

Instruction Set

R&D Designed	R&D Approved	Editor
Robert	Peng Jun	Emma

Table of Contents

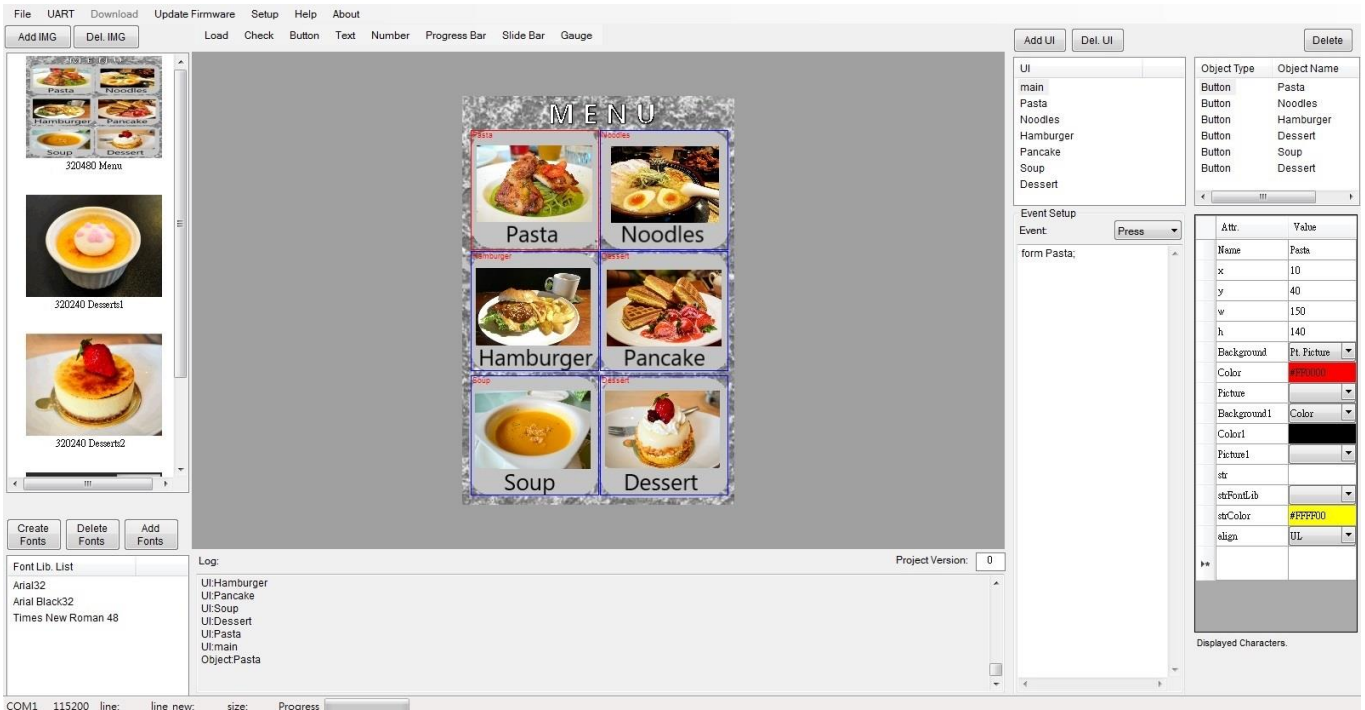
List	Description	Page No.
	Cover	1
	Revision Record	2
	Table of Contents	3
1	Scope	4
2	General Information	4
3	User Interface	4
4	Libraries	5
5	Functions	6
6	Widgets	7
7	Programming	11
8	Update Project with SD Card	16

1. Scope

This manual defines general instruction for the UI Editing Tool for UART TFT LCD module supplied by AMSON electronics.

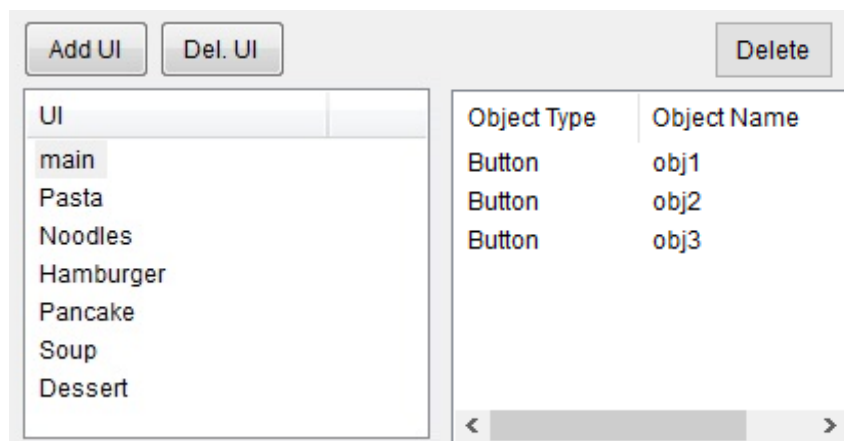
2. General Information

There are several widgets for designing the user interface. By clicking any widget, the corresponding object will be shown on the window. User can set various attributes, e.g., the location, size, text in the right column.



3. User Interface

User can create new User Interface, in which they can set pictures, buttons, gauges, and so on. Note that the maximum quantity of UI is 255.



* A maximum of 32 objects is limited for each UI.

4. Libraries

User can store up to 40 images. Note that the resolution of the image must be the same as that of the LCD module while choosing Ext. Picture for the background of the UI.



User must Create Fonts before inputting text.





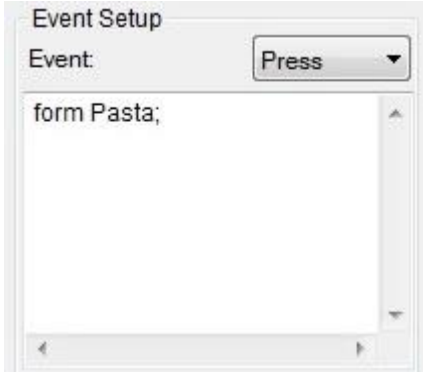







* A maximum of 16 ascii characters is limited for the name length of each file.

5. Functions

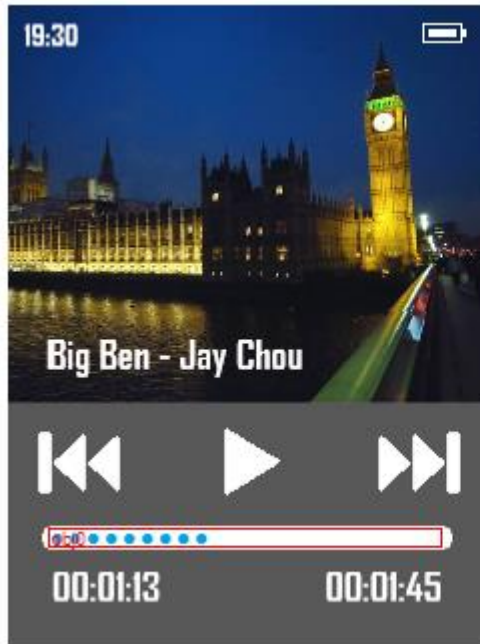
Item	Attributes
File	New, open, save and exit.
UART	After connecting the LCD module to the computer, choose the corresponding port and Baud Rate. The Baud Rate is used for setting the communication baud rate between the LCD module and the host.
Update Firmware	Open a *.bin file and update the firmware.
Setup	To be decided.
Help	Link to the UI Editing Tool Manual.
About	Describe the software version.
Load	Download the project (user interfaces) into the LCD module.
Check	Choose the object and check its code, which is written in the Event column; Simulation works as a simulator.

6. Widgets

Item	Attributes																																
<p data-bbox="236 1193 338 1227">Button</p>	<p data-bbox="379 383 1278 488">User can set the Background (color, picture or transparent) & Background1, and input text to make a simple color button or a self-made one by Ext. Picture.</p> <div data-bbox="467 533 893 734">  </div> <p data-bbox="608 741 831 801">↓ press</p> <div data-bbox="467 813 893 1014">  </div> <div data-bbox="933 533 1268 1014"> <table border="1"> <tr><td>Background</td><td>Color</td></tr> <tr><td>Color</td><td>#808080</td></tr> <tr><td>Picture</td><td></td></tr> <tr><td>Background1</td><td>Color</td></tr> <tr><td>Color1</td><td>#C0C0C0</td></tr> <tr><td>Picture1</td><td></td></tr> <tr><td>str</td><td>Send</td></tr> <tr><td>strFontLib</td><td>Arial48</td></tr> <tr><td>strColor</td><td>#000080</td></tr> <tr><td>align</td><td>CC</td></tr> </table> </div> <div data-bbox="443 1066 850 1267">  </div> <p data-bbox="571 1261 770 1321">↓ press</p> <div data-bbox="443 1328 850 1529">  </div> <div data-bbox="943 1149 1278 1440"> <table border="1"> <tr><td>Background</td><td>Ext. Picture</td></tr> <tr><td>Color</td><td></td></tr> <tr><td>Picture</td><td>BUTTON1</td></tr> <tr><td>Background1</td><td>Ext. Picture</td></tr> <tr><td>Color1</td><td></td></tr> <tr><td>Picture1</td><td>BUTTON2</td></tr> </table> </div>	Background	Color	Color	#808080	Picture		Background1	Color	Color1	#C0C0C0	Picture1		str	Send	strFontLib	Arial48	strColor	#000080	align	CC	Background	Ext. Picture	Color		Picture	BUTTON1	Background1	Ext. Picture	Color1		Picture1	BUTTON2
	Background	Color																															
Color	#808080																																
Picture																																	
Background1	Color																																
Color1	#C0C0C0																																
Picture1																																	
str	Send																																
strFontLib	Arial48																																
strColor	#000080																																
align	CC																																
Background	Ext. Picture																																
Color																																	
Picture	BUTTON1																																
Background1	Ext. Picture																																
Color1																																	
Picture1	BUTTON2																																
<p data-bbox="379 1570 1358 1641">In the Event column, user can set up the program when the button is pressed or released.</p> <div data-bbox="662 1675 1086 2045">  </div>																																	

<p>Text</p>	<p>Besides location, background and font color, the alignment of the text is available.</p> <div style="display: flex; align-items: center; justify-content: center;">  <table border="1" style="margin-left: 20px;"> <tr><td>Background</td><td>Color</td><td>▼</td></tr> <tr><td>Color</td><td>#80FF80</td><td></td></tr> <tr><td>Picture</td><td></td><td>▼</td></tr> <tr><td>str</td><td>AMSON</td><td></td></tr> <tr><td>strFontLib</td><td>1</td><td>▼</td></tr> <tr><td>strColor</td><td>#0080FF</td><td></td></tr> <tr><td>align</td><td>CC</td><td>▼</td></tr> </table> </div>	Background	Color	▼	Color	#80FF80		Picture		▼	str	AMSON		strFontLib	1	▼	strColor	#0080FF		align	CC	▼									
Background	Color	▼																													
Color	#80FF80																														
Picture		▼																													
str	AMSON																														
strFontLib	1	▼																													
strColor	#0080FF																														
align	CC	▼																													
<p>Number</p>	<p>The number can be integer or float. Note that the number of Float digits must be the same while calculating. E.g., if the value of obj.0 is 20.12, there must be 2 numbers following the decimal point upon calculation.</p> <div style="display: flex; align-items: center; justify-content: center;">  <div style="margin: 0 20px; text-align: center;">  </div> <div style="border: 1px solid gray; padding: 5px;"> <p>Event Setup</p> <p>Event: Press ▼</p> <pre>obj0.num -= 0.01;</pre> <div style="text-align: center; margin-top: 10px;">  </div> </div> </div>																														
<p>Progress Bar</p>	<p>User can set the Background & PColor to make a simple color progress bar.</p> <div style="display: flex; align-items: center; justify-content: center;">  <table border="1" style="margin-left: 20px;"> <tr><td>Background</td><td>Color</td><td>▼</td></tr> <tr><td>Color</td><td>#80FFFF</td><td></td></tr> <tr><td>Picture</td><td></td><td>▼</td></tr> <tr><td>Progress</td><td>Color</td><td>▼</td></tr> <tr><td>PColor</td><td>#0080FF</td><td></td></tr> <tr><td>PPicture</td><td></td><td>▼</td></tr> <tr><td>Orientation</td><td>Horizontal</td><td>▼</td></tr> <tr><td>Maximum</td><td>100</td><td></td></tr> <tr><td>Minimum</td><td>0</td><td></td></tr> <tr><td>Value</td><td>66</td><td></td></tr> </table> </div> <p>(See next page.)</p>	Background	Color	▼	Color	#80FFFF		Picture		▼	Progress	Color	▼	PColor	#0080FF		PPicture		▼	Orientation	Horizontal	▼	Maximum	100		Minimum	0		Value	66	
Background	Color	▼																													
Color	#80FFFF																														
Picture		▼																													
Progress	Color	▼																													
PColor	#0080FF																														
PPicture		▼																													
Orientation	Horizontal	▼																													
Maximum	100																														
Minimum	0																														
Value	66																														

For a different design, user should prepare a full-resolution picture and locate the progress bar at the corresponding coordinates, choosing certain color for the Background and Pt. Picture for the Progress bar.



Background	Color
Color	#FFFFFF
Picture	
Progress	Pt. Picture
PColor	
PPicture	240320 ...
Orientation	Horizontal
Maximum	100
Minimum	0
Value	41

User can set the Background & SColor to make a simple color slide bar.

Slide Bar



Background	Color
Color	#CACACA
Picture	
Slider	Color
SColor	#008000
SPicture	slider
Orientation	Horizontal
Width	30
Maximum	100
Minimum	0
Value	60

Like Button, user can set up the program in the **Event** column,.

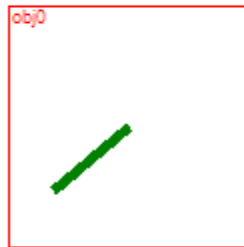
(See next page.)

For the other design, user can also choose Ext. Picture to combine specific Background and Slider.



Background	Ext. Picture ▾
Color	
Picture	slider bar ▾
Slider	Ext. Picture ▾
SColor	
SPicture	slider ▾
Orientation	Horizontal ▾
Width	40
Maximum	100
Minimum	0
Value	60

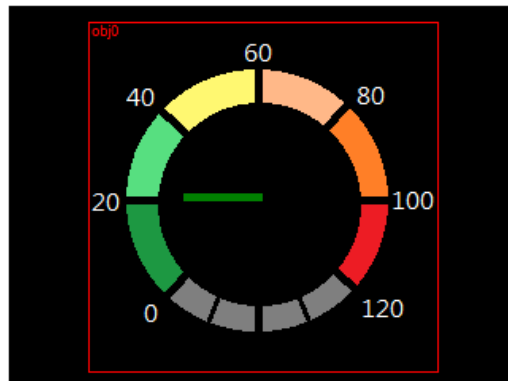
User can set the Background & GColor to make a simple color gauge.



Background	Color ▾
Color	#FFFFFF
Picture	▾
Gauge	Color ▾
GColor	#008000
Length	50
Thickness	5
Value	320

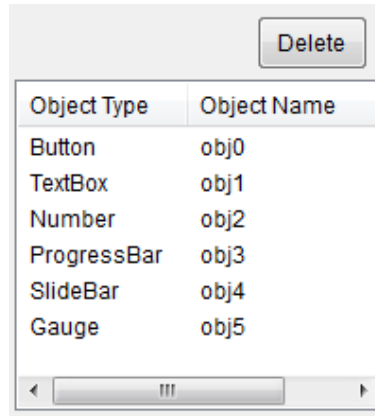
Gauge

For the other design, user can choose Ext. Picture, transparent or Pt. Picture for the Background.



Background	Ext. Picture ▾
Color	
Picture	Gauge ▾
Gauge	Color ▾
GColor	#008000
Length	50
Thickness	5
Value	0

There's a box on the top right corner, in which user can pick the corresponding object to arrange the settings.



7. Programming – All Case-Sensitive

HEX Command

Function	Protocol Header (4Bytes)	Reserve (1Byte)	Command Type (1Byte)	Data Length (1Byte)	Data (n Bytes)	CRC (2Bytes)
Adjust Backlight PWM	FELP	0x00	11	2	0xA8+0x00(0%=off)~0x0A(100%) e.g., change to 10% 0x46,0x45,0x4C,0x50,0x00,0x0B,0x02,0xA8,0x01 PWM adjustment range, total10 levels: 0x01=10%; 0x05=50%; 0x0A=100%	CRC 2Bytes
Adjust Backlight PWM Response	FELP	0x00	11	1	0x00=succeed; Otherwise failed.	CRC 2Bytes
LCD Display On/ off	FELP	0x00	12	1	0x00=LCD Display off. 0x46,0x45,0x4C,0x50,0x00,0x0C,0x01,0x00 0x01=LCD Display on. 0x46,0x45,0x4C,0x50,0x00,0x0C,0x01,0x01	CRC 2Bytes
LCD Display On/ off Response	FELP	0x00	12	1	0x00=succeed; 0x01=failed.	CRC 2Bytes
Shift UI	FELP	0x00	36	Length of following data	UI name length + UI name	CRC 2Bytes
Shift UI Response	FELP	0x00	36	1	0x00=succeed; Otherwise failed.	CRC 2Bytes
Change Attribute of Object str	FELP	0x00	37	Length of following data	UI name length +UI name + Object name length + Object name + str length + str	CRC 2Bytes
Change Attribute of Object str Response	FELP	0x00	37	1	0x00=succeed; Otherwise failed.	CRC 2Bytes

Example of hex command: Command Type = 37, change contains of str in object name "STULN1" in UI name "standby1" to "ABC". Then the host needs to send following data to LCD module through UART.

Header FELP (4 bytes)	Reserve 0x00 (1 byte)	Command 37 (1 byte)	Data length (1 byte)	UI name length (1 byte)*	UI name: standby1 (8 bytes)*	Object name length (1 byte)*	Object name: STULN1 (6 bytes)*	str length (1 byte)*	str: ABC (3 bytes)*	CRC (2 bytes)
0x46,0x45,0x4C,0x50	0x00	0x25	0x14	0x08	0x73,0x74,0x61,0x6E,0x64,0x62,0x79,0x31	0x06	0x53,0x54,0x55,0x4C,0x4E,0x31	3	0x41,0x42,0x43	0xB1,0x11

Remark: Data length is accumulated by those items with star * marks. Hence, 1+8+1+6+1+3 = 20 (hex number: 0x14) is Data Length.

The CRC is calculated base on following code

```

const unsigned short CRC_VALUE=10;
unsigned short crc16_compute(unsigned char const * p_data, unsigned int size, unsigned short const * p_crc)
{
    unsigned short crc = (p_crc == NULL) ? 0xFFFF : *p_crc;
    unsigned int i;
    for ( i = 0; i < size; i++)
    {
        crc = (unsigned char)(crc >> 8) | (crc << 8);
        crc ^= p_data[i];
        crc ^= (unsigned char)(crc & 0xFF) >> 4;
        crc ^= (crc << 8) << 4;
        crc ^= ((crc & 0xFF) << 4) << 1;
    }
    return crc;
}

```

Usage Example:

```

unsigned short crc16_result = crc16_compute(COMM_buff, count, &CRC_VALUE);

```

//Where count is amount of bytes from Header to last data, for the "Example of Hex Command" shown above, it is from Header to str, the total data length is 27 bytes, hence count=27.

Char/ ASCII Command

Function	Protocol Header (4Bytes)	Reserve (1Byte)	Command Type (1Byte)	Data Length (1Byte)	Data (n Bytes)	CRC (2Bytes)
Char/ ASCII Command	FELP	0x00	127	Length of following data	Char/ ASCII Command, ended with 0x00. Refer to the list below.	CRC 2Bytes
Char/ ASCII Command Responds	FELP	0x00	127	1	0x00=succeed; Otherwise failed.	CRC 2Bytes

Char/ ASCII Command List:

Function	Example	Description
Switch UI	form main	Switch to the UI named "main"
Change str value of an object.	obj1.str=test	Change the str value of the object named "obj1" to "test".
Change color of str of an object	obj1.fcolor=#ff0000	Change the str color of the object "obj1" to color #ff0000.
Change background type of an object.	obj1.btype=pic obj1.btype=ppic obj1.btype=bcolor obj1.btype=null	pic: Change background type of the object "obj1" to Picture. The picture size must be the same dimension of the object. ppic: Change background type of the object "obj1" to Portion of a Picture. bcolor: Change background type of the object "obj1" to Color. null: Change background type of the object "obj1" to Transparent.
Change background picture of an object.	obj1.pic=cat	Change background picture of the object "obj1" to the picture named cat where cat needs to exist in the project, and background type of "obj1" must be pic.
Change background color of an object.	obj1.bcolor=#ff0000	Change background color of the object "obj1" to color #ff0000 (Red). The background type of "obj1" needs to be bcolor.
Change the display location of an object.	obj1.x=160 obj1.y=82 obj1.w=20 obj1.h=43	Change x coordinate of the object "obj1" to 160. Change y coordinate of the object "obj1" to 82. Caution: x, y is based on upper left corner of an object. Change the coordinate improperly may cause the object to be out of LCD display range. Change width of the object "obj1" to 20 pixels. Change height of the object "obj1" to 43 pixels.

Remark: Color #ff0000 is hex color code (Red), which is based on RGB color value.

Tips on changing object with different size picture-

While the background of an object is set as picture, the size of the selected picture must be exactly the same. For example, for object size: 20*50 pixels, the picture size needs to be exactly 20*50 pixels. If user wants to change background picture of an object, he/ she should change background type to null first, then change object width/ height to exactly the same as new picture size. Then set background picture to new picture name and finally set background type to picture.

Following example shows the process to change the object named "NUM12" to a new picture named "TinyCar_Red", and its size is 51*51. Each step means sending out the string to LCD Module through UART.

Step 1: NUM12.btype=null

Step 2: NUM12.w=51

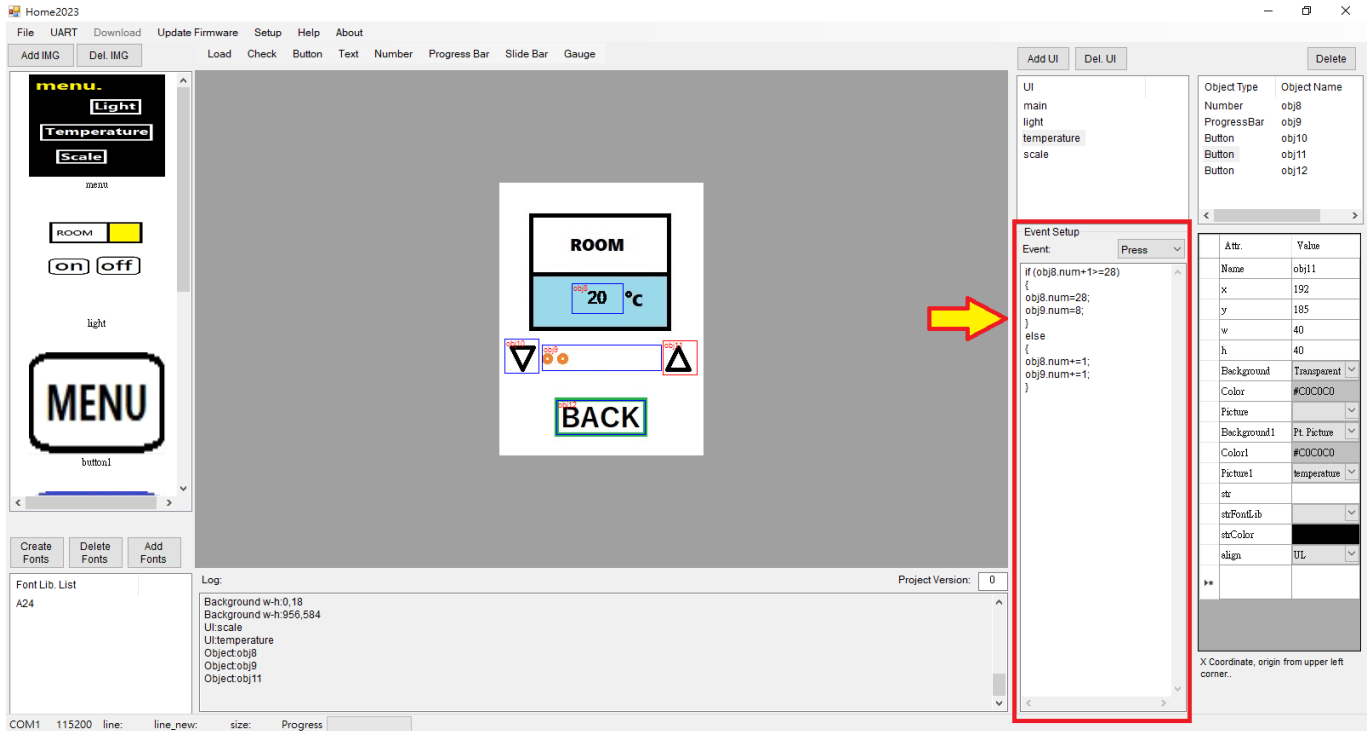
Step 3: NUM12.h=51

Step 4: NUM12.pic=TinyCar_Red

Step 5: NUM12.btype=pic

Keyword in Code

User can implement C-Like Language code in LCD module to run user's designed process. Following table lists keywords in the code. Code needs to be ended with semi-column ";" for each line except condition statement (if else, while, and, for.)



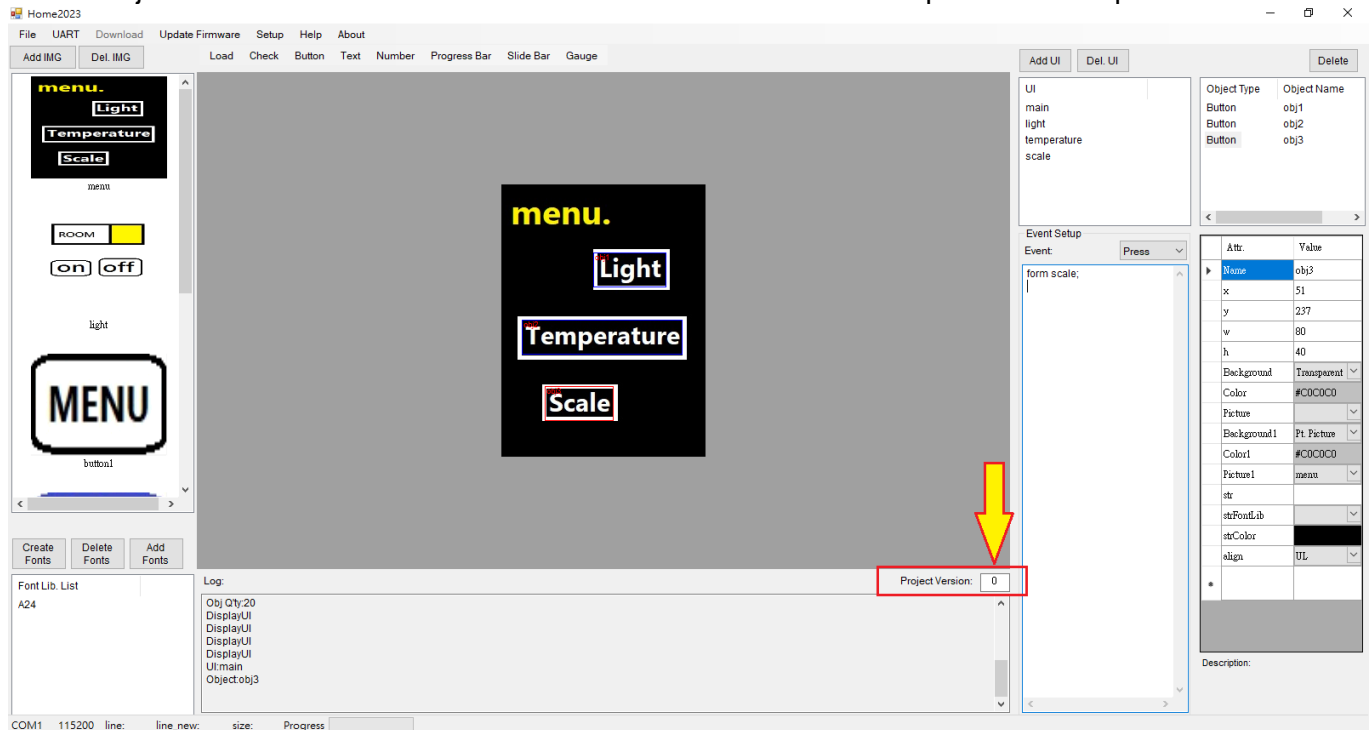
Keyword	Variable Type	Assignable	Return Type	Description
obj.str	string	V	char	Object named "obj" and its content is treated as string.
Examples: TB2.str += "W"; TB2.str = "Hello World"; TB2.str -= 1; TB2.str = TB2.str - 2;				Description: Append string "W" to object TB2. If TB2.str contains "H", TB2.str becomes "HW" after TB2.str += "W"; Assign string "Hello World" to TB2.str. Delete last character in the TB2.str. Delete last 2 characters in the TB2.str.
obj.num	integer/ float	V	int32 or float	Object named "obj" and its content is treated as integer (1, 2, -3...) or float (-1.2, 2.54...), note that the sign "+" shouldn't be used. However, user can set the sign "-" to make a negative number.
Examples: NU2.num -= 1; NU3.num = TB3.num +2.00;				Description: Set number of object NU2 to subtract 1. If NU2 is 4, it becomes 3 after this operation. Set number of object NU3 as TB3+2.00. If TB3 is 8.00, NU3 becomes 10.00 after this operation. Float digit must have same digit count set in object following decimal point (the number of digits after decimal point of NU3, TB3 must be the same). For example, all numbers have two digits after decimal. NU3 needs to be 0.00 (two digits after decimal) TB3 needs to be 0.00 (two digits after decimal) The number 2 needs to be 2.00 (two digits after decimal)
obj.len	integer	NA	int32	Get the current string length (in byte) of the object named "obj".
Example: if (TB2.len > 5)...				Description: To judge in conditional routing. If obj.str = "ABCDEF", obj.len returns 6 in 4 bytes length. E.g., 0x06, 0x00, 0x00, 0x00 For numbers type of object, Number/ Progress Bar/ Slide Bar/ Gauge object, the returned data for this operate will always be 4 bytes sent by LSB order, 0x04, 0x00, 0x00, 0x00 For example, if the string length is 258 bytes, the returned data is 0x02 0x01 0x00 0x00. This demo what LSB means, the string length cannot be over 31 characters for an object.

obj.fcolor	string	V	string	Get/ set the font color (as string) of the object named "obj".
Example: TB2.fcolor = "#FF00FF";		Description: Assign the TB2 string color to "#FF00FF"		
obj.btype	string	V	string	Get/ set the background type (as string) of the object named "obj".
Example: TB2.btype = "bcolor";		Description: Change TB2 background type to bcolor (use color as background). Options are listed below- [bcolor] Use color as background; [pic] Use extended picture as background; [null] Use background as background (transparent); [ppic] Use portion of picture as background.		
obj.pic	string	V	String	Get/ set the pic (as string) of the object named "obj".
Example: TB2.str = OBJ1.pic;		Description: Assign the picture name of background of OBJ1 to TB2 string. If the name of background picture of OBJ1 is TinyCar, the TB2.str becomes "TinyCar" after this operation.		
obj.bcolor	string	V	string	Get/ set the background color (as string) of the object named "obj".
Example: TB2.str = OBJ1.bcolor;		Description: Assign the RGB color value of background of OBJ1 to TB2 string. If the RGB color value of background of OBJ1 is #FF00FF, the TB2.str becomes "#FF00FF" after this operation.		
obj.x, obj.y	integer	V	int32	Get/ set the coordinates of the object named "obj".
Example: TB2.x=160; TB2.y=30;		Description: Assign the location of object TB2 to coordinate x=160 pixel, and y=30 pixel. The origin of an object is on the top-left corner of that object. Be careful on changing the location of object to avoid the object area is out of LCD display area.		
obj.w, obj.h	integer	V	int32	Get/ set the width/ height of the object named "obj".
Example: TB2.w=160; TB2.h=25;		Description: Assign the width of object TB2 to 160 pixels, and h=25 pixels. The origin of an object is on the top-left corner of that object. Be careful in changing the size of object to avoid that the object area is out of LCD display area. If object's background is extended picture, changing the size of object causes error if the changed size doesn't match the size of existing extended picture.		
send	command	NA	NA	Send data out through UART port.
Example: (Send out as string.) send "Text"; send BT2.str; (Send out as number) send 513; send -300; send BT2.num;		Description: String types are str, fcolor, bcolor, and pic. Send out string "Text" through UART. Send out string content of BT2.str. If BT2.str contains "STOP", string "STOP" is sent out through UART. Number types are num, x, y, w, h, and len. Send out integer number 513 (int32, LSB format) through UART. The 513 will be sent as 0x01,0x02,0x00,0x00. Send out integer number -300 (int32, LSB format) through UART. The negative value will be sent as 1's complement value, e.g., -300 will be sent as 0xD4,0xFE,0xFF,0xFF Send out content of object BT2.num, it could be integer (int32, LSB format) or float value. The float value is IEEE-754 Floating Point with hexadecimal representation. E.g., if float number is 2.58, it's sent as 0xB8,0x1E,0x25,0x40 through UART.		
form	command	NA	NA	Change to UI page.
Example: form main;		Description: Change current displayed UI to the UI named "main".		
if else	command	Conditional statement.		
while	command	Conditional statement.		
for	command	Conditional statement.		

8. Update Project with SD Card

- After downloading the project to the LCD module, a file named file.hex will be generated in the path of the UI Editing Tool.
- To be able to update project with SD card, user must have a LCD module in hand so as to download the project and get the file.hex.
- Copy the file.hex to the root directory of the SD card in the computer. After that, unplug the SD card from the computer and insert it into the LCD module. The screen will automatically show the current project version in module and the project version in the SD card. It shows nothing if there's no file.hex in the SD card.
- If user decides to update the project, please remove and insert the SD card within 30 seconds.
- After the project update is completed, remove the SD card. Then the first UI of the new project will be shown on the LCD module.
- If user decides not to update the project, please do nothing within 30 seconds. Then remove the SD card to restart the previous project.

****The Project Version can be set below the screen. Please refer to the position in the picture below.**



If you have any questions about this manual, feel free to contact us anytime.