

Demo Project for Cyrillic ASCII2Unicode

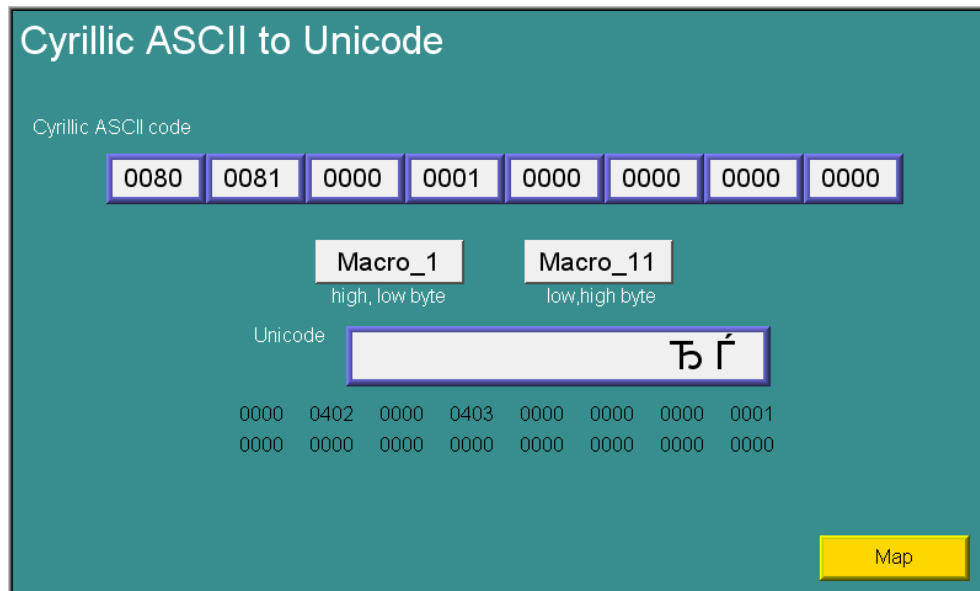
Table of Contents

1. Overview and Operation
2. Setting Up the Screen
3. Addresses

1. Overview and Operation

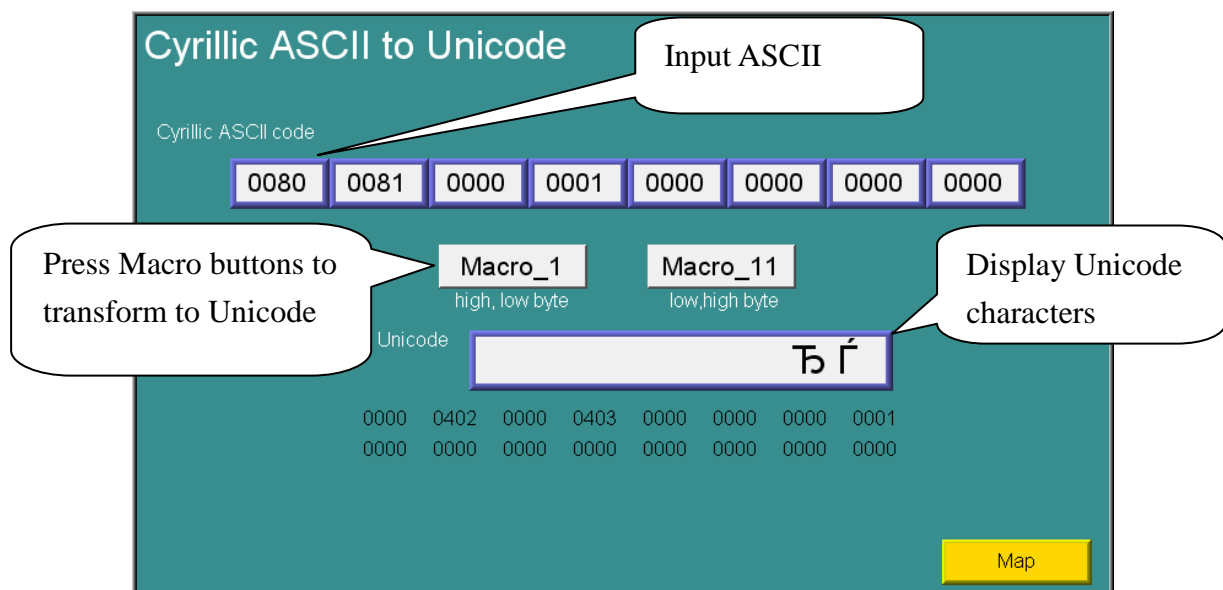
[Overview]

This demo project demonstrates how to use Macro to transform PLC ASCII string to Unicode characters according to the Unicode included in Character Map and display them on HMI.



[Operation]

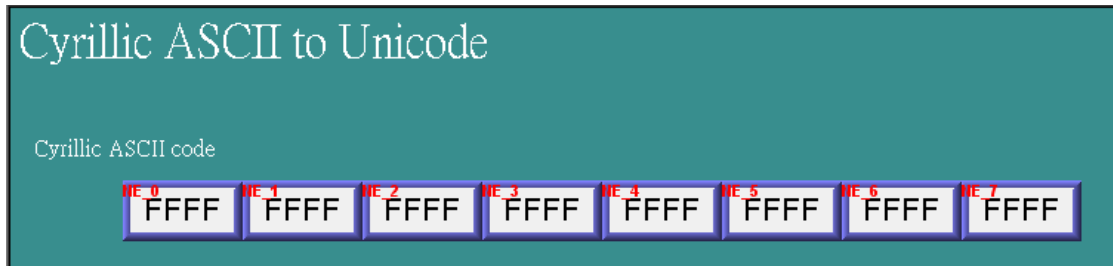
Input ASCII using Numeric Input objects then press Macro_1 to display Unicode using ASCII Display object. Macro_1 and Macro_11 display in different order. Macro_1 displays high byte first, then low byte. Whereas Macro_11 displays low byte first then high byte as shown in the 2 lines at the bottom.



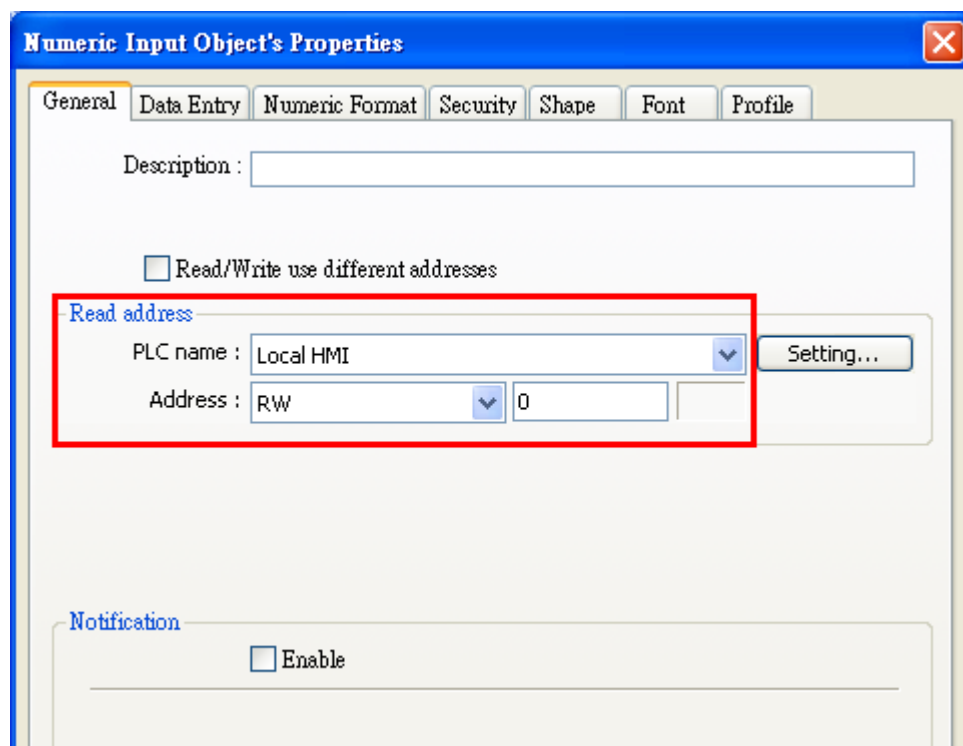
2. Setting up the screen

2-1 Setting objects on Window 10

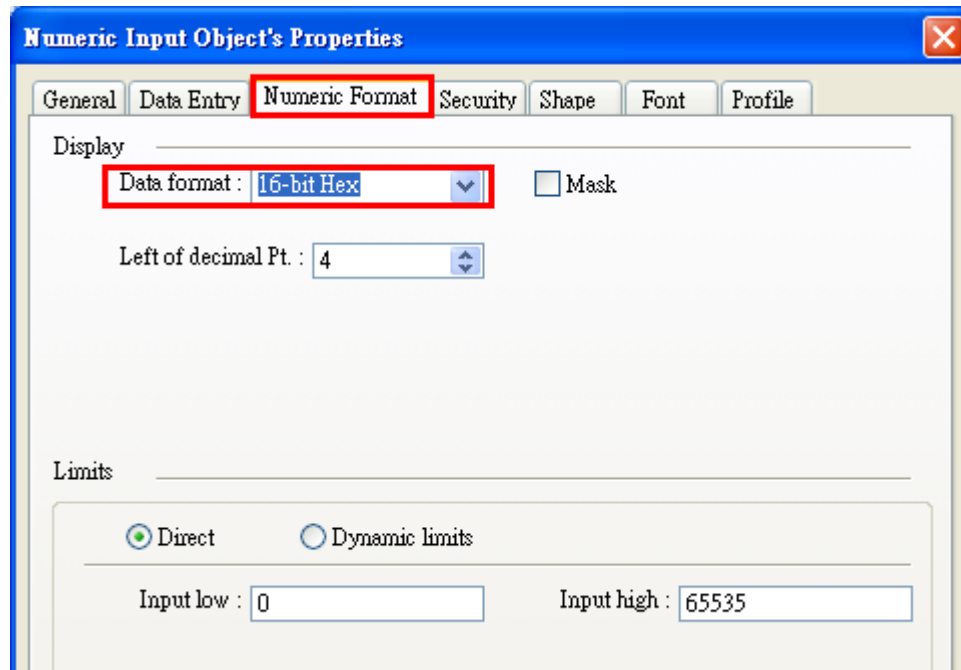
- Add Numeric Input objects with RW addresses in 16-bit HEX format.



Set **[Read Address]** to “RW”.

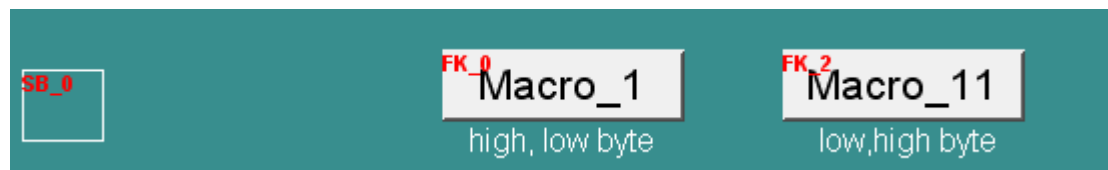


For Numeric Input object, select **[Numeric Format]** tab. For **[Data format]**, select “16-bit Hex”.

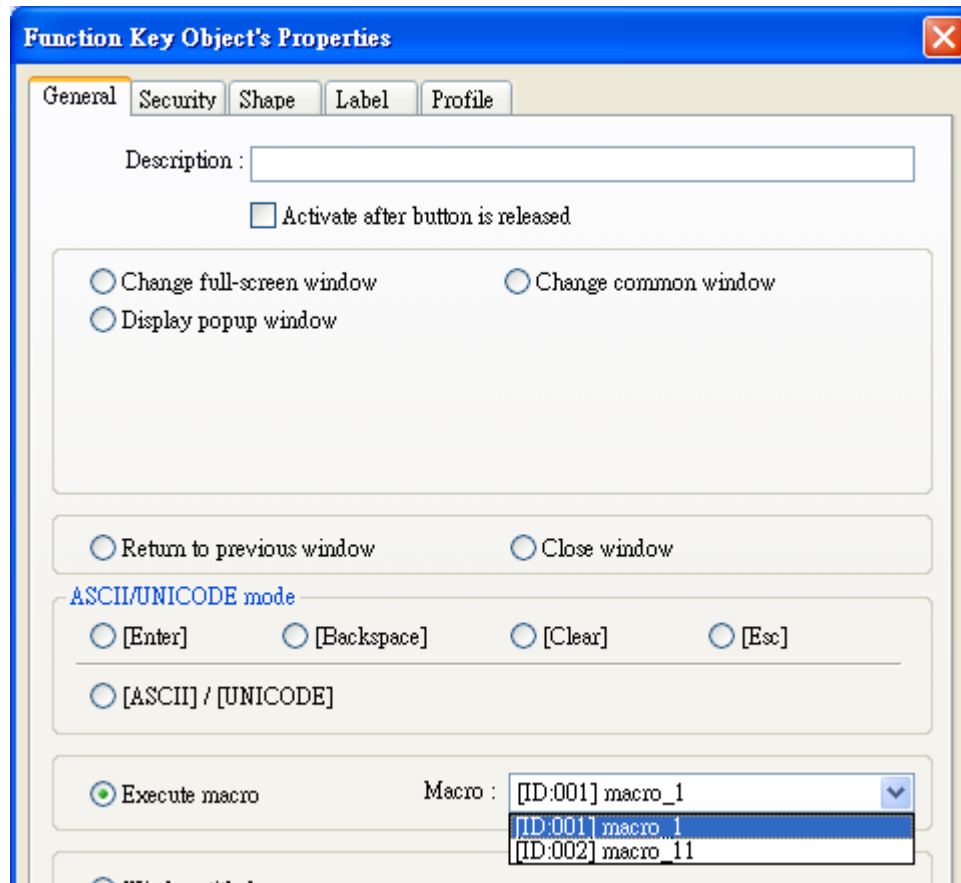


Create 8 Numeric Input objects with addresses "RW0~RW7".

- Create 2 Function Key objects to execute Macro_1 and Macro_11.



Tick **[Execute Macro]**, and then select the Macro to be executed from the pull down list.

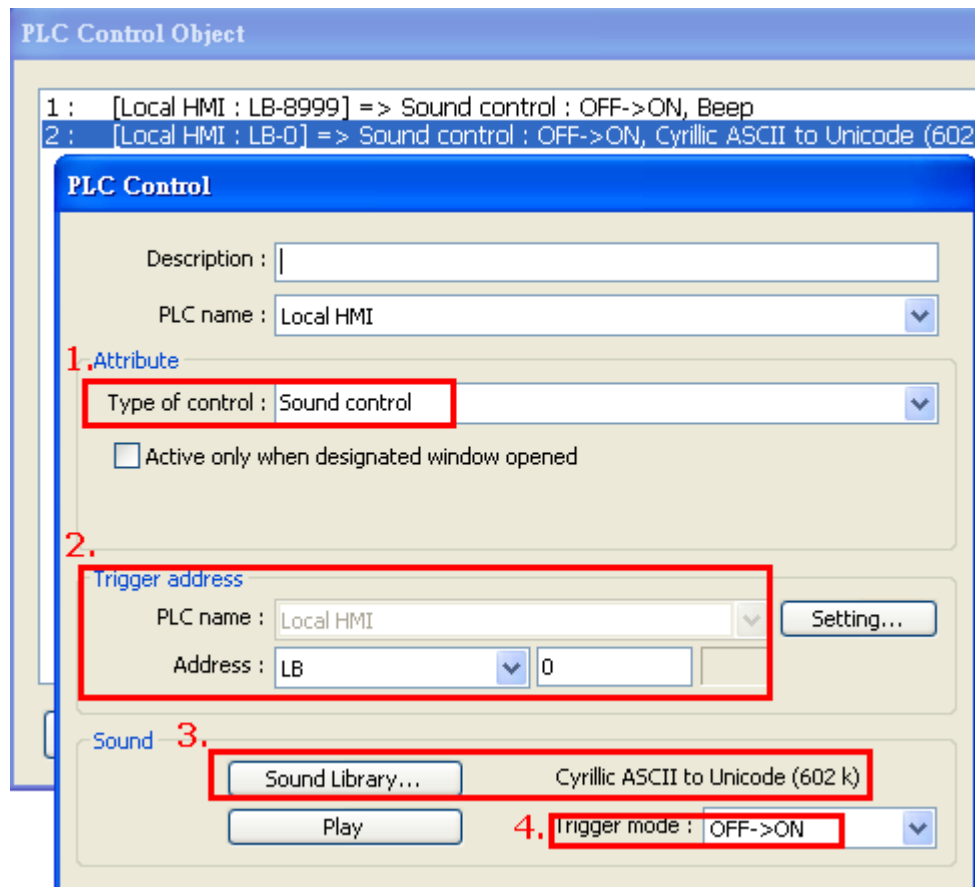


- Create a Set Bit object for sound control.

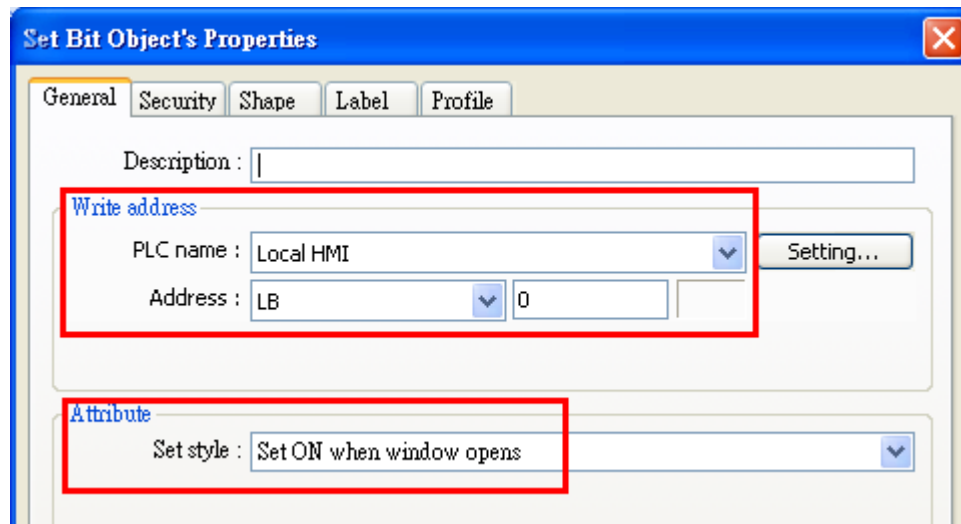


Set the following functions for PLC Control object:

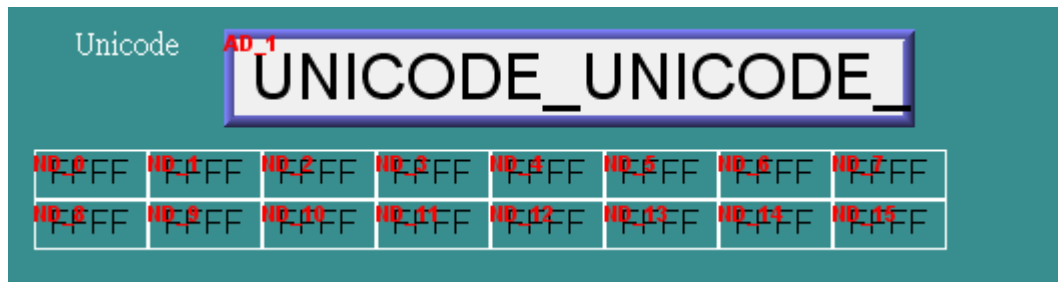
1. **[Type of control]:** Sound control
2. **[Trigger address]:** LB 0
3. Select the sound form **[Sound Library]**. When the project is started this sound will be played.
4. **[Trigger mode]:** OFF->ON



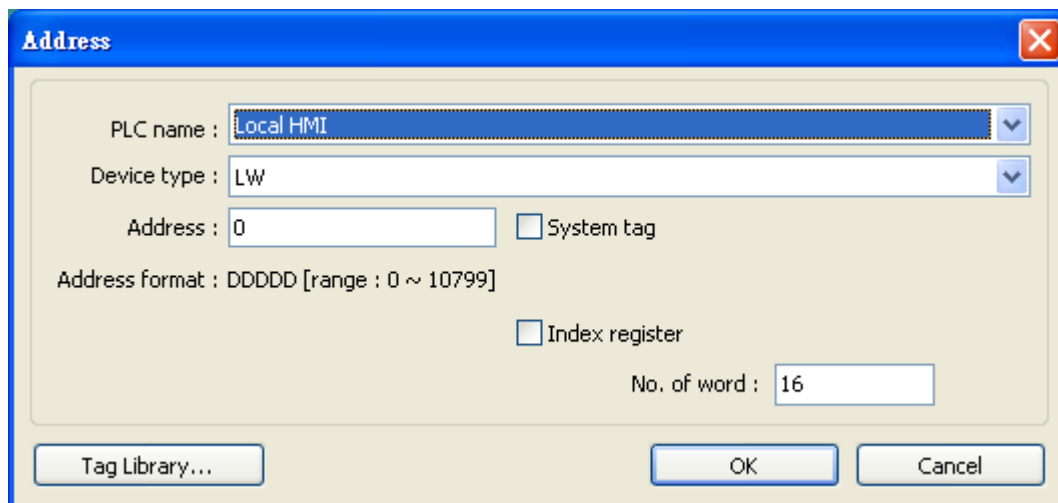
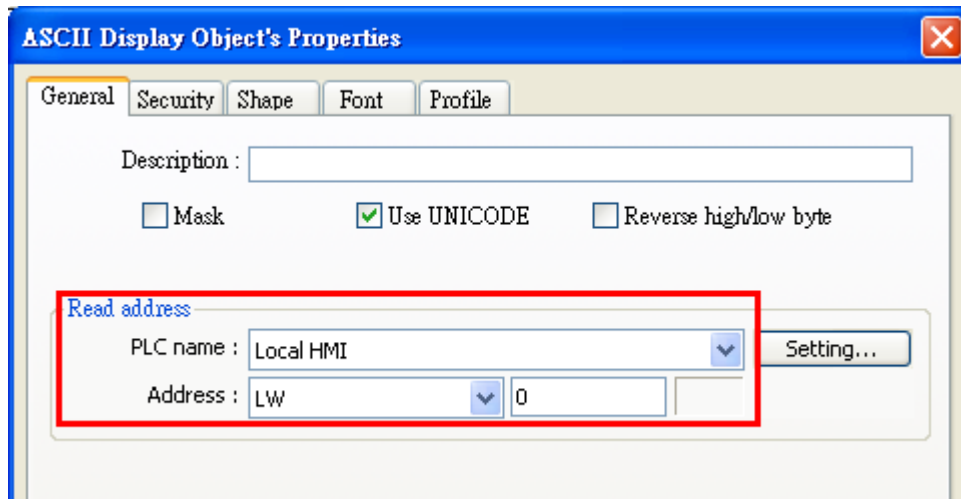
When LB 0 status changes from OFF to ON, the sound chosen will be played. Add another Set Bit object, set **[Write address]** to "LB0" and **[Attribute]** to "Set ON when window opens".



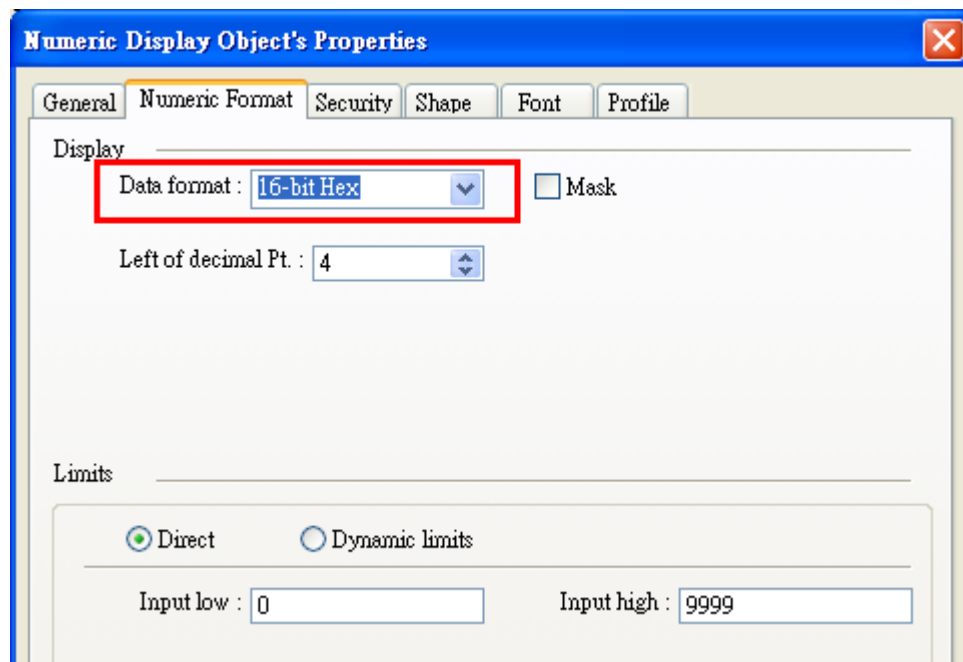
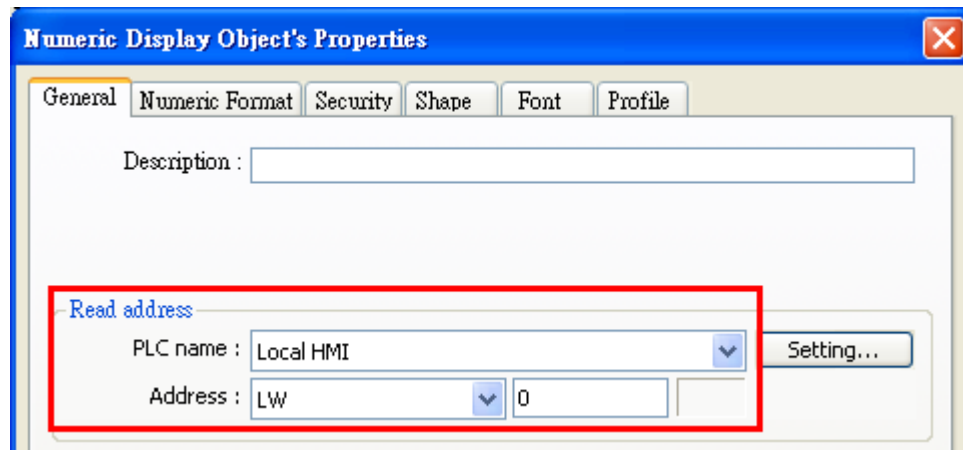
- Create 1 ASCII Display object and 16 Numeric Display objects.



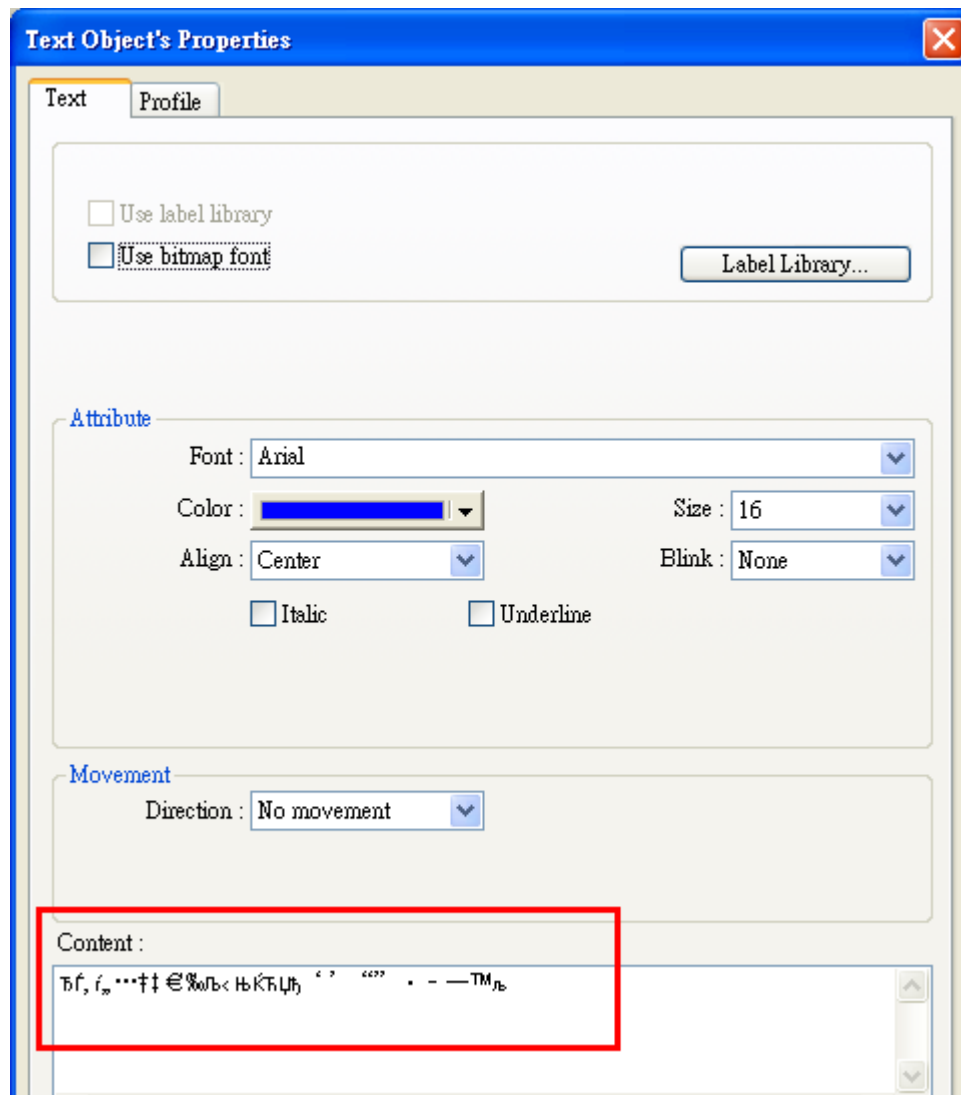
For ASCII Display object, set **[Read address]** to “LW0”. Press **[Setting...]**, set **[No. of word]** to “16”.



Create 16 Numeric Display objects from addresses “LW0~LW15”. Under **[Numeric Format]** tab select “16bit- Hex”.





2-2 On Window 11, create Text object for inputting Cyrillic characters used in the project. In this way this font can be downloaded and displayed correctly on HMI



2-3 Character Map on window 12

The ASCII map is shown below; users can check the ASCII number to be filled in Numeric Input objects on window 10.

For example, refer to the value left side and upper side.

To display , write "0081" to Numeric Input object, and for , "00A8".

GP_0

FONT: 24 Ascfont.24

	0	<u>1</u>	2	3	4	5	6	7	<u>8</u>	9	A
002		!	"	#	\$	%	_	'	()	*
003	0	1	2	3	4	5	6	7	8	9	:
004	@	A	B	C	D	E	F	G	H	I	J
005	P	Q	R	S	T	U	V	W	X	Y	Z
006	`	a	b	c	d	e	f	g	h	i	j
007	p	q	r	s	t	u	v	w	x	y	z
<u>008</u>	Б	<u>Г</u>	,	г	//	...	†	‡	€	%	Ђ
009	ђ	`	'	`	"	•	—	—	□	™	Љ
<u>00A</u>		Ÿ	ÿ	J	ø	Г		§	<u>È</u>	©	€
00B	°	±	⊥	ı	Г	μ	℥	•	ë	№	€

Cyrillic ASCII to Unicode



2-4 Macro_1

According to the Character Map that shows how Unicode corresponds to ASCII code, write the code to Macro.

For example: UNICODE[128]=0x0402 is the transformation of U+0402(0x80). As shown below, 0x80 is transformed into decimal, 128.

Ј	Ѱ	—		а	в	с	д	е	г	у	и	п	и	и	о	р
q	r	s	t	u	v	w	x	y	z	{		}		ђ	џ	...
†	‡	€	%	љ	ќ	њ	џ	џ	џ	џ	џ	џ	џ	џ	џ	џ
›	њ	ќ	ћ	џ		Ў	ў	Ј	џ	Ѓ	Ѓ	Ѓ	Ѓ	Ѓ	Ѓ	Ѓ
Ѓ	°	±	і	г	μ	¶	·	ё	№	е	»	ј	ѕ	ѕ	ї	А
Г	Д	Е	Ж	З	И	Й	К	Л	М	Н	О	П	Р	С	Т	У
Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я	а	б	в	г	д	е	ж	з
и	й	к														

Characters to copy :

☒ Advanced view

Character set :

Group by :

Search for :

U+0402 (0x80): Cyrillic Capital Letter Dje

WorkSpace

Macro ID :

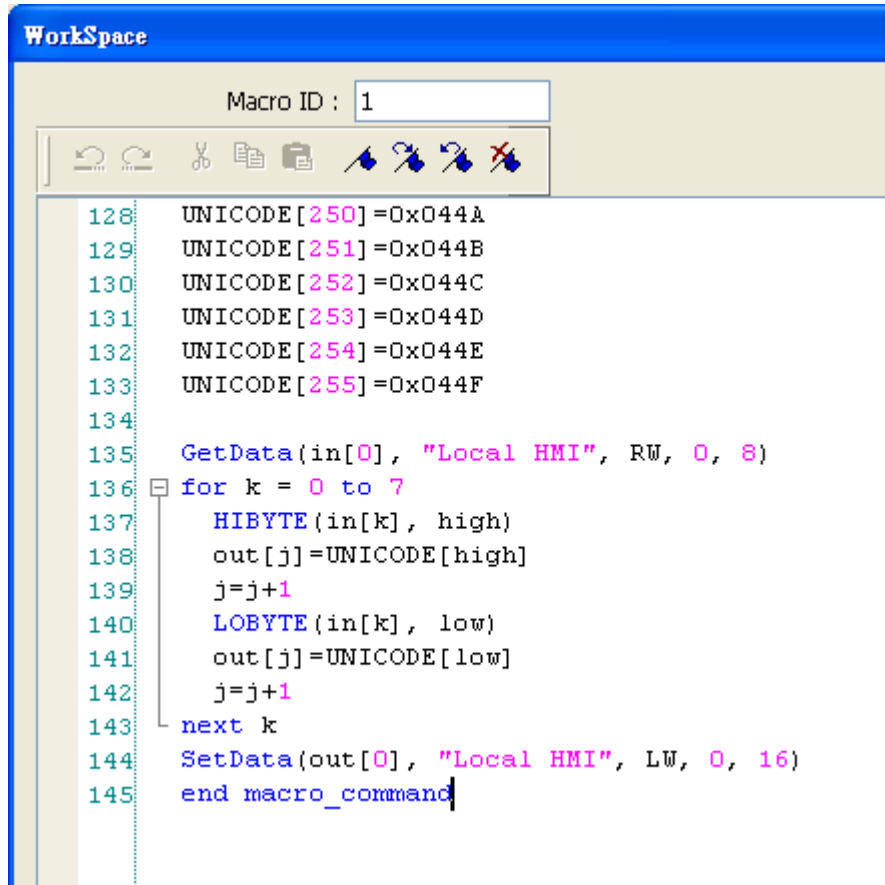
```

1
2  macro_command main()
3  short UNICODE[256], i, j=0,k, in[8],out[16], high, low
4  for i = 0 to 255
5      UNICODE[i]=i
6  next i
7  UNICODE[128]=0x0402
8  UNICODE[129]=0x0403
9  UNICODE[130]=0x201A
10 UNICODE[131]=0x0453
11 UNICODE[132]=0x201E
12 UNICODE[133]=0x2026
13 UNICODE[134]=0x2020
14 UNICODE[135]=0x2021
15 UNICODE[136]=0x20AC
16 UNICODE[137]=0x2030
17 UNICODE[138]=0x0409
18 UNICODE[139]=0x2039
19 UNICODE[140]=0x040A
20 UNICODE[141]=0x040C
21 UNICODE[142]=0x040B
22 UNICODE[143]=0x040F
23 UNICODE[144]=0x0452
24 UNICODE[145]=0x2018

```

In the end of macro, from line 135 to line 144 change the value received

form RW0~RW7 to high byte and low byte of Unicode, then display on LW0~LW15.

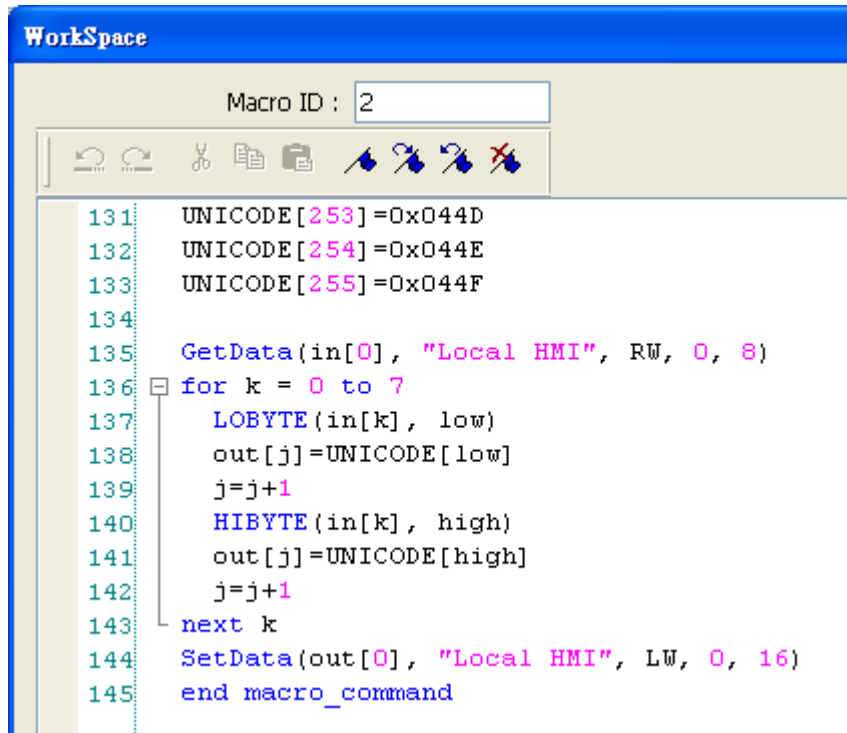


```

128  UNICODE[250]=0x044A
129  UNICODE[251]=0x044B
130  UNICODE[252]=0x044C
131  UNICODE[253]=0x044D
132  UNICODE[254]=0x044E
133  UNICODE[255]=0x044F
134
135  GetData(in[0], "Local HMI", RW, 0, 8)
136  for k = 0 to 7
137      HIBYTE(in[k], high)
138      out[j]=UNICODE[high]
139      j=j+1
140      LOBYTE(in[k], low)
141      out[j]=UNICODE[low]
142      j=j+1
143  next k
144  SetData(out[0], "Local HMI", LW, 0, 16)
145  end macro_command

```

As for macro_11, low byte is displayed first and then high byte.



```

131  UNICODE[253]=0x044D
132  UNICODE[254]=0x044E
133  UNICODE[255]=0x044F
134
135  GetData(in[0], "Local HMI", RW, 0, 8)
136  for k = 0 to 7
137      LOBYTE(in[k], low)
138      out[j]=UNICODE[low]
139      j=j+1
140      HIBYTE(in[k], high)
141      out[j]=UNICODE[high]
142      j=j+1
143  next k
144  SetData(out[0], "Local HMI", LW, 0, 16)
145  end macro_command

```

3. Addresses

The Object Addresses used in this demo project are listed below: Users can change Addresses and Object ID base on actual usage.

Addresses		Object's ID	Detail
Operating objects		Window 10	
Word	RW0	NE_0	Numeric input object
	RW1	NE_1	Numeric input object
	RW2	NE_2	Numeric input object
	RW3	NE_3	Numeric input object
	RW4	NE_4	Numeric input object
	RW5	NE_5	Numeric input object
	RW6	NE_6	Numeric input object
	RW7	NE_7	Numeric input object
	LW0	AD_1	ASCII Display with 16 words
	LW0	ND_0	Numeric Display
	LW1	ND_1	Numeric Display
	LW2	ND_2	Numeric Display
	LW3	ND_3	Numeric Display
	LW4	ND_4	Numeric Display
	LW5	ND_5	Numeric Display
	LW6	ND_6	Numeric Display
	LW7	ND_7	Numeric Display
	LW8	ND_8	Numeric Display
	LW9	ND_9	Numeric Display
	LW10	ND_10	Numeric Display
	LW11	ND_11	Numeric Display
	LW12	ND_12	Numeric Display
	LW13	ND_13	Numeric Display
	LW14	ND_14	Numeric Display
	LW15	ND_15	Numeric Display
Bit	LB0	SB_0	Sound control
Others		FK_0	Execute Macro_01
		FK_1	Execute Macro_11
Cyrillic character		Window 11	
Others		TX_0~TX_5	Cyrillic characters
Character Map		Window 12	
		GP_0	Cyrillic ASCII Character Map