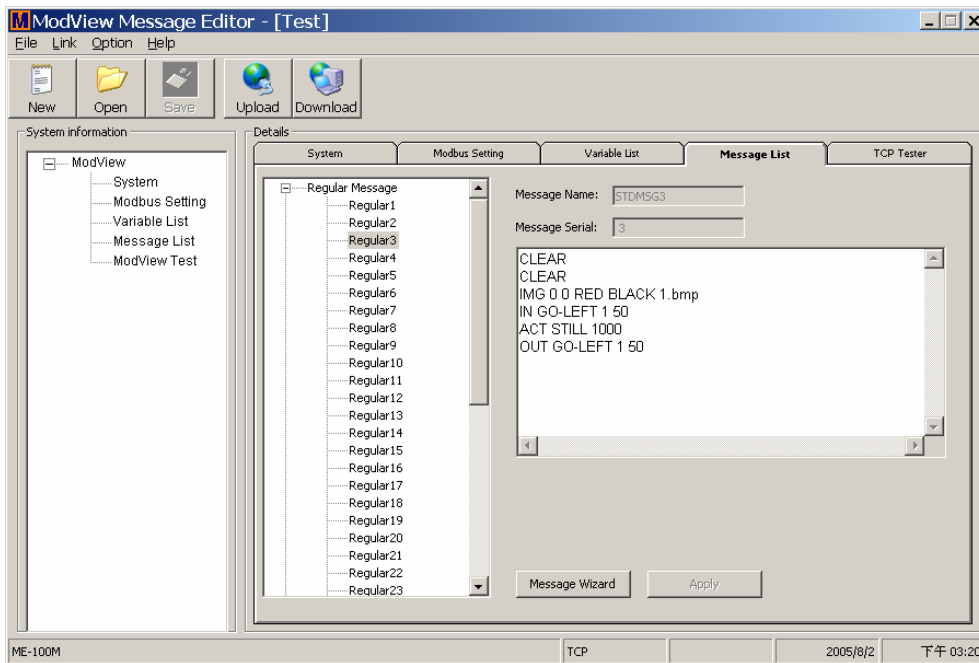
Before you upload the BMP file, you **MUST** copy BMP files into same directory **into current project file directory** (by default it's located at C:\Program Files\eSoftsystem\ModView). Please copy used BMP file into the ModView project directory for message upload. Or will have BMP load error message ("F07 bmp load", "lb 1.bmp" error).

Chapter 4 Upload and Download message project

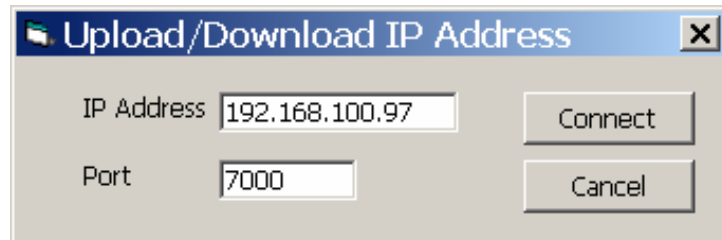
After you finish message editing, you could upload the Message project. Before you upload the project, make sure you have ModView IP address in hand and all the BMP files copied into the program directory

4-1 Upload message files

Press **[Save]** Button to make sure project saved. Then press **[Upload]** button to upload message configuration files.

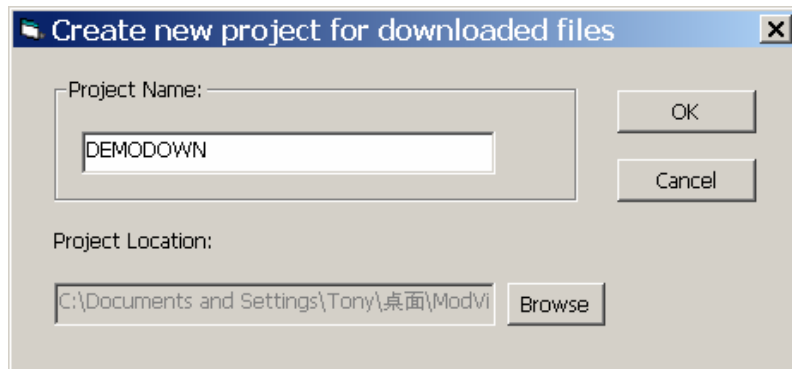


Check the IP address is correct before you upload. The EKAN ME-100M will use port 7000 as default port for message upload.

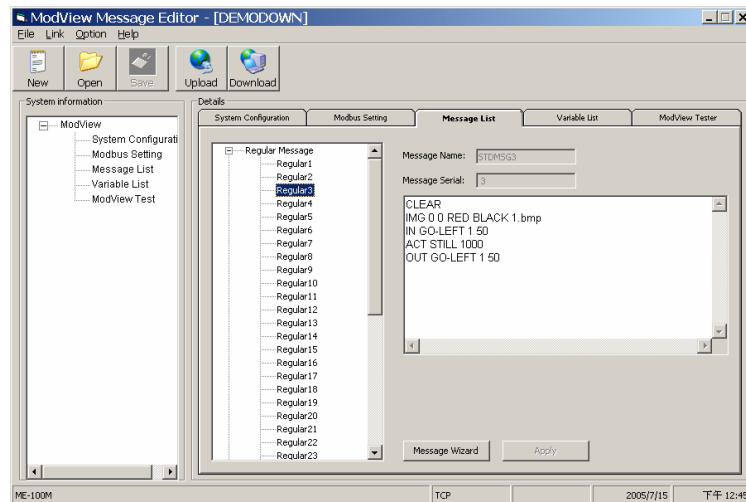


4-2 Download message files

You could press **[Download]** button to download the message configuration setting. Before you download, you must give a specific project name for later save.



Press **[Save]** button to save downloaded message project:



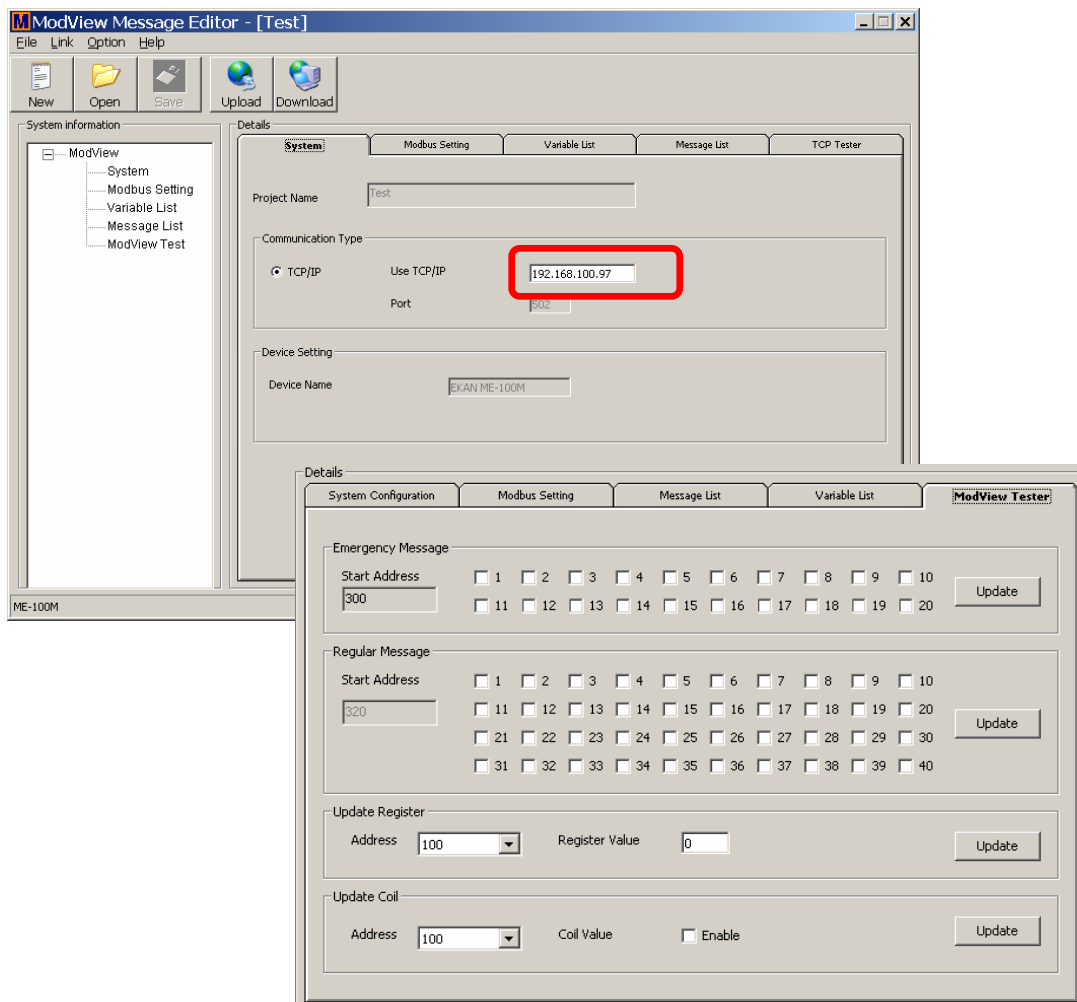
The DEMODOWN.mvp will be saved on the program installation directory.

Chapter 5 Test message display

ModView message editor provide a simple tool used for message display test. You could use this tool to test message display via MODBUS TCP protocol.

First you have setup IP address on the system configuration page:

1. Input the IP address
2. After IP address was set, switch to **[ModView Tester]** Page:

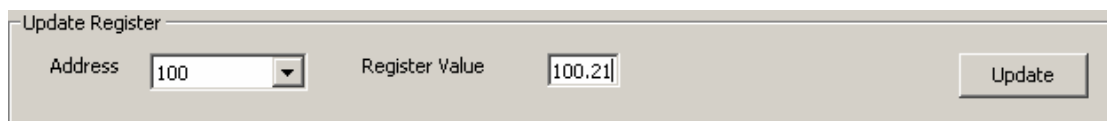


The message address default is 300, means MODBUS address 00300,

And the Regular message will automatically start from 00320 to 00359.

And the register default address is 40100 (Represent by 100 in the screen) and the coil address is 00100.

For example, you could set "100.21" on float variable 1, which use MODBUS address 40100:



The image shows a software dialog box titled "Update Register". It contains two input fields: "Address" with a dropdown menu showing "100" and a small downward arrow, and "Register Value" with a text box containing "100.21". To the right of these fields is a button labeled "Update".

Press **[Update]** button to make it effects. If your network connection has error, the update will be fail with warning dialog box.

Chapter 6 MODBUS protocol Introduction

MODBUS protocol is a widely used open industry control message format. It's could used for read/write single bit data (coil) or multi-byte data (register).

EKAN ME-100M implements MODBUS function 1,3,5,15,16.

6-1 MODBUS frame format (Write Single Coil)

- Write single coil

Following is setting a single digital coil use function 5, Write single coil (0xxxx) for DO:

[Prefix]

Byte 0: Transaction identifier - copied by server - usually 0

Byte 1: Transaction identifier - copied by server - usually 0

Byte 2: Protocol identifier=0

Byte 3: Protocol identifier=0

Byte 4: Length field (upper byte)=0 (since all messages are smaller than 256)

Byte 5: Length field (lower byte)=number of bytes following

Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6
1	2	0	0	0	6

[Request]

- Byte 0: Net ID (Station number)
- Byte 1: FC=05
- Byte 2-3: Reference number, we use address 100 here (Hex 64)
- Byte 4: =FF to turn ON coil, =00 to turn OFF coil
- Byte 5: =00

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
1	5	0	64	255	0

The complete request byte array looks like following:

Prefix						ID	Function	Address		Value	Pad
1	2	0	0	0	6(Hex)	1	5 (Hex)	0	64	FF	0

[Response]

- Byte 0: Net ID (Station number)
- Byte 1: FC=05
- Byte 2-3: Reference number
- Byte 4: =FF to turn ON coil, =00 to turn OFF coil (echoed)
- Byte 5: =00

- Write multiple registers

Setting analog register use function 16 (10 Hex) to write multiple registers (4xxxx) for AO:

6-2 MODBUS frame format (Write Multiple Registers)

[Prefix]

Byte 0: Transaction identifier - copied by server - usually 0

Byte 1: Transaction identifier - copied by server - usually 0

Byte 2: Protocol identifier=0

Byte 3: Protocol identifier=0

Byte 4: Length field (upper byte)=0 (all messages in MODBUS are smaller than 256)

Byte 5: Length field (lower byte)=number of bytes following, in the sample we use 11 bytes (Hex B) for setting single float (2 registers)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
0	0	0	0	0	B

[Request]

Byte 0: Net ID (Station number)

Byte 1: FC=10 (hex)

Byte 2-3: Reference address number

Byte 4-5: Word count (in this case is 2 words for float)

Byte 6: Byte count (B=2 x word count, in this case is B = 4 bytes))

Byte 7-(B+6): Register values (in this case, we used 4 bytes for float. Byte7 to Byte 10)

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7	Byte 8	Byte 9	Byte 10
1	10	0	64	0	2	4	00	00	00	00

The complete request byte array looks like following:

Prefix						ID	Function	Address		Word		Byte	Float			
0	0	0	0	0	B (Hex)	1	10 (Hex)	0	64	0	2	4	0	0	0	0

[Response]

Byte 0: Net ID (Station number)

Byte 1: FC=10 (hex)

Byte 2-3: Reference number

Byte 4-5: Word count

Byte 0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5
1	10	0	64	0	2

The complete response byte array will be:

Prefix						ID	Function	Address		Word	
0	0	0	0	0	6	1	10 (hex)	0	64 (Hex)	0	2

On the demo project, the following address is used:

	Start Address	End Address	Total Number	Type	Using MODBUS function Number
Emergency message	00300	00319	20	Boolean	5
Regular Message	00320	00359	40	Boolean	5
Float variable	40100	40126	64	Float	16
Boolean Variable	00100	00131	32	Boolean	5

Chapter 7 Connection with Visual Basic Client (MODBUS TCP)

There is a Visual Basic sample using MODBUS TCP as controlling protocol under CD-ROM's sample folder. You could use Visual Basic 6 to open the project files. All samples comes with source code and project file.

7-1 Trigger Regular message

The regular message will be automatically play when message file finished upload. If you didn't trigger any specific regular or emergency message, the system will automatically display all message on the message play list.

If you want to make a new play list, you could trigger the message you need to make a new play list.

In the sample, regular message address is start from 320 (First 20 messages, from 300 to 319 is reserved for emergency messages). You could check those check box that named after R (for example, R01 means regular message 1) to trigger specific regular message.

7-2 Update variables

There are 2 types of variable in the ModView display:

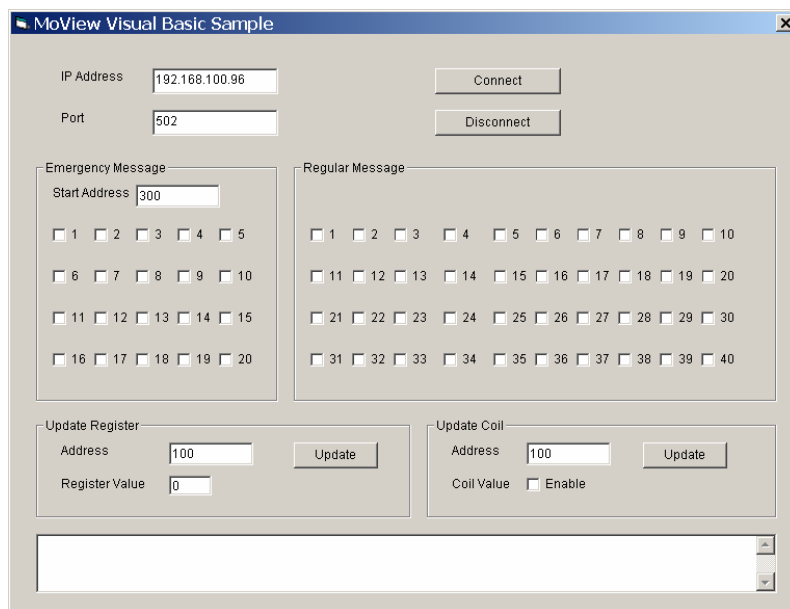
1. Float type variable usually using for analog data display
2. Boolean type variable, usually using for digital data display.

Messages in the ModView are usually combining with text and variable. Text usually for static description and the variables are dynamic data that will be updated from the MODBUS device.

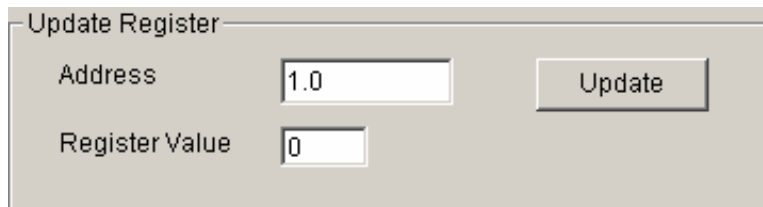
- Update float variable

Float type variable using 2 registers in the ModView (DWORD, 32 bit), usually represent the analog type data. Because float use 2 registers, The address will be all odd or even numbers. For example, if address start from 100, the next address will be 102, and third address will be 104...so on.

You could input the address of the variable, and give a float value, then press **[Update]** button to update the variable in the message. In the sample, the analog variable start from 40100, using MODBUS function call 16 (Hex 10) to modify the float variable.



1. Input the variable address (100 in the sample)
2. Input the variable value (1.0, for example)
3. Press **[Update]** command button



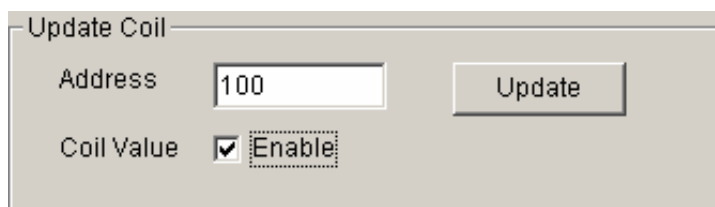
The screenshot shows a dialog box titled "Update Register". It has two input fields: "Address" containing the text "1.0" and "Register Value" containing the text "0". To the right of the "Address" field is a button labeled "Update".

After you update the register value, you will see the new value shows in the message.

- Update Boolean variable

The coil value is Boolean type variable, usually represent the digital I/O value. You could input the address of the variable, and give a Boolean value. Then press **[Update]** button to update the Boolean variable in the message.

1. Input the variable address (100 in the sample)
2. Click the check box to select true or false
3. Press **[Update]** command button



The screenshot shows a dialog box titled "Update Coil". It has two input fields: "Address" containing the text "100" and "Coil Value" containing a checked checkbox and the text "Enable". To the right of the "Address" field is a button labeled "Update".

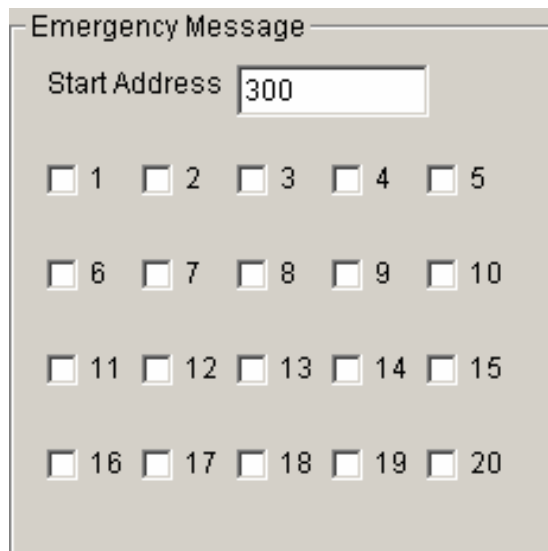
After you update the coil value, you will see the new value shows in the message.

7-3 Trigger Emergency message

Emergency messages are all high priority messages; it will override the any original message play list. Once the emergency message was triggered, the emergency messages will continuous play until it stopped.

The message format as follow:

1. Input the variable address (300 in the sample)
2. Click the chick box to select true or false



Emergency Message

Start Address

1 2 3 4 5

6 7 8 9 10

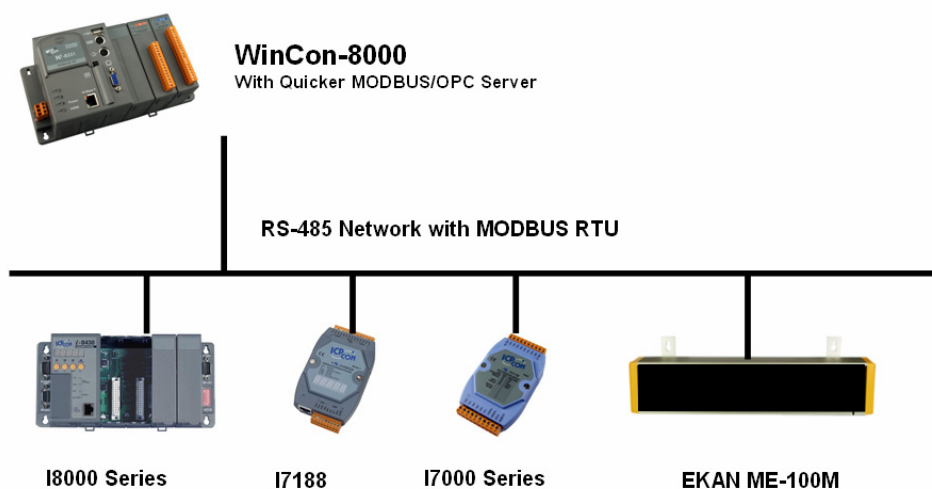
11 12 13 14 15

16 17 18 19 20

Once the emergency message is triggered, it will play the message until it switch off. If multiple Messages were triggered, you must switch all emergency messages off in order to switch back original message play list.

Chapter 8 Connect with OPC server

The OPC means “OLE for processing control”, an open standard provides open connectivity between different systems. Most OPC server could use MODBUS protocol to control device and update device data. OPC application could control ModView via OPC Server (like free ICPDAS NAPOPC server, here we use OPCsystems OPC server for screen shot).



8-1 Tag configuration

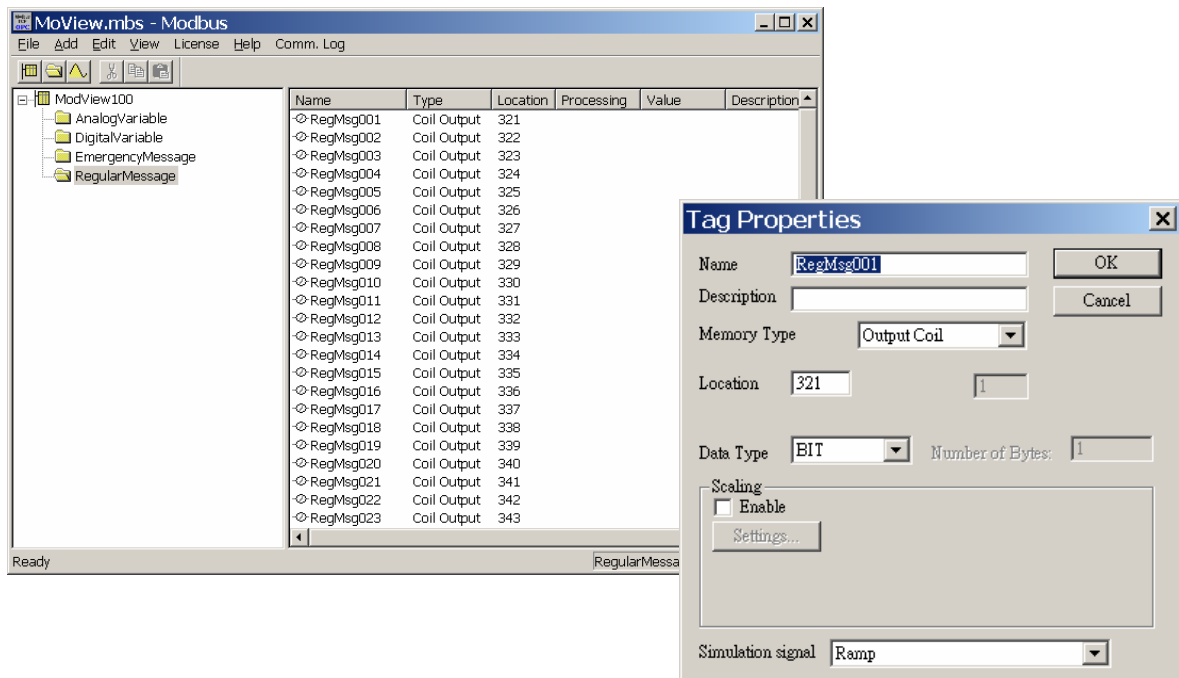
OPC client use tags to mapping device data instead of address. So you must establish the address and tag relationship in the OPC server. OPC are open standard for the device manufacture and the 3rd party software vender. So the interface of the OPC server might looks slight different with each other, but the configuration steps should be very similar.

The OPC tag naming is hierarchy structure. For example, the “IO.Analog.Tag0” may direct to analog IO device’s first tag. OPC server will transfer the tag into MODBUS address, and then execute the I/O command.

8-2 Set regular message coil

The message control function of ModView is controlled by 40 continuous coil. Here we use register 00320 to 00359 to set the register. Following is the complete setting step for setting function register.

Because OPC is used 1 based calculation, so it will minus 1 for all MODBUS address, You must use 321 as start address instead 320.

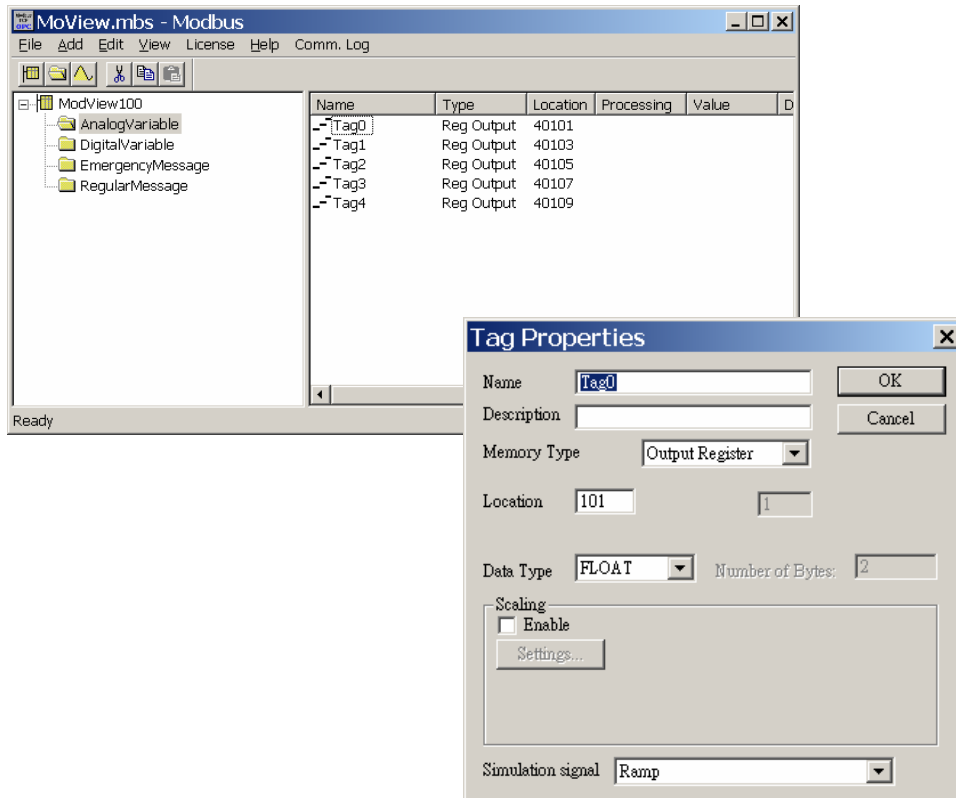


Then create tags for regular message in the folder.

1. Create new tags for coil 00321
2. Give tag name
3. Set the memory type, Output Coil
4. Set the address location
5. Set the data type to Bit
6. Press OK

8-3 Set float variable register

ModView LED could have 64 register type variables. But you could just set the variable you need in the OPC server.

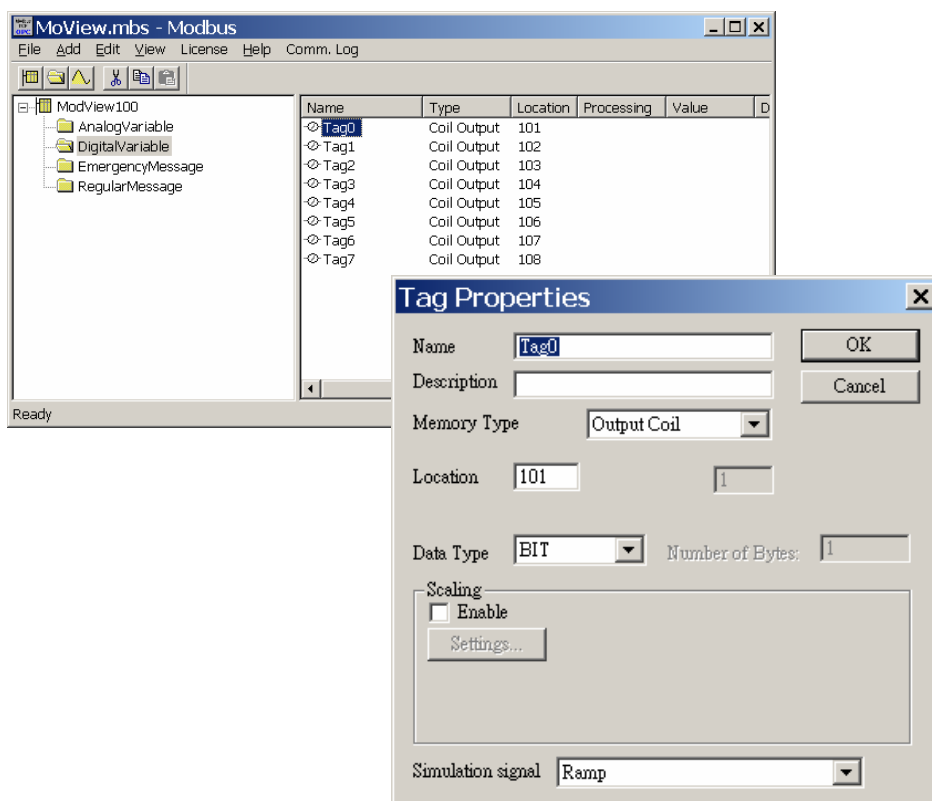


1. Create new tags for register 101 (40100 for MODBUS definition)
2. Give tag name
3. Set the memory type, Register
4. Set the address location, 101
5. Set the data type to **Float**
6. Press **[OK]** button

8-4 Set Boolean variable coil

The coil variable is Boolean type variable, which means it only could have "True" or "False" value. This type of variable usually map to ON/OFF switch or digital I/O device. The coil variable is start from 00100 in default configuration.

The Boolean variable of ModView is controlled by 32 continuous coils. Following is the complete setting step for setting coil variable.

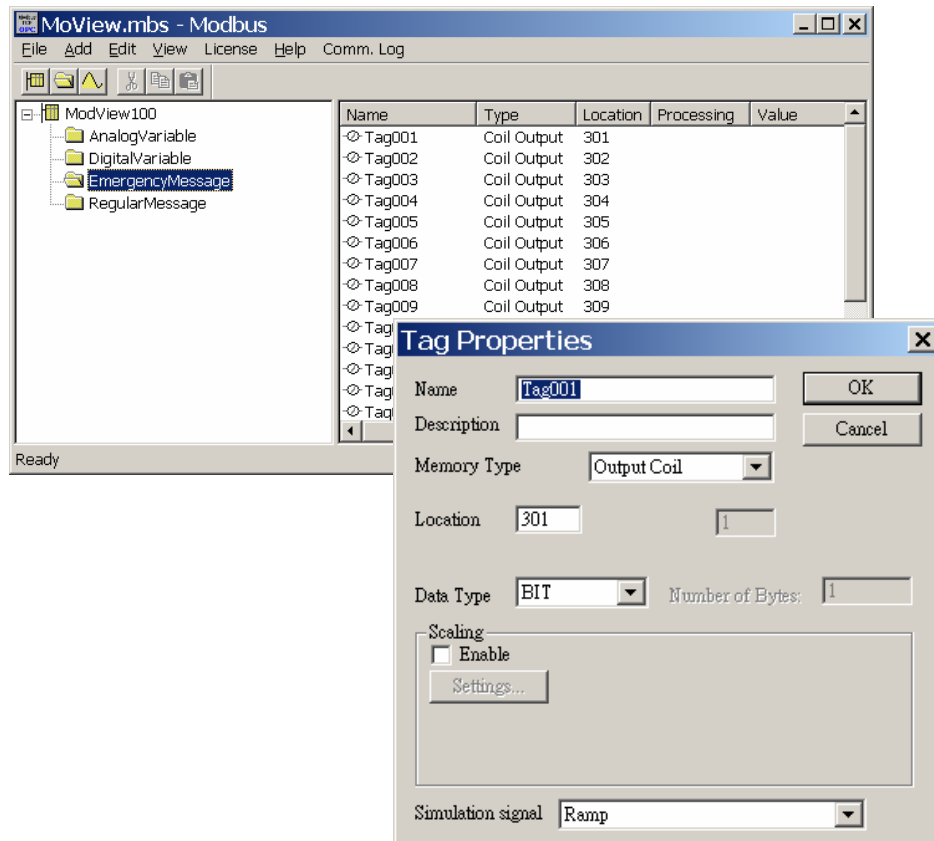


1. Create new tags for coil 00101
2. Give tag name
3. Set the memory type, Output Coil
4. Set the address location
5. Set the data type to **Bit**
6. Press **[OK]** button

8-5 Set emergency message coil

ModView have 20 emergency address, we use 2 emergency messages here.
The tag definition is

Then create tags for emergency message in the folder.

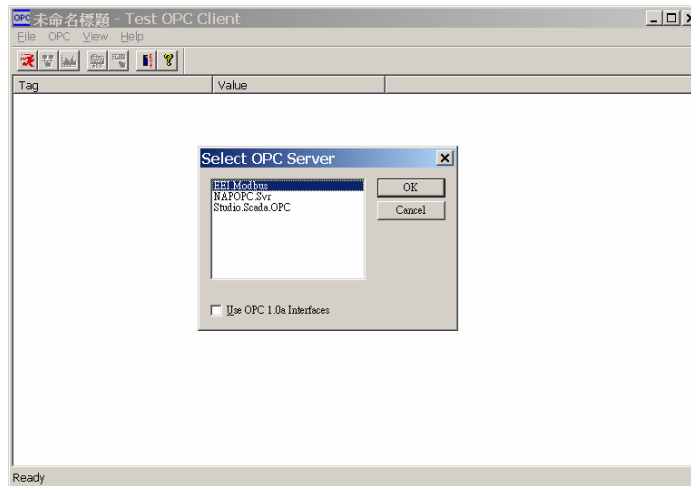


1. Create new tags for coil 00301
2. Give tag name
3. Set the memory type, Output Coil
4. Set the address location
5. Set the data type to **Bit**
6. Press **[OK]** button

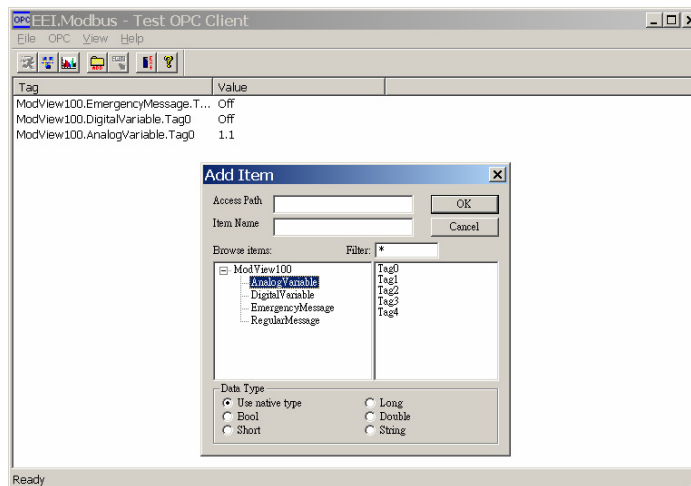
8-6 Create OPC Client

The OPC client communicates with server by OPC protocol. There are many 3rd party software vender provides OPC component for programmer to create OPC clients. Major brand HMI also provides OPC component for communicating with OPC server.

For OPC client with fast scan rate (like 20ms), you could modify the RTU timeout setting in the configuration page. **EKAN ME-100M default factory refresh rate (RTU timeout) setting is 50ms.** You should use 10ms on EKAN ME-100M on refresh rate 20ms. Here we use the default client comes with OPC server:

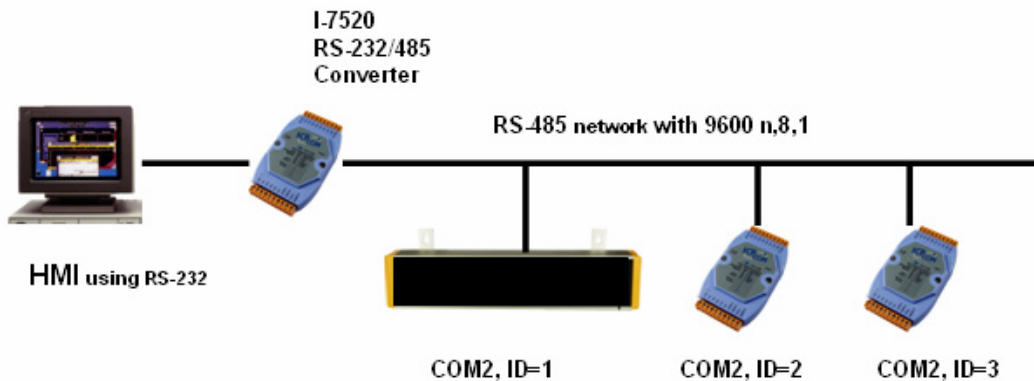


1. Connect OPC server
- 2, Select tags that need to use, and set the value if needed.

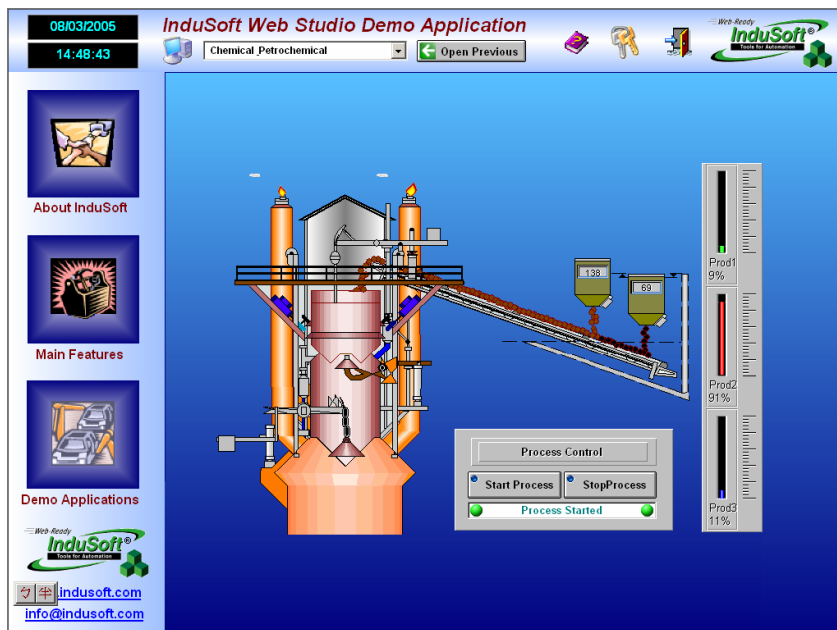


Chapter 9 Connect with InduSoft (HMI/SCADA)

Indusoft is monitor/control software for industry automation. InduSoft could easily integrate application and the help you quickly build up a human machine interface application. ICPDAS WinCon-8000 also provides run-time environment for Indusoft software.

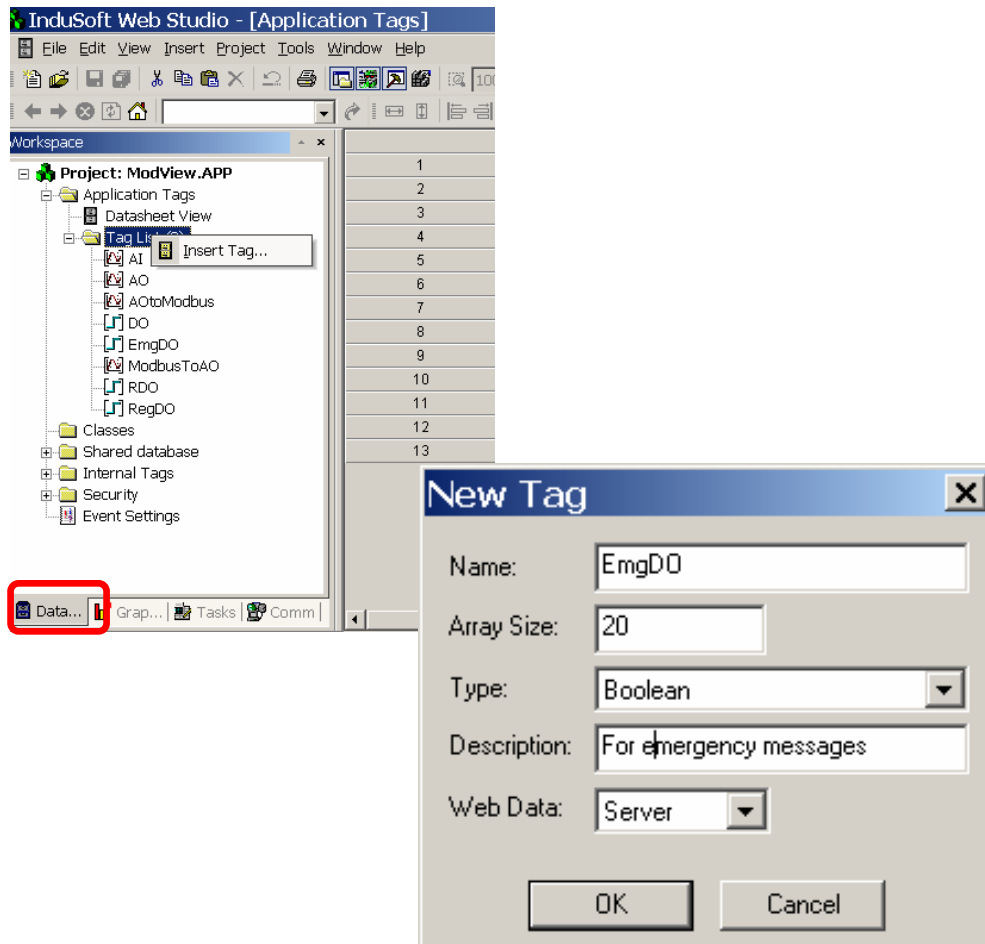


Like most HMI software, Indusoft natively support MODBUS TCP/RTU protocol. You could define tags and mapping to MODBUS slave like EKAN ME-100M. Following is a short introduction about how to use indusoft to connect EKAN ME-100M. You could find the sample in the



9-1 Create Indusoft project and create tags

First we create a empty project and create required tags for control EKAN ME-100M. We create DO for message and DO control. We create a Emergency Message DO variable array, size is 20 Boolean variables.

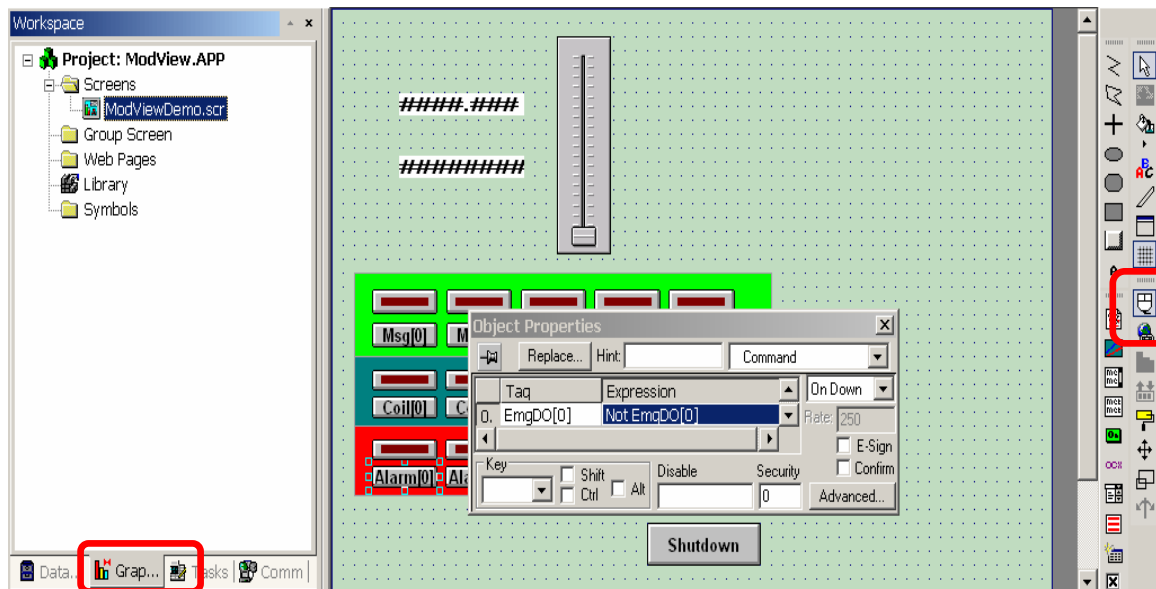


After you create DO, you could use for later user interface mapping. You could follow same steps to add more tags for trigger regular messages.

9-2 Mapping variables to button

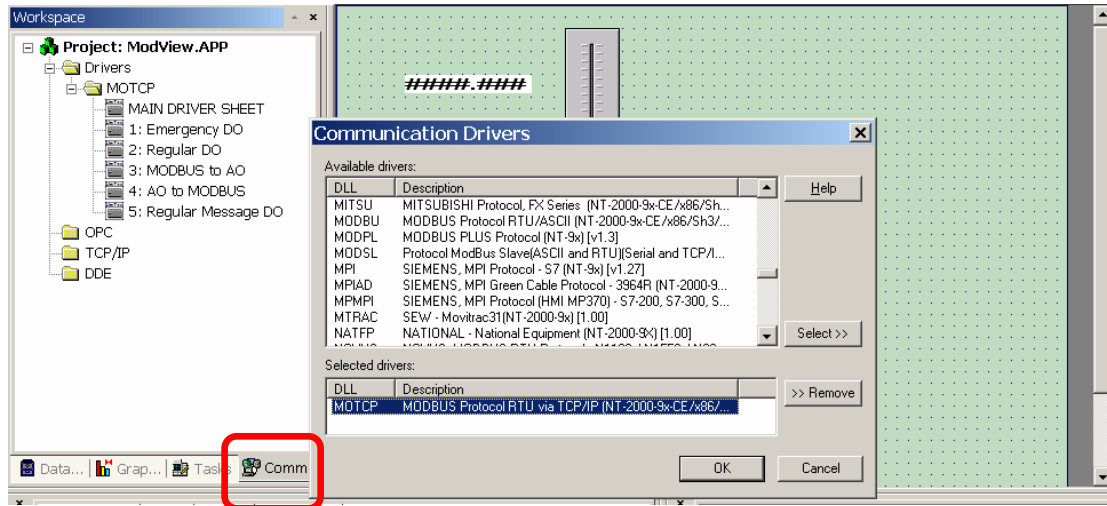
Indusoft provides variety of user interface components to interactive with user. We use a button here to trigger messages. First we use an empty template to put buttons on, and then set the action property of button.

1. Switch to screen view, then add button in the Indusoft workspace. And use the mouse icon to change the dynamic property to **[On Down]**. Mapping tag to EmgDO[0].
2. When press, inverse current state by using expression **Not EmgDO[0]**



9-3 Mapping variables with MODBUS TCP

The third part is to mapping button to the MODBUS TCP address, we use following setting for Emergency DO setting. First we switch to comm page. Then select the MOTCP (MODBUS RTU protocol via MODBUS TCP)



Next, we create mapping between Tag and MODBUS address.

Description: Increase priority

Read Trigger: Enable Read when Idle: Read Completed: Read Status:

Write Trigger: Enable Write on Tag Change: Write Completed: Write Status:

Station: Header: Min: Max:

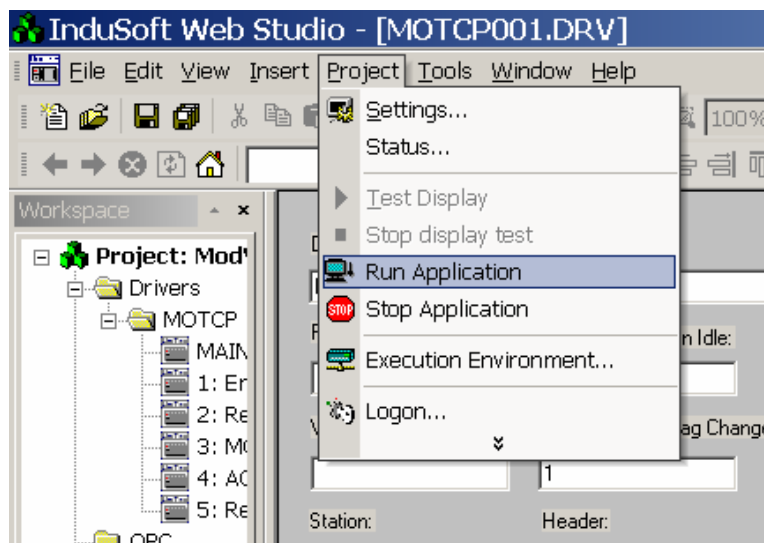
	Tag Name	Address
1	EmgDO[0]	301
2	EmgDO[1]	302
3	EmgDO[2]	303
4	EmgDO[3]	304
5	EmgDO[4]	305
6	EmgDO[5]	306

Station setting: 192.168.100.95:502:1

Header use (MODBUS function 5): 0x:0

Enable write on tag change: 1

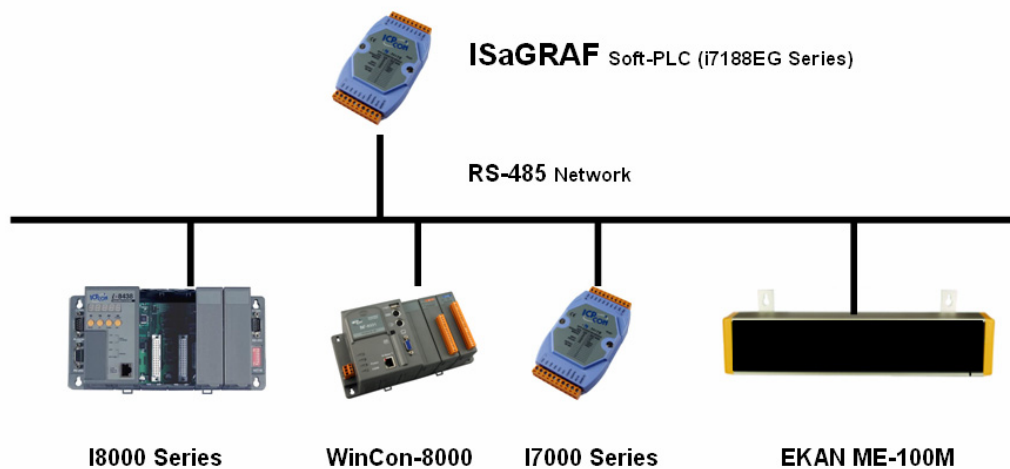
After, setting, You could run the sample application.



You could find the complete sample under **sample/InduSoft** folder in the CD-ROM.

Chapter 10 Connect with IsaGRAF (SoftPLC)

ISaGRAF is provides popular standard Industry Automation control programming method. Includes SD, FBD, SI, IL...programming language. You could use ISaGRAF to generate controlling firmware, and then download to the 7188EG or similar controlling firmware.

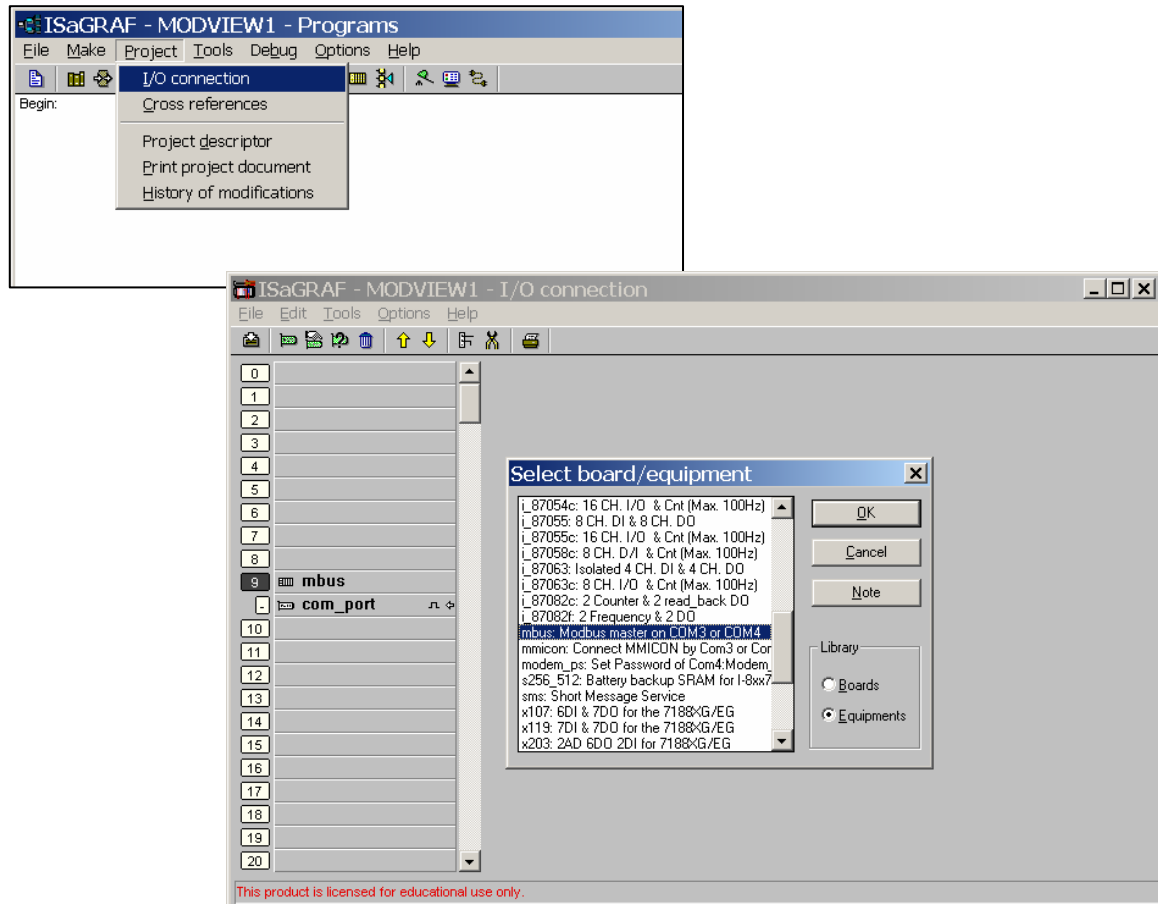


I7188 EG could be used as MODBUS master device to control MODBUS slave device on the RS-485 network. Client PC could act as monitor console linked via Ethernet.

Detailed information about ISaGRAF and I7188EG module, you could refer to ICPDAS web site for more information.

10-1 Set 7188EG as MODBUS master device

You could set 7188 EG as MODBUS by using the mbus component. You must install the "Mbus" library before you adding the MODBUS support. You could refer 7188 EG manual or ISaGRAF manual for more information.

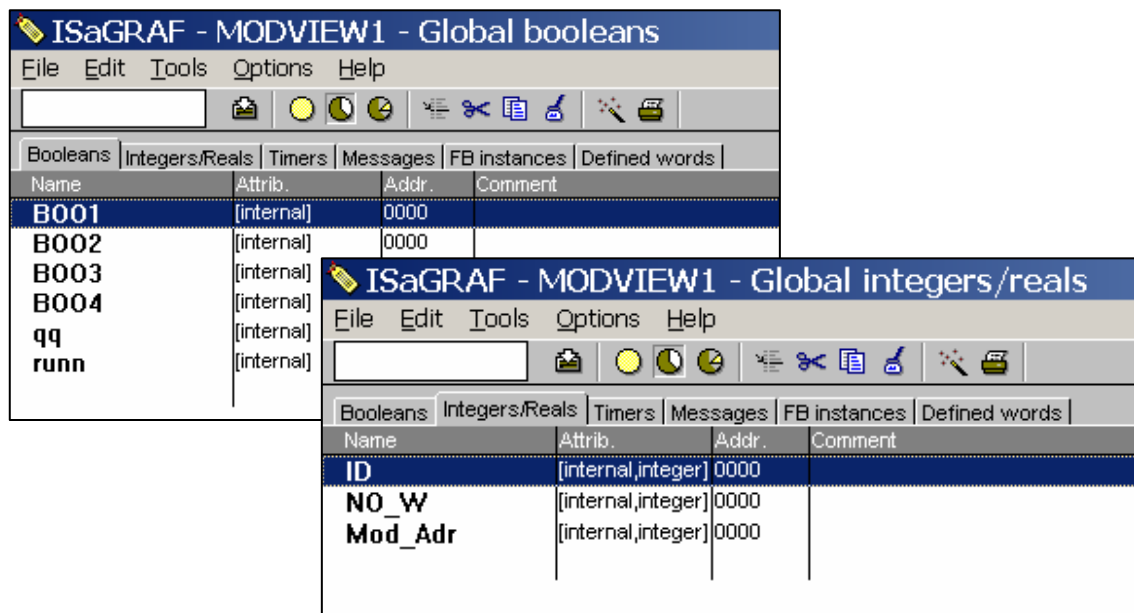


After you set the I7188EG as MODBUS master device, You could using MODBUS related function block to control MODBUS slave like EKAN ME-100M.

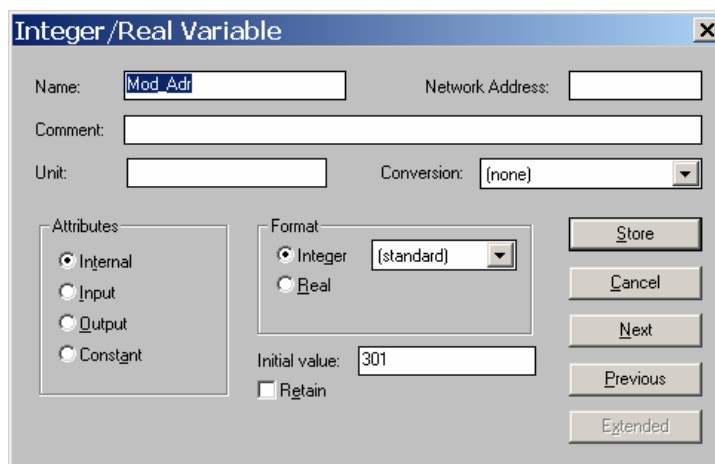
10-2 Using function block

We use ISaGRAF to control the coil variable and trigger messages on the EKAN ME-100M. We have to create 2 different types of variable to mapping with MODBUS function block.

- Boolean variable for the trigger message, and control function block's state
- Integer variables for MODBUS address and ID setting.



Following is address setting examples:



10-3 MODBUS function block on ISaGRAF

There are 4 MODBUS function blocks support by Mbus library:

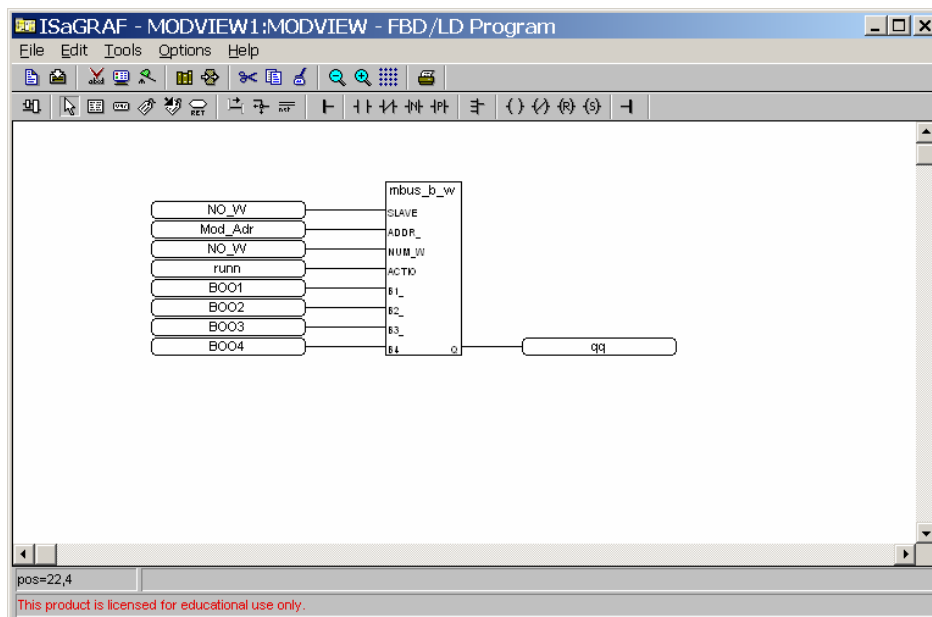
Mbus_b_r : Read 8 bits (Boolean) from MODBUS device.

Mbus_b_w: Writes 1 to 4 bits to MODBUS device

Mbus_n_r : Read 8 words (short integers) from MODBUS device

Mbus_n_w: Writes 1 to 4 words to MODBUS device

Following is a sample using Mbus_b_w to write 4 bits on address 00301. You could compiler this program, and then download to I7188EG firmware to see the effects.



This sample is under sample/ISaGRAF folder, you could refer to sample and 7188EG information on ICPDAS website.

About eSoftsystem Corp.

eSoftsystem Technology Corp. is the most innovative embedded solution provider. It has built up a team of world experts in embedded software and hardware systems to provide customer high-performance and high-quality embedded solution product and services.