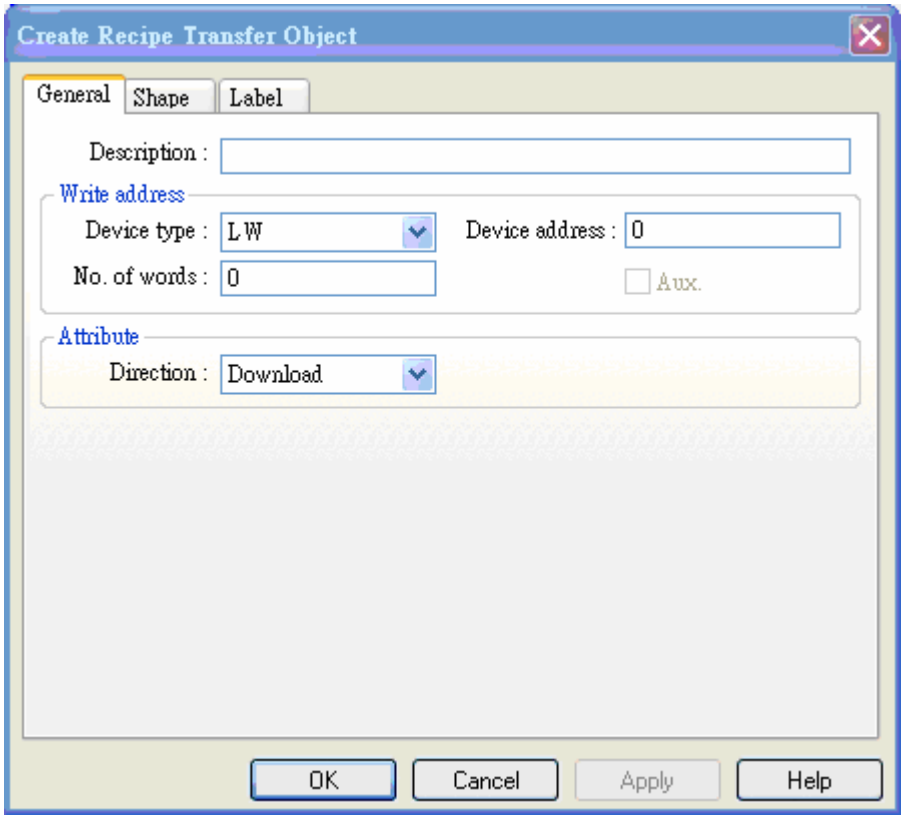
Chapter 8. Recipe Transfer

The Recipe Transfer part activates the transfer of a block of contiguous registers from the HMI to the PLC or from the PLC to the HMI. HMI storage address is determined by an internal word. For MT500, 64K is selected to store the recipe data.

8.1 Procedure to create a Recipe Transfer Part

1. Click the Recipe Transfer Tool or select Recipe Transfer from the Parts menu. 

2. Fill in General Tab items:



Description: A reference name that you assign to the Recipe Transfer. (not displayed)
Write Address: Word that begins the block of registers to write or receive upload from the PLC.
No. of words : How many registers are transferred.

Direction:
Download : Moves values from the HMI retentive memory to the PLC.
Save: Transfers values from the controller to the HMI retentive memory area.
3. Go to Shape Tab: Select Shape or Bitmap of the button to activate the transfer.
4. Go to Label Tab: Fill in fields to denote states, if desired.
5. Click OK to position the part and resize it.

8.2 Recipe memory

Recipe memory card should be chose for using recipe memory. The recipe memory resides in battery backed SRAM. The memory contents are preserved for at least half year after power off. The battery is recharged whenever the system is powered. The total size of recipe memory is 64K words.

There are two ways to represent the recipe memory: “RW” represents absolute address,”RW1” represents index address and the number of words you specify in LW9000 offsets an index address from its indicated address. For example if (LW9000)= 50, an RWI 0 index address points to the address with data 50. If we change (LW9000)=51, an RWI 0 index address points to the address with data 51. The table shows as below:

Address	Data
RW50	0X1111
RW51	0X2222
RW52	0X3333
RW53	0X4444
RW54	0X5555
RW55	0X6666
RW56	0X7777
....	....

LW9000 = 50	
Address	Data
RWI0	0X1111
RWI1	0X2222
RWI2	0X3333
RWI3	0X4444
RWI4	0X5555
RWI5	0X6666
....	....

LW9000 = 51	
Address	Data
RWI0	0X2222
RWI1	0X3333
RWI2	0X4444
RWI3	0X5555
RWI4	0X6666
RWI5	0X7777
....	....

Basing on the concept above, here we take an example:

We create a project and select PLC type as [MODBUS RTU]. We upload 5 consecutive numbers starting at address 200 from device type as 4x to RW300 address of recipe memory and the consequence explains as below:

Address	Data
4x200	'A','B'
4x201	'C','D'
4x202	'E','F'
4x203	'G','H'
4x204	'I','J'

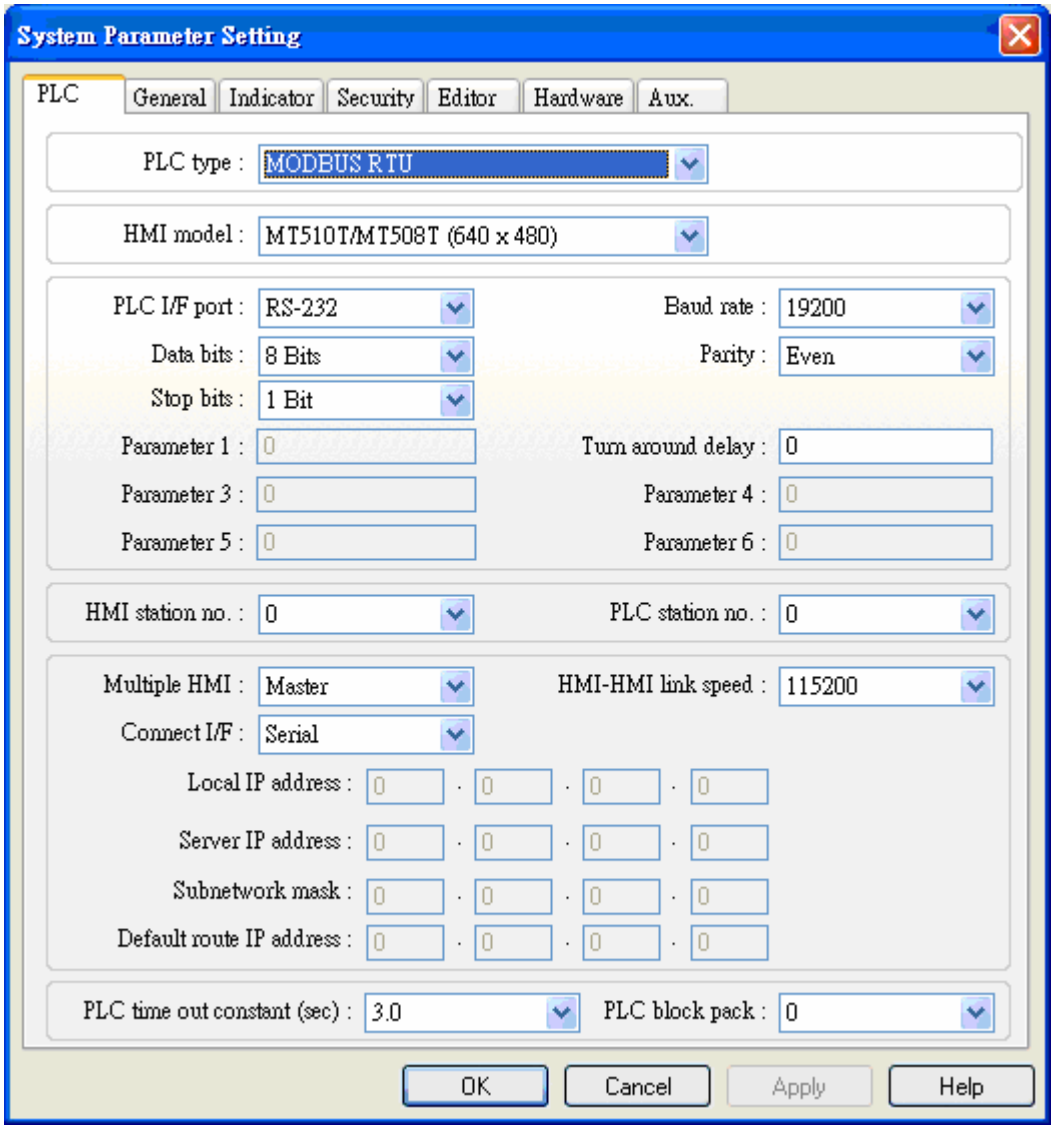
Before Upload

Address	Data
RW300	'B','B'
RW301	'C','C'
RW302	'D','D'
RW303	'E','E'
RW304	'F','F'

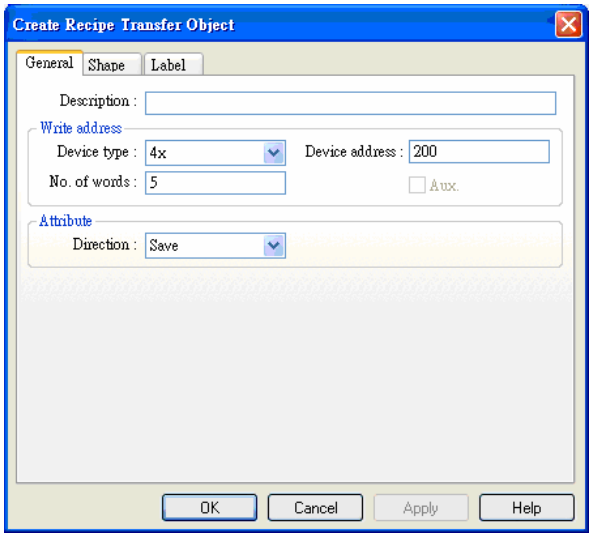
After Upload

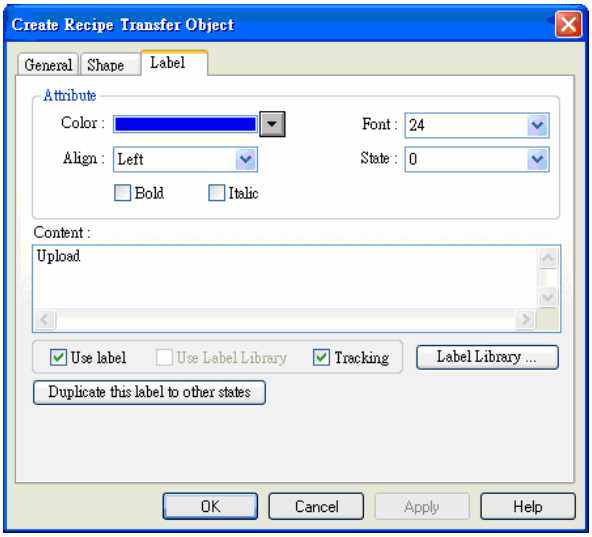
Address	Data
RW300	'A','B'
RW301	'C','D'
RW302	'E','F'
RW303	'G','H'
RW304	'I','J'

We create a new project and choose PLC type as [MODBUS RTU]:

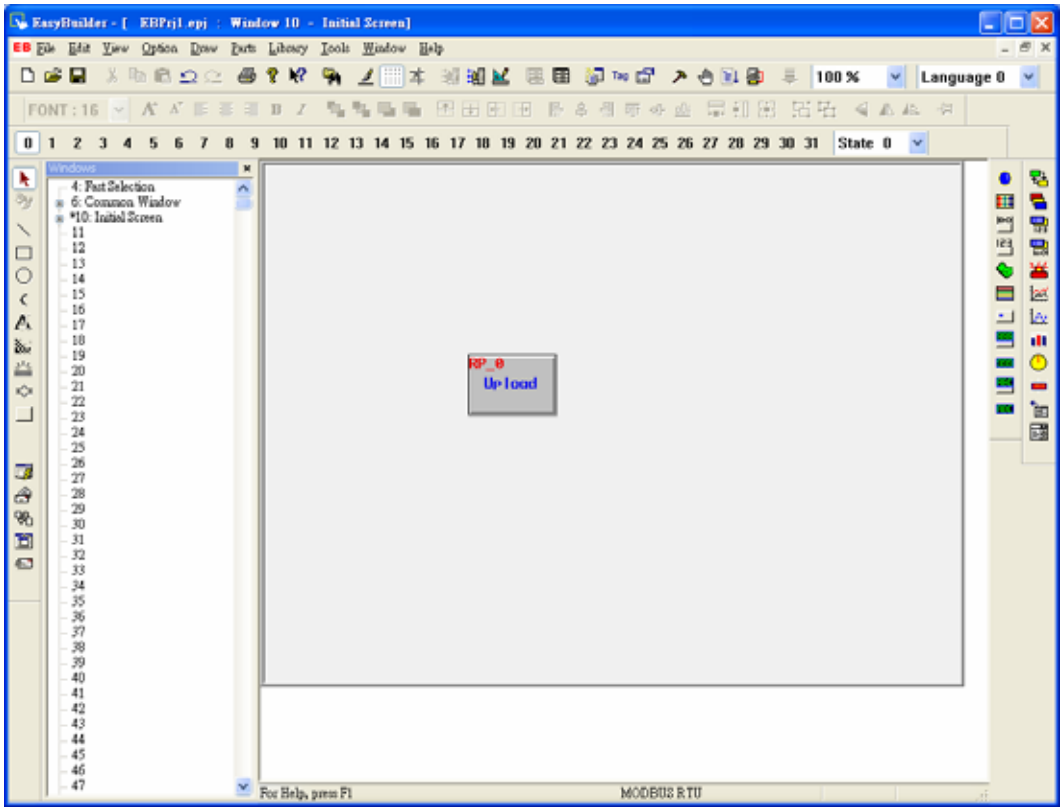


First of all, we add a Recipe Transfer object with device type=4x, Device address=200, No. of words=5, Direction is “Save” and content as “ Upload” :



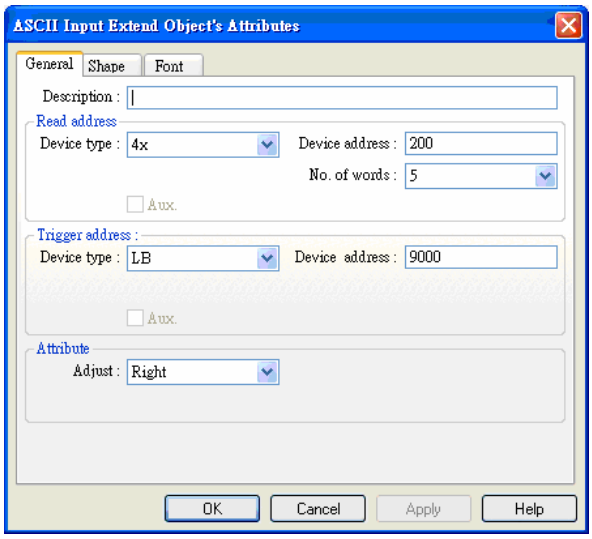


We place it on the window.

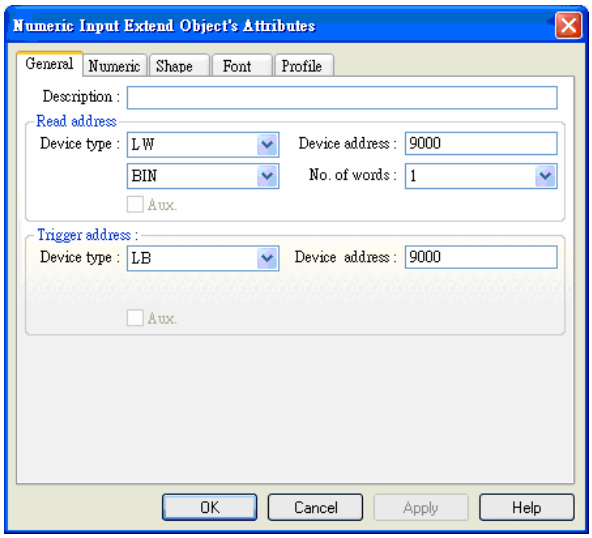


The recipe transfer object is done. However, the object just points out to transfer 5 numbers starting from the address of 4x200 to recipe memory but doesn't indicate the specific locations. That's why we need LW9000. We set LW9000 as 300. Press transfer button and then we transfer the 5 continuous words followed by the address of 4x200 to 5 continuous locations after the recipe memory RW300. If we would like to 5 continuous data starting from 4x200 to RW100, we just need to set LW9000 as 100.

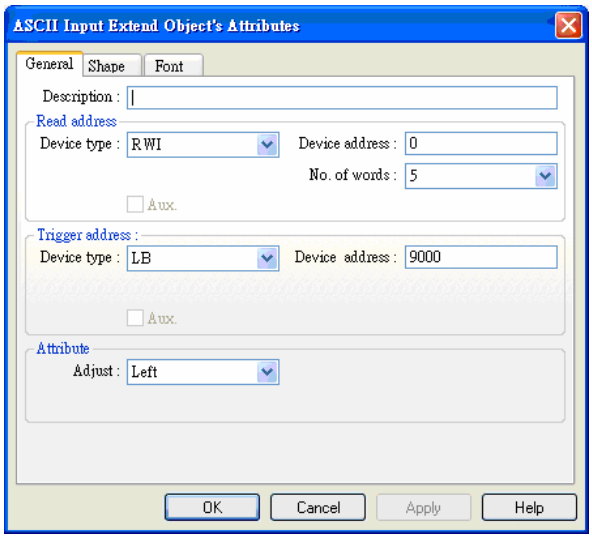
To complete the example, we create a ASCII input extend object to modify the updated data. Set Device type as 4x, device address as 200, No. of words as 5. In trigger address, set device type as LB and device address as 9000.



Create a Numeric Input Extend object to amend the data of LW9000 where device type is LW, device address is 9000; in trigger address, device type is LB and device address is 9000.

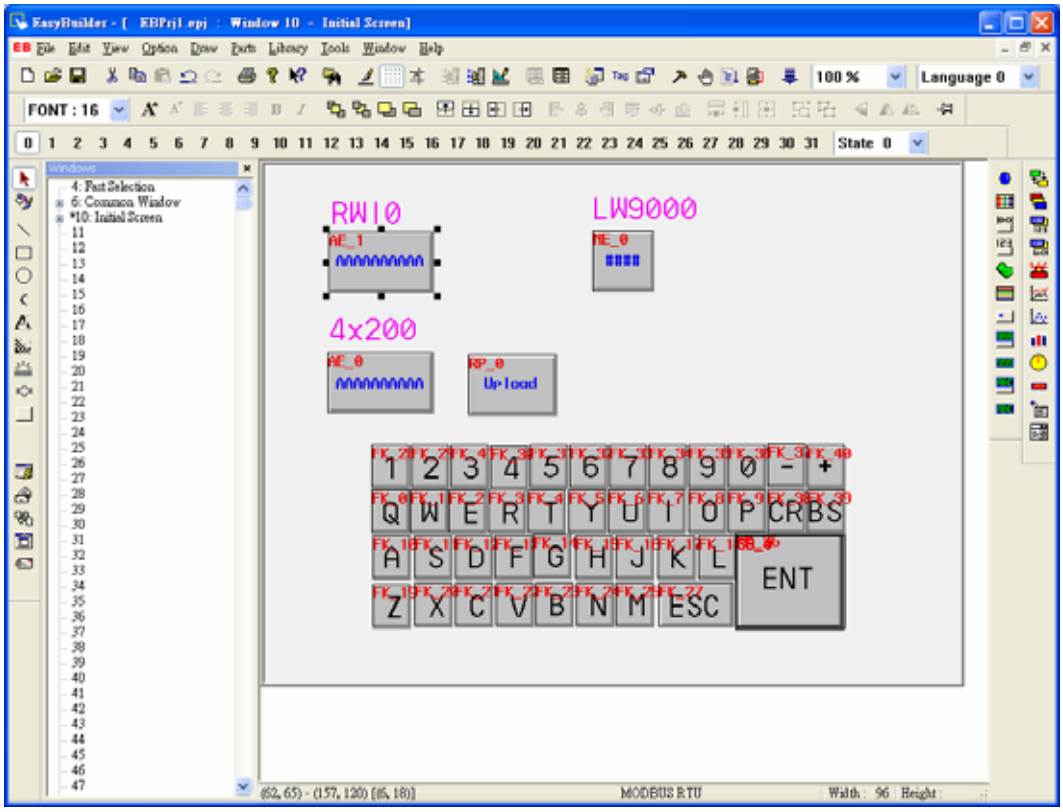


We place a ASCII input extend object to display the data of RW300 and check if the data is transferred. The setting shows as the dialog below.



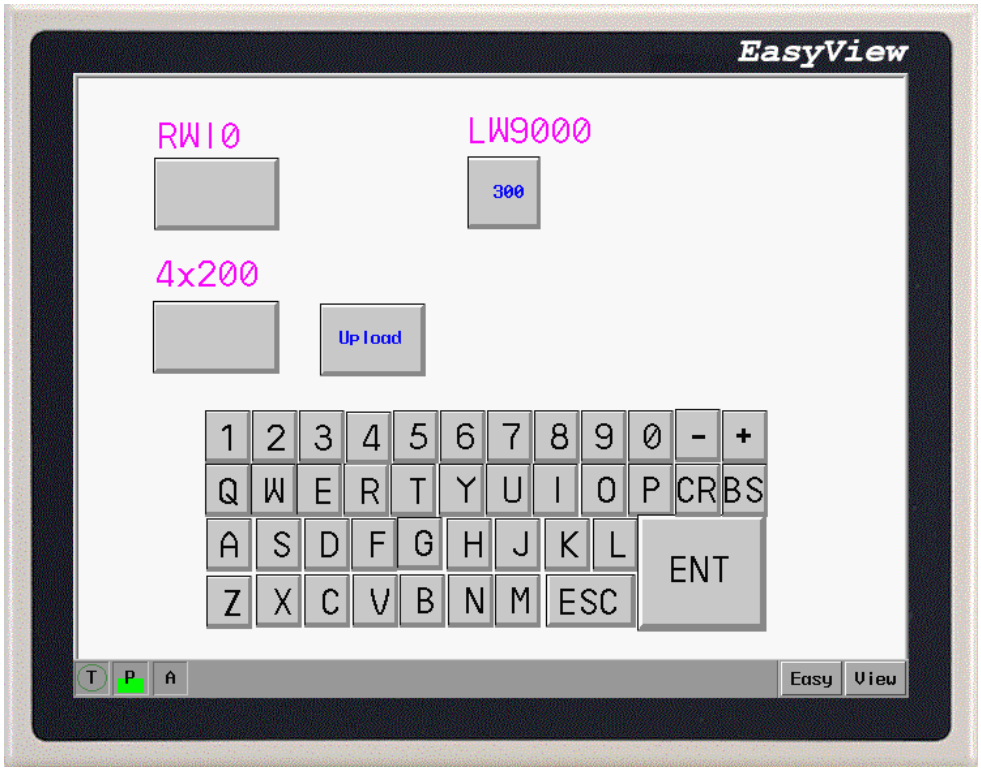
Then we place a keypad.

A complete project displays as follows:



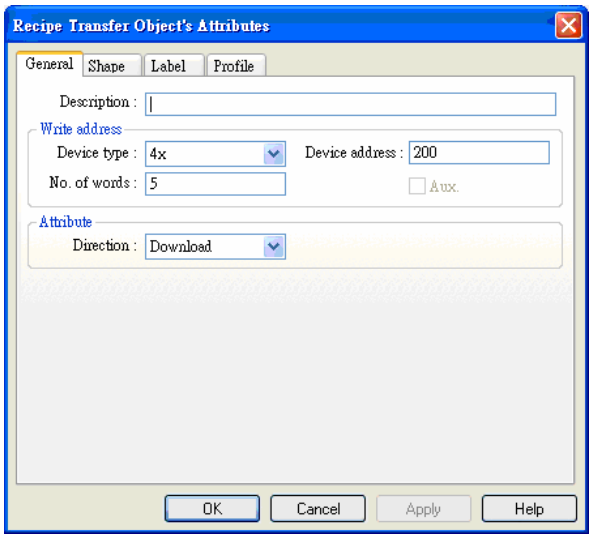
Save, compile and off-line simulate to run the project.

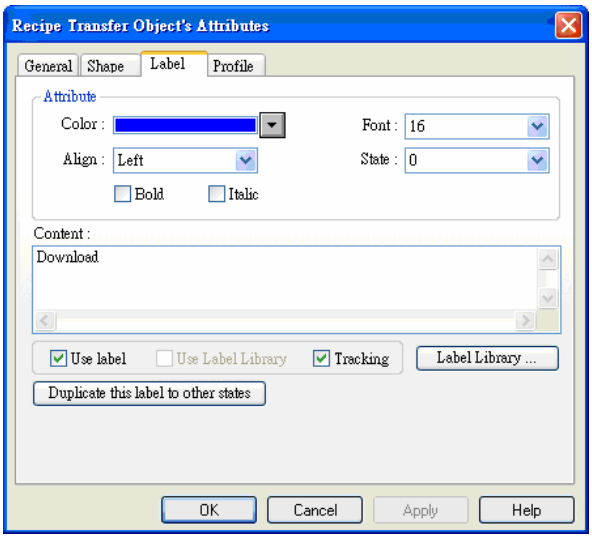
We set LW9000 as 300 first so that RW10 shows the data of RW300.



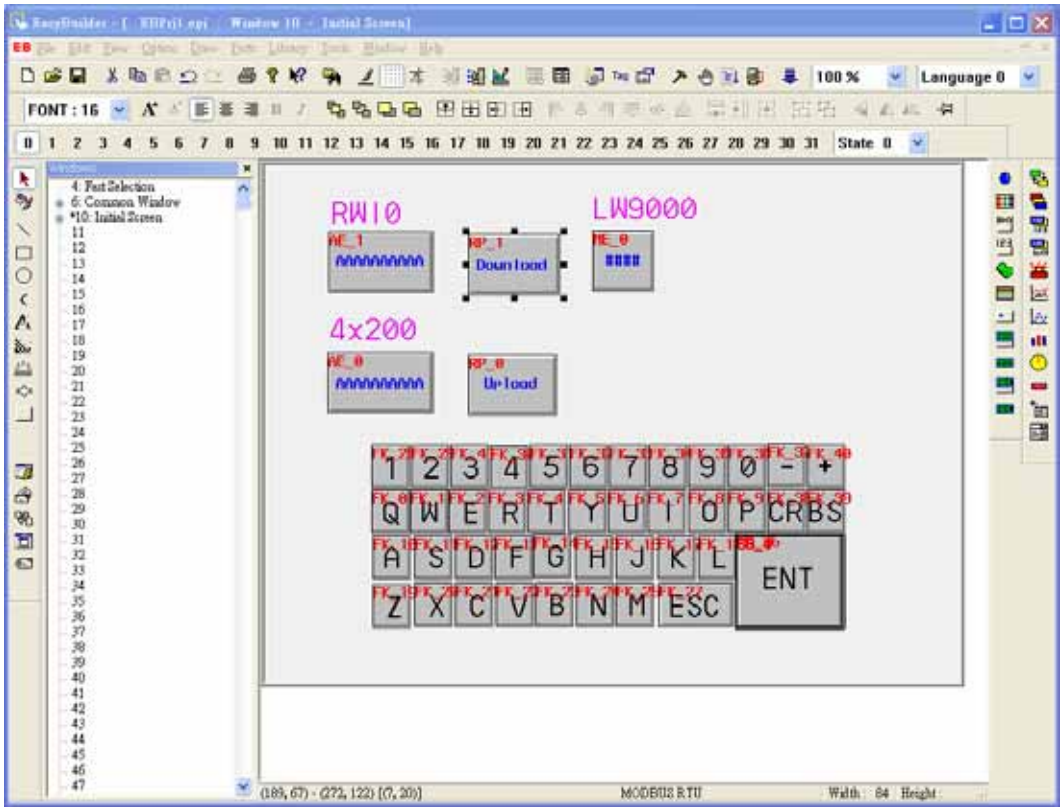
Then we input “ABCDEFGHIJ” into 4x200 and press save. You will find the data of RWI0 is the same as the data of 4x200 which means the upload succeed.

How could we download 5 continuous data after RW300 to the 5 continuous locations after 4x200? We add a recipe transfer object on the project where device type is 4x, device address is 200, No. of word is 5 and change the direction to download.





The following is the complete project:



When off-line simulation, we set LW9000 as 300 and input “AAAAAAAAAA”into RWI0:

When we press the download button, you will find the data is transfer from RW300 to 4x200.

From the example above, we can find that whether uploading the data of PLC to retentive memory or downloading the data to PLC, the starting addresses of retentive memory are all the corresponding address of LW9000.

8.3 Upload/ Download of the recipe memory between HMI and PLC

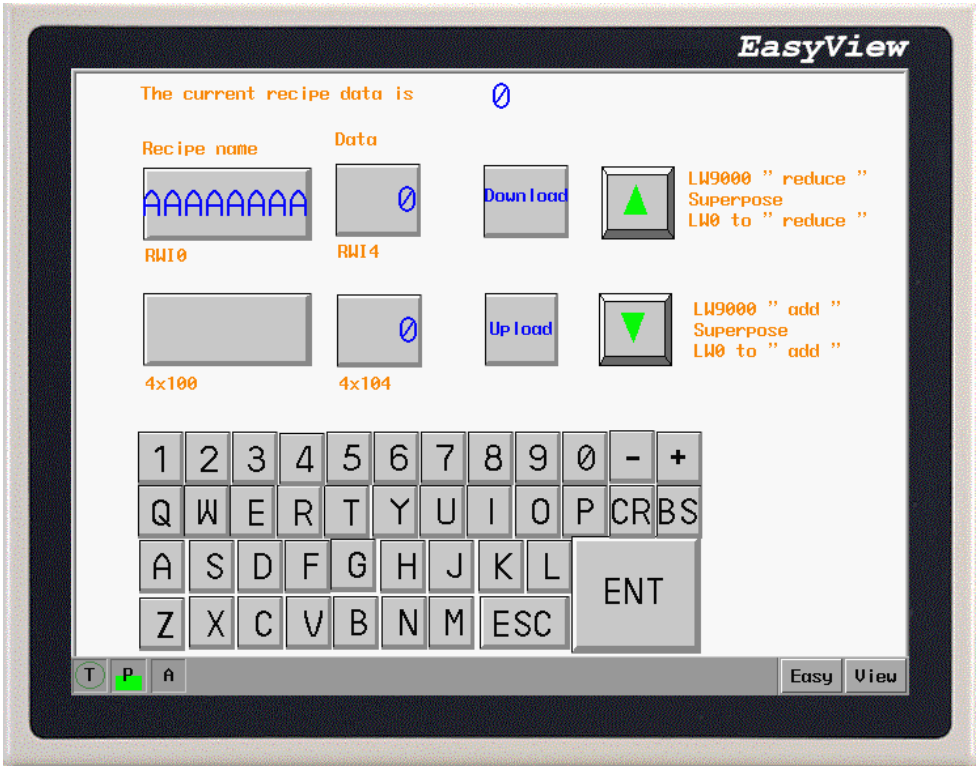
Recipe memory is very useful. Take production line as an example, the production facilities complete different tasks by different parameters provided. Now we can save the set of data to recipe memory according to the specific format. When we need them, we retrieve them without inputting a plenty of data temporarily.

There is example below of how to conveniently retrieve several recipe data:

Assume that there are 10 sets of recipes, each recipe is composed by 5 words, including Recipe name which takes up 4 words and recipe data which takes up 1 word. We arrange the recipe from RW0.

Serial number of the recipe	Register address	Recipe name(4 words)	Recipe data(1 word)
The 0th group	RW0~RW4	"AAAAAAAA"	0
The 1st group	RW5~RW9	"BBBBBBBB"	1111
The second group	RW10~RW14	"CCCCCCCC"	2222
The third group	RW15~RW19	"DDDDDDDD"	3333
The 4th group	RW20~RW24	"EEEEEEEE"	4444
The 5th group	RW25~RW29	"FFFFFFFF"	5555
The 6th group	RW30~RW34	"GGGGGGGG"	6666
The 7th group	RW35~RW39	"HHHHHHHH"	7777
The 8th group	RW40~RW44	"IIIIIIII"	8888
The 9th group	RW45~RW49	"JJJJJJJJ"	9999

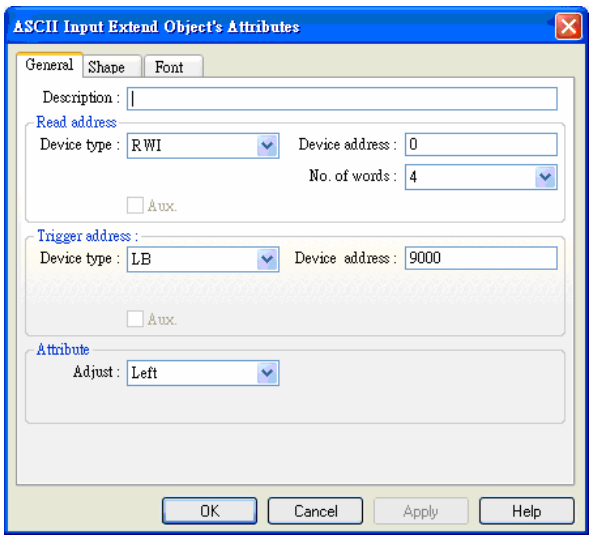
Through the project design, we plan to effectively exchange the data between PLC register 4x100 and each set of recipe data above. In a project, RWI0 and RWI4 display the receipt data of set number 0. Press download button to download the recipe data to 4x100; press upload button to upload the data of 4x100 to recipe memory. The upward button executes the upward lookup of the recipe data and the downward button executes the downward lookup of the recipe data.



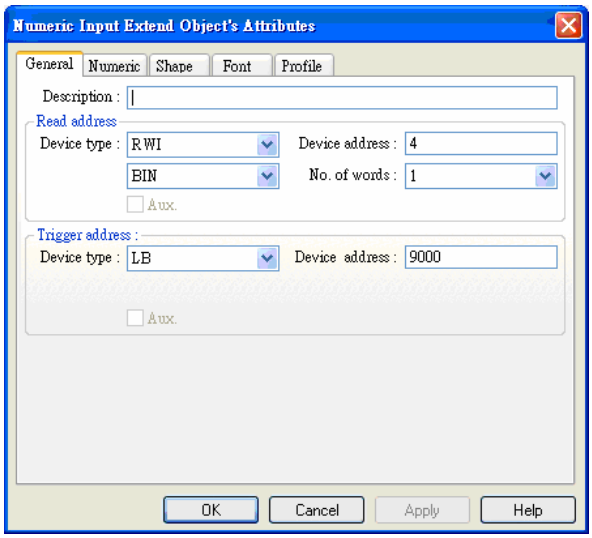
After roughly understanding the purpose of the project, we explicate the procedure of the project below.

At first, create a new project and choose PLC type as [MODBUS RTU] in [Edition]/[System parameters].

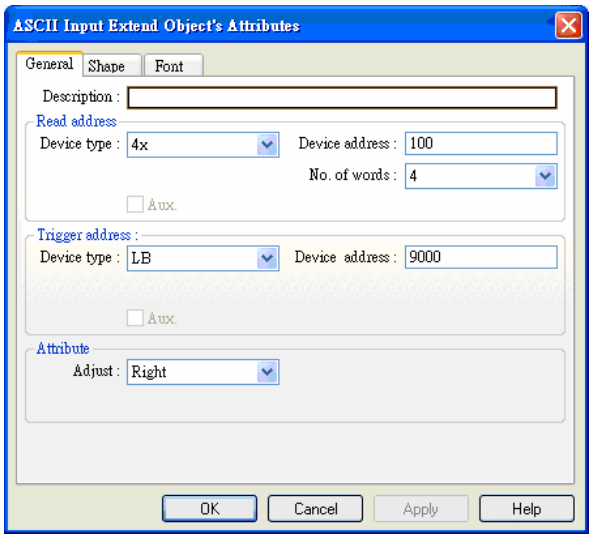
Create a ASCII Input Extend object to display and amend the recipe name.

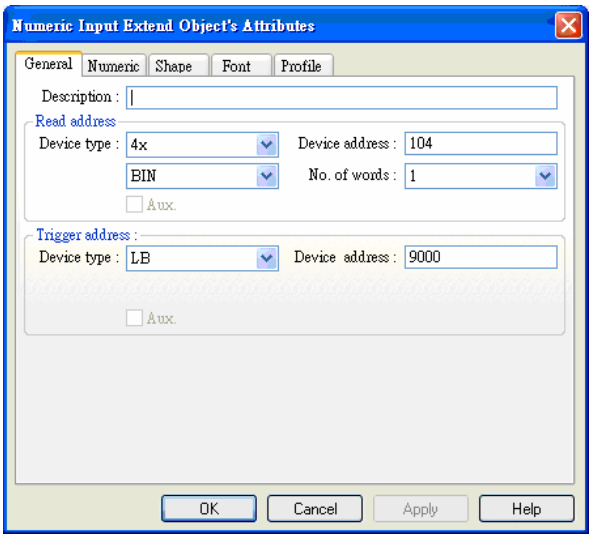


Create a Numeric Input Extend to display and amend the recipe data.

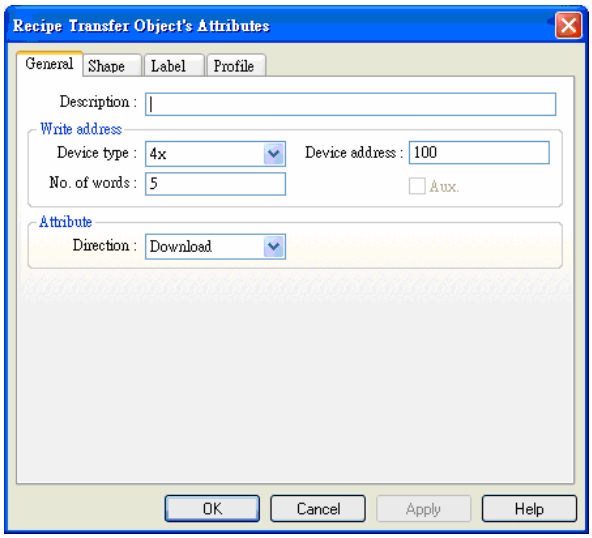


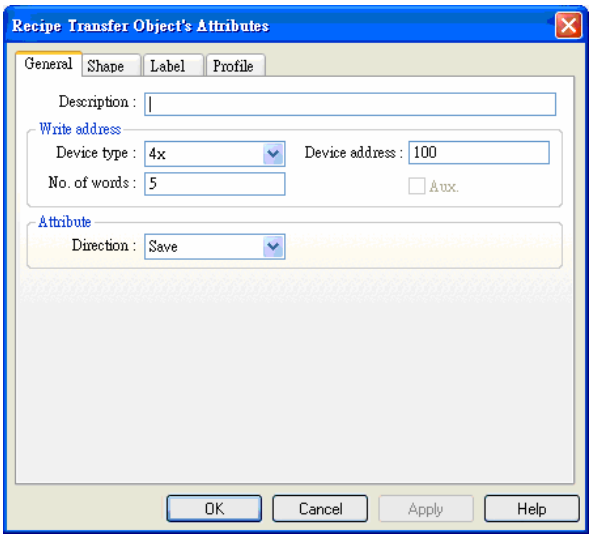
Create a ASCII Input Extend and a Numeric Input Extend to display and amend the recipe data in PLC.





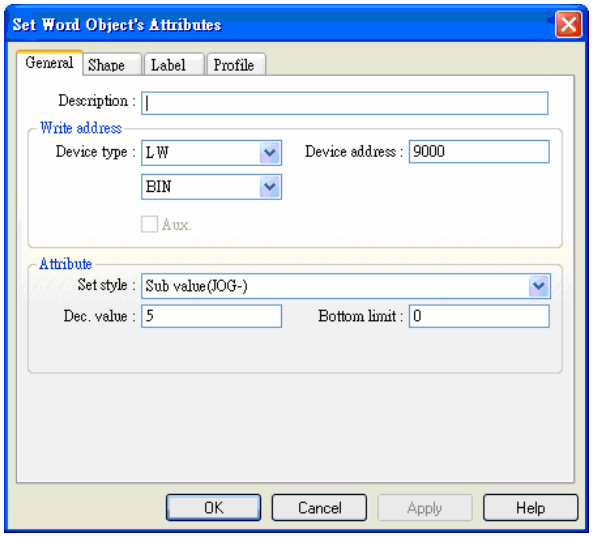
Create two recipe transfer objects: one is for downloading recipe data and another is for uploading recipe data.



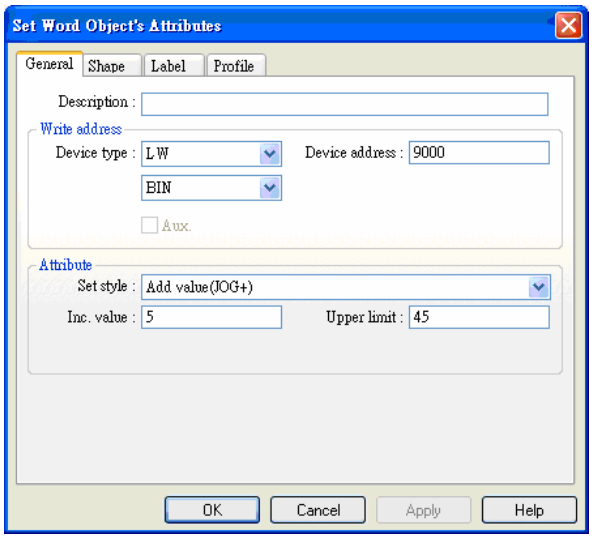


Then we design two buttons for users to conveniently look up and amend each set of recipe data: one is for looking up forward and another is for looking up backward.

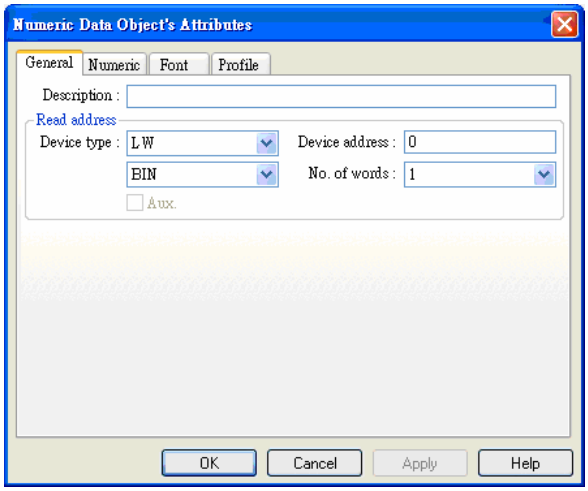
We set look up forward as a multi-state switch object. Every time when you press this object, system deducts 5 from the value of LW9000. Because each set of recipe data includes 5 words, RWI0 displays the previous recipe data each press to reach the purpose of looking up forward.



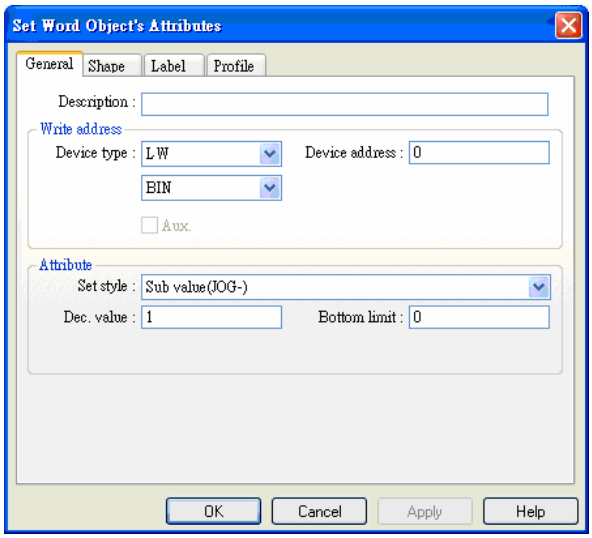
We set look up backward as a multi-state switch object. In the same theory, every time when you press this object, system adds 5 from the value of LW9000. Because each set of recipe data includes 5 words, RWI0 displays the previous recipe data each press to reach the purpose of looking backward. Here the upper limit is 45 (10 sets of recipe).

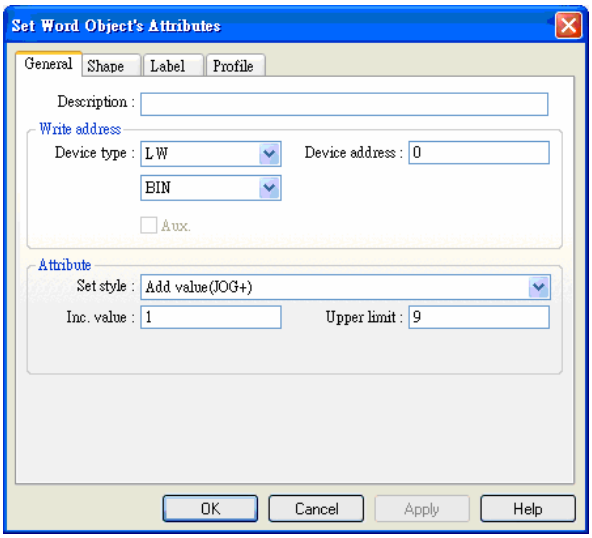


Create a Numeric Data object to display the current recipe data.



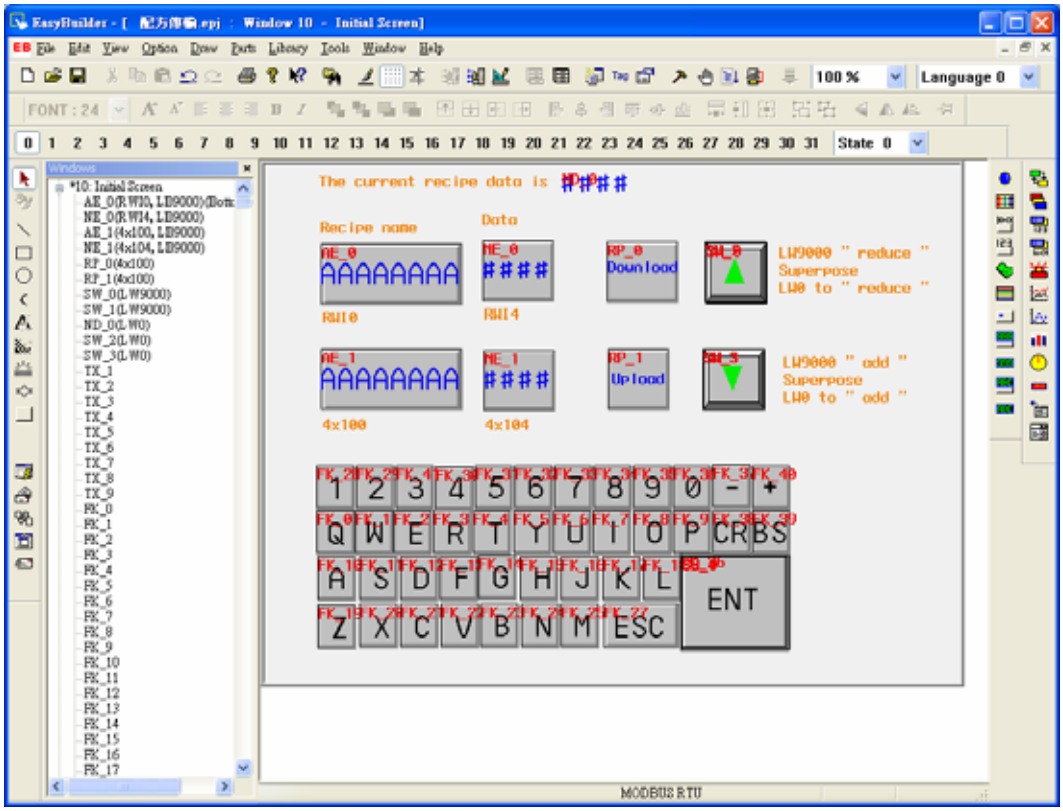
However, how do we know which set of recipe data is currently displayed? How to control the value of LW0? Here we create two more multi-state switch objects, one is subtraction and another is addition, which display as follows:



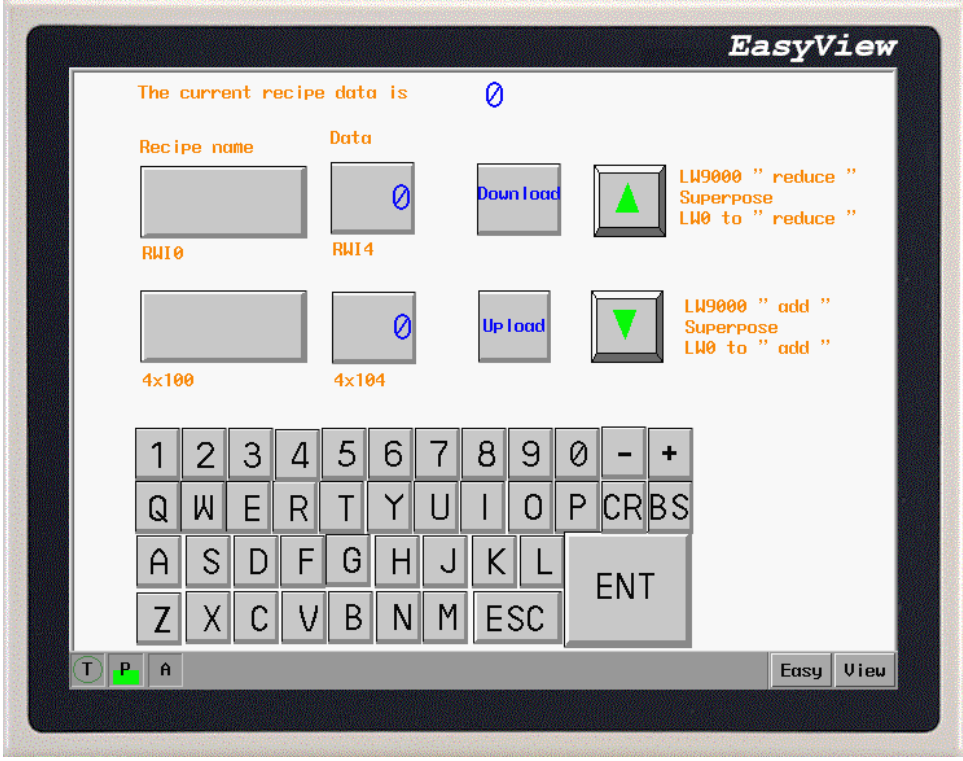


LW9000 " reduce " Superpose LW0 to " reduce "; LW9000 " add "Superpose LW0 to " add ", Thus, when we look up the recipe data, the value of LW0 changes and display the current recipe data.

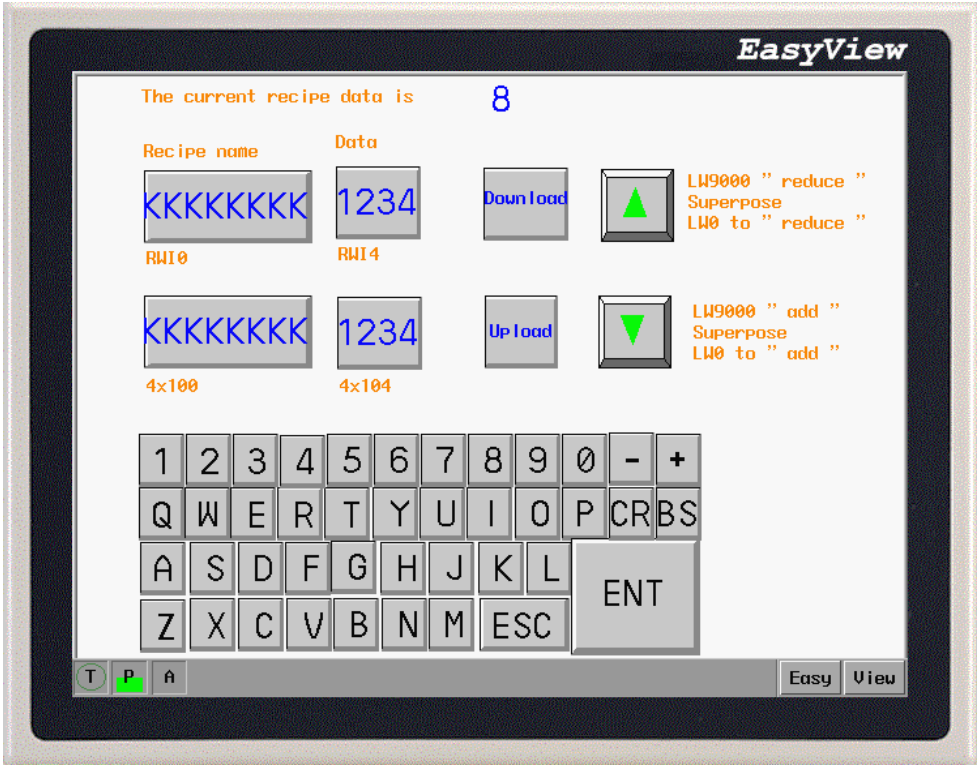
Then we place a keypad, add some context for embellishment. One project is done as below:



Save, compile and off-line simulation to run the project:



We input the 10 sets of recipes into recipe memory card and then jump to the eighth set. Change the recipe name as“KKKKKKKK”, recipe data as 1234 and press the download button:



We find the data in 4x100 becomes " KKKKKKKK " , 1234. The changes in memory show as below:

Serial number of the recipe	Register address	Recipe name(4 words)	Recipe data(1 word)
The 0th group	RW0~RW4	“AAAAAAAA”	0
The 1st group	RW5~RW9	“BBBBBBBB”	1111
The second group	RW10~RW14	“CCCCCCCC”	2222
The third group	RW15~RW19	“DDDDDDDD”	3333
The 4th group	RW20~RW24	“EEEEEEEE”	4444
The 5th group	RW25~RW29	“FFFFFFFF”	5555
The 6th group	RW30~RW34	“GGGGGGGG”	6666
The 7th group	RW35~RW39	“HHHHHHHH”	7777
The 8th group	RW40~RW44	“IIIIIIII”	8888
The 9th group	RW45~RW49	“JJJJJJJJ”	9999

Address	Data
4x100-4x103	"KKKKKKKK"
4x104	1234

Through the procedure of the project, we grasp the basic idea on designing recipe data.