



TM57 系列

TM57FLA80 DEMO CODE
FOR LCD FUNCTION

Application Note

**Tenx reserves the right to change or
discontinue this product without notice.**

tenx technology inc.

CONTENTS

PRODUCT NAME	2
TM57 系列IC	2
TITLE	2
TM57FLA80 DEMO CODE FOR LCD FUNCTION	2
APPLICATION NOTE	2
01. DEMO功能介绍	2
1. 具体DEMO程序请参考 TM57FLA80_LCD.ASM	2
2. DEMO功能说明（如下图片）	2
3. 线路图	3
02. LCD初始化设置方法	3
1. 设置LCD的驱动基础时钟	3
2. 选择合适的COM及SEG端口	5
3. 清LCD显示RAM	5
4. 使能LCD并配置适当的DUTY、BAIS及LCDBRIT	5
03. LCD显示RAM的配置方法	7
1. 显示RAM与COM和SEG的关系	7
2. 编写LCD接口函数	8
04. LCD模拟面板图的制作及使用方法	14
1. LCD模拟面板图的制作	14
2. LCD模拟面板图的使用方法	25

PRODUCT NAME

TM57 系列 IC

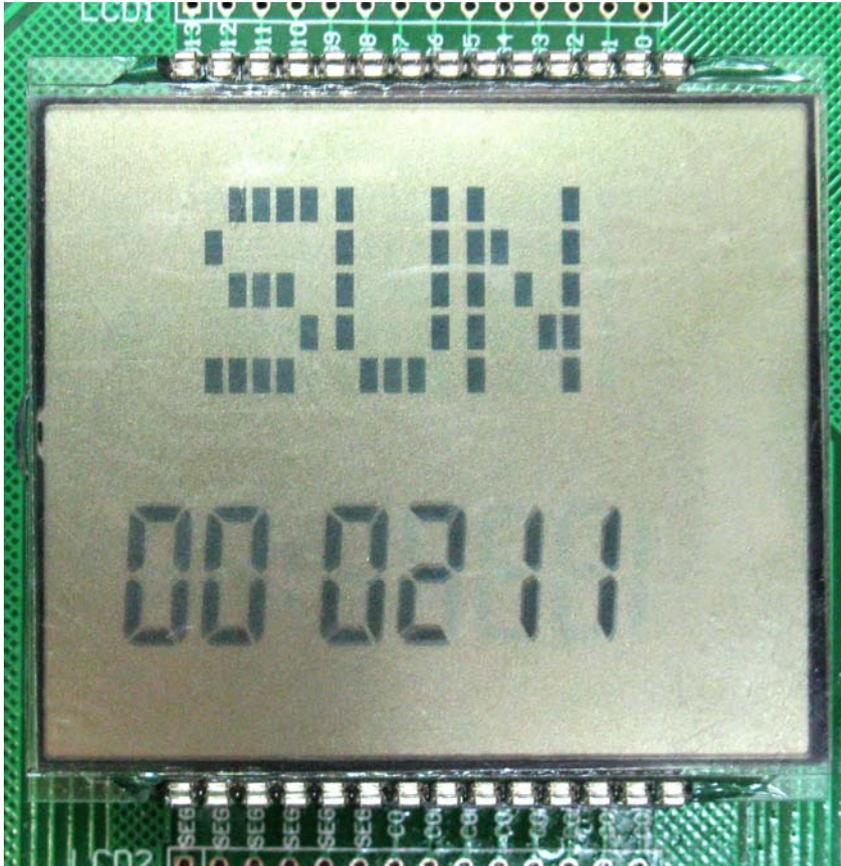
TITLE

TM57FLA80 DEMO CODE FOR LCD FUNCTION

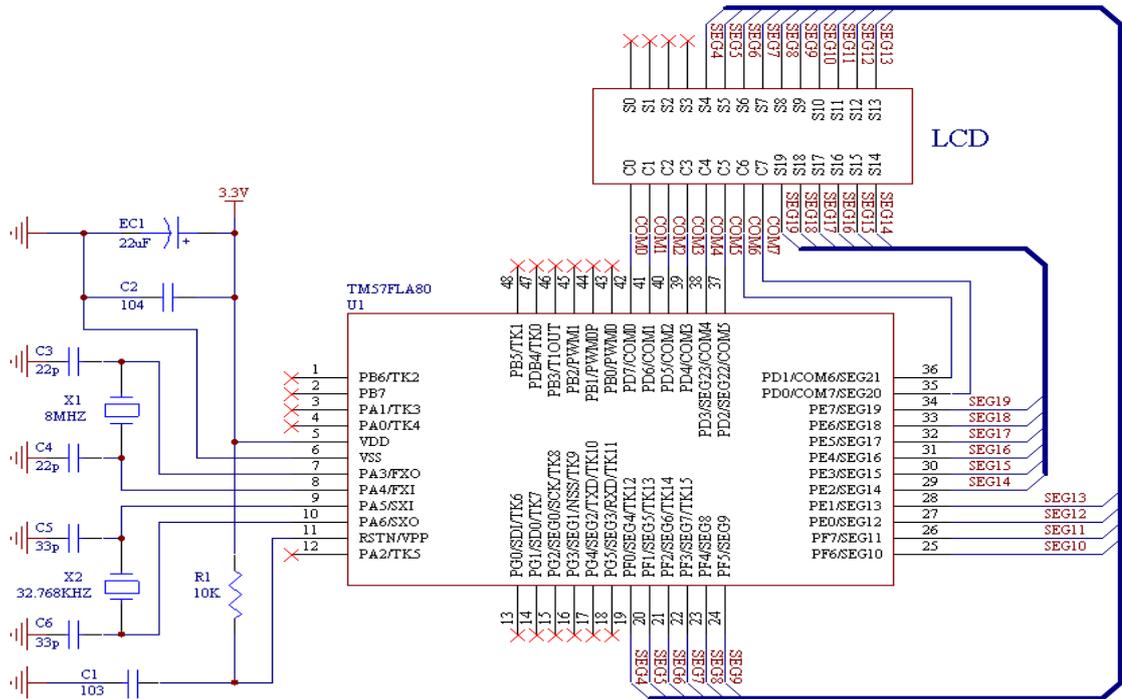
APPLICATION NOTE

01. DEMO 功能介绍

1. 具体 DEMO 程序请参考 TM57FLA80_LCD.ASM
2. DEMO 功能说明（如下图片）
 - 1) 显示“SUN”字符。
 - 2) 在“SUN”字符下方显示时钟，包括时钟（00~23）、分钟（00~59）、秒钟（00~59），显示数据随时间的变化而变化。所使用的 LCD 参数为 1/8 Duty ,1/3 Bias, 工作电压 3.0V



3. 线路图



02. LCD 初始化设置方法

1. 设置 LCD 的驱动基础时钟

LCD 的驱动基础时钟与 TM2 相关，用户可以在 TM2CTRL 寄存器里面设置，其中 TM2CTRL 的地址是 R 区的 0DH。

首先选择 TM2 的时钟源，如果将 TM2CTRL 的 BIT4 置 1，那么 TM2 的时钟源为 $F_{osc}/128$ (F_{osc} 为系统时钟)；否则如果清零，时钟源则为系统子时钟。

然后选择 LCD 的 alternative frequency (设定 LCDCLK register)，有 4 种选择，由 TM2CTRL 的 BIT[1:0] 两个位控制：

- BIT[1:0]=00 LCDCLK=128, LCD alternative freg=system sub-clk/128
- BIT[1:0]=01 LCDCLK=64, LCD alternative freg=system sub-clk/64
- BIT[1:0]=10 LCDCLK=32, LCD alternative freg=system sub-clk/32
- BIT[1:0]=11 LCDCLK=16, LCD alternative freg=system sub-clk/16

举个例子，假如 TM2 的时钟源为系统子时钟，且系统子时钟的频率是 32.768KHz，LCD 的 DUTY 为 1/8，客户希望使用的 LCD frame frequency 为 32Hz，也就是说 LCD 扫描一周的时间为 31.25 毫秒，根据公式：

$$(\text{LCD 扫描一周的时间}) = 2 \times [(1/\text{系统子时钟频率}) \times (\text{LCDCLK})] \div (\text{LCD 的 DUTY})$$

得到：

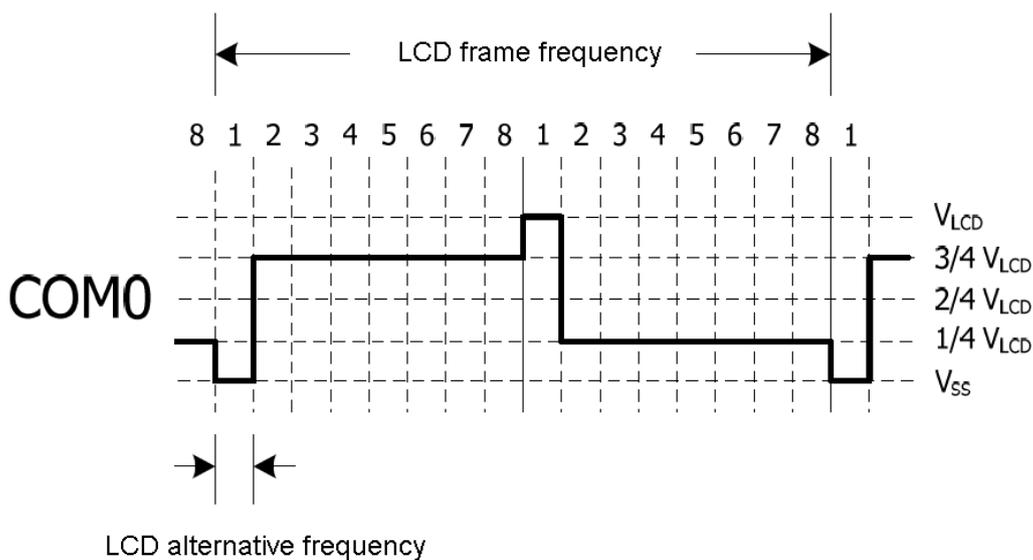
$$\begin{aligned} (\text{LCDCLK}) &= (\text{LCD 扫描一周的时间}) \times (\text{LCD 的 DUTY}) \times (\text{系统子时钟频率}) \div 2 \\ &= (31.25 \times 0.001) \times (1/8) \times (32.768 \times 1000) \div 2 \\ &= 64 \end{aligned}$$

于是得出 TM2CTRL 的 BIT[1:0]=01，所以执行的指令如下：

```
movlw    TIMER2CLK_subClock    ;b' 00000000'
iorlw    LCDCLK_64              ;b' 00000001'
movwrr   TM2CTRL
```

如下是 COM0 端口的波形图，图中 1 Frame 的时间就是 LCD 扫描一周的时间，即 31.25 毫秒，每个栅格代表的时间则是 LCD 驱动基础时钟，也就是：

$$(1/\text{系统子时钟频率}) \times (\text{LCDCLK}) = 1.953125 \text{ 毫秒}$$



2. 选择合适的 COM 及 SEG 端口

COM 及 SEG 端口的选择与 LCD_PU 寄存器（R 区 0Ah）BIT[6:4]三个位有关，用户可以根据下表来设置相应的参数

BIT[6:4]=000	LCDpin_use_NO	;b'00000000'
BIT[6:4]=001	LCDpin_use_com[0~3]andseg[20~23]	;b'00010000'
BIT[6:4]=010	LCDpin_use_com[0~7]andseg[16~19]	;b'00100000'
BIT[6:4]=011	LCDpin_use_com[0~7]andseg[12~19]	;b'00110000'
BIT[6:4]=100	LCDpin_use_com[0~7]andseg[8~19]	;b'01000000'
BIT[6:4]=101	LCDpin_use_com[0~7]andseg[4~19]	;b'01010000'
BIT[6:4]=110	LCDpin_use_com[0~7]andseg[2~19]	;b'01100000'
BIT[6:4]=111	LCDpin_use_com[0~7]andseg[0~19]	;b'01110000'

例如，我们需要用到 COM0~COM7 及 SEG4~SEG19，由于无需用到上拉，因此将 COM 与 SEG 对应的 PD、PE、PF、PG 端口的上拉电阻全部禁止。这样我们可以执行如下指令实现：

```
movlw LCDpin_use_com[0~7]andseg[4~19] ;b'01010000'
iorlw PDU_Disable ;b'00000001'
iorlw PEPU_Disable ;b'00000010'
iorlw PFPU_Disable ;b'00000100'
iorlw PGPU_Disable ;b'00001000'
movwr LCD_PU
```

3. 清 LCD 显示 RAM

LCD 显示 RAM 地址在 R 区的 20H~33H，共 20 个寄存器。将这 20 个寄存器清零的目的就是使 LCD 初始化状态为全灭模式。指令见 DEMO 程序中的 LCDINIT 子程序。

4. 使能 LCD 并配置适当的 DUTY、BAIS 及 LCDBRIT

这项配置与地址为 R 区 11H 的 LCDCTRL 寄存器有关，如下是该寄存器各个位的说明。

BIT[7]控制 LCD 的开和关，1=LCD 开 0=关

BIT[6:5] 控制 LCD 的 DUTY

BIT[6:5] =00	static
BIT[6:5] =01	1/3duty
BIT[6:5] =10	1/4duty

BIT[6:5] =11 1/8duty

BIT[4:3] 控制 LCD 的 BIAS

BIT[4:3]=00 1/2bias

BIT[4:3]=01 1/3bias

BIT[4:3]=1x 1/4bias

BIT[2:0] 控制 LCD 的 LCDBRIT

BIT[2:0] =000 LCDBrightness_(12Div20)VDD
;b'00000000'

BIT[2:0] =001 LCDBrightness_(12Div19)VDD
;b'00000001'

BIT[2:0] =010 LCDBrightness_(12Div18)VDD
;b'00000010'

BIT[2:0] =011 LCDBrightness_(12Div17)VDD
;b'00000011'

BIT[2:0] =100 LCDBrightness_(12Div15)VDD
;b'00000100'

BIT[2:0] =101 LCDBrightness_(12Div14)VDD
;b'00000101'

BIT[2:0] =110 LCDBrightness_(12Div13)VDD
;b'00000110'

BIT[2:0] =111 LCDBrightness_VDD
;b'00000111'

用户可以根据具体的要求来选对应的值填入 LCDCTRL 寄存器，例如，将 LCD 的 DUTY、BIAS、LCDBRIT 分别设为 1/8duty、1/3bias、工作电压 3.0V((12/13)*3.3)，并且使能 LCD 的指令如下：

```
movlw    LCD_Enable           ; b'10000000'
iorlw    Duty_1Div8           ; b'01100000'
iorlw    Bais_1Div3           ; b'00001000'
iorlw    LCDBrightness_(12Div13)VDD ; b'00000110'
movwr    LCDCTRL
```

03. LCD 显示 RAM 的配置方法

1. 显示 RAM 与 COM 和 SEG 的关系

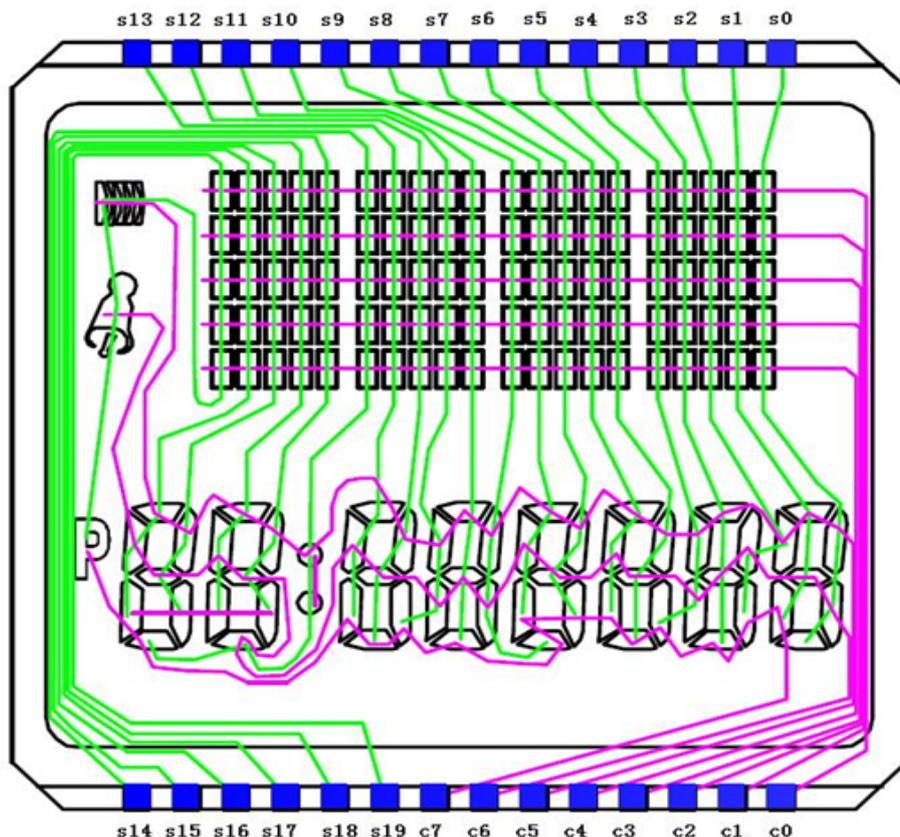
8 COM	COM7	COM6	COM5	COM4	COM3	COM2	COM1	COM0
R-Plane 20	SEG0							
21	SEG1							
22	SEG2							
23	SEG3							
24	SEG4							
25	SEG5							
26	SEG6							
27	SEG7							
28	SEG8							
29	SEG9							
2a	SEG10							
2b	SEG11							
2c	SEG12							
2d	SEG13							
2e	SEG14							
2f	SEG15							
30	SEG16							
31	SEG17							
32	SEG18							
33	SEG19							

显示 RAM 与 COM 和 SEG 的关系图

显示 RAM 与 COM 和 SEG 的关系如上图所示，比如说图中的寄存器 20H 控制着 SEG0，而 20H 的 BIT7~BIT0 这 8 个 BIT 分别控制着 COM7~COM0，也就是说要想使 LCD 面板图中 COM7 与 SEG0 相交的点点亮，那么只要将 20H 的 BIT7 置 1，当然如要灭掉，则只要清零即可。其它的点依此类推。

2. 编写 LCD 接口函数

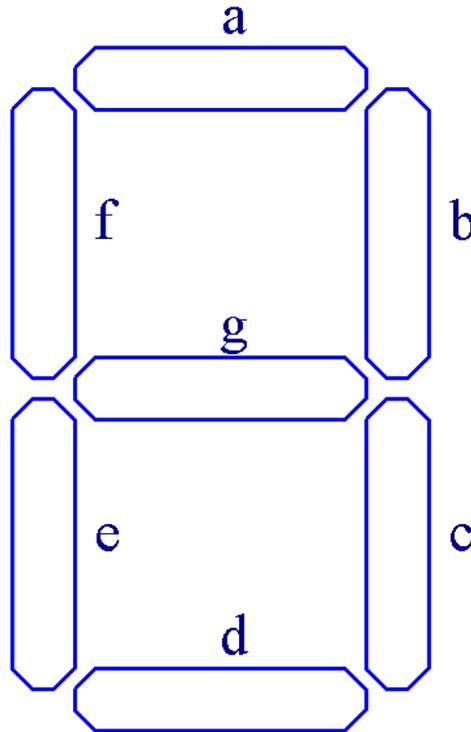
为方便程序的编写，一般点亮 LCD 需要根据特定的面板图来编写相应的接口函数，这里以如下所示的 LCD 面板图为例来说明怎样编写 7 段数码管显示及字符显示的子程序，面板图中 s 表示 SEG（如 s0 表示 SEG0），c 表示 COM（如 c0 表示 COM0）。



LCD 面板图

1) 7 段数码管显示子程序

一个数码管由“a” “b” “c” “d” “e” “f” “g” 7 个部分组成，因此被称为 7 段数码管，其对应位置如下图所示：



LCD 面板图中一共有 8 个 7 段数码管。这里只介绍最左边的数码管，其它 7 个用户可以依此类推。

首先，找出 7 个点的位置，也就是这 7 个点分别是哪个 SEG 和哪个 COM 的交叉点，然后推算出他们在显示 RAM 区的对应的字节位置。结果如下：

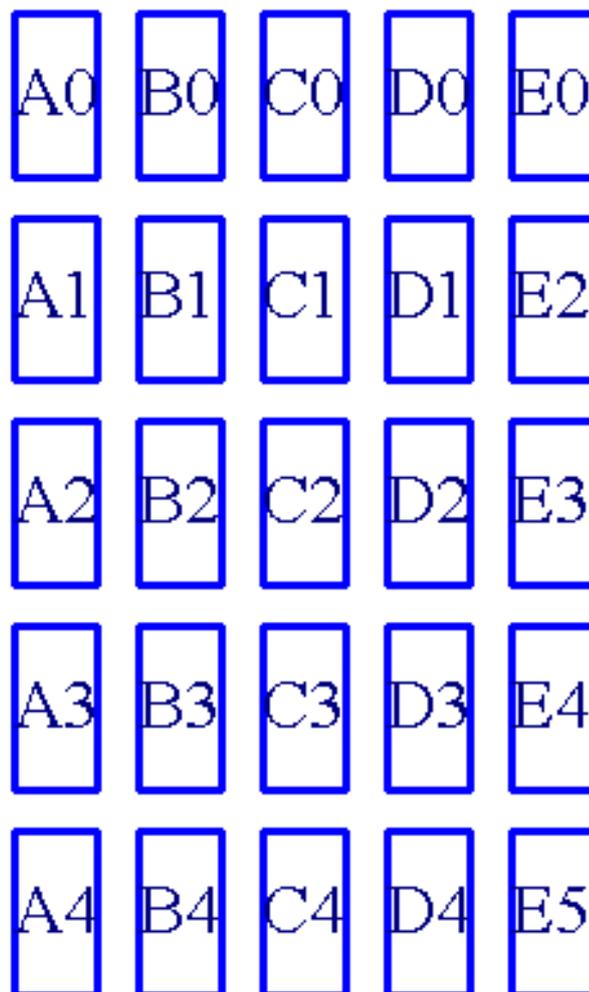
点	交叉	RAM 位置
a	s18 与 c5	32H 的 BIT5
b	s17 与 c5	31H 的 BIT5
c	s17 与 c7	31H 的 BIT7
d	s14 与 c7	2eH 的 BIT7
e	s18 与 c7	32H 的 BIT7
f	s18 与 c6	32H 的 BIT6
g	s17 与 c6	31H 的 BIT6

这样，当这个 7 段数码管要显示 0 时该点亮的是将是 abcdef，而 g 则要熄灭。如下是显示 0 时 a~g 点的亮灭状态：

a	s18 与 c5	32H 的 BIT5=1
b	s17 与 c5	31H 的 BIT5=1
c	s17 与 c7	31H 的 BIT7=1
d	s14 与 c7	2eH 的 BIT7=1
e	s18 与 c7	32H 的 BIT7=1
f	s18 与 c6	32H 的 BIT6=1
g	s17 与 c6	31H 的 BIT6=0

搞清了这些问题，编程就变得相当容易了，至于详细的程序就请用户参照 DEMO 程序中的 nSMG_Dis_1 子程序。

2) 字符显示子程序

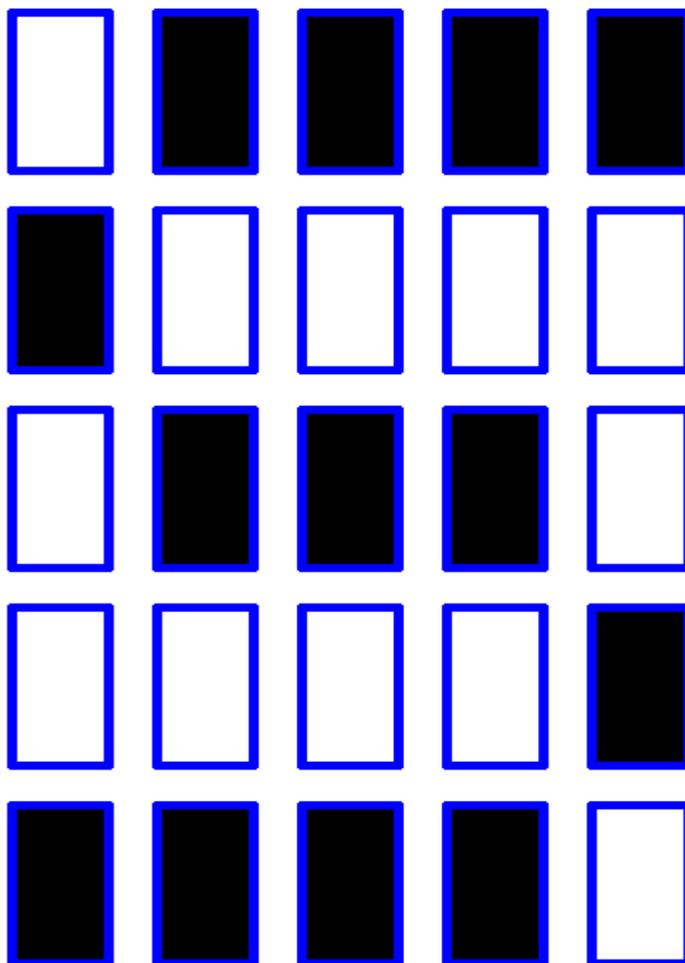


上图是一个字符矩阵各点对应的位置编号图，这里我们以 LCD 面板图中的最左边的一个矩阵为例来讲述字符子程序的编写过程。

首先要找到矩阵中 A0~A4、B0~B4、C0~C4、D0~D4、E0~E4 各点在 LCD 面板图中对应的交叉点及相应 RAM 位置，根据与“7 段数码管显示”相类似的方法，得到如下结果：

点	交叉	RAM 位置
A0	s19 与 c0	32H 的 BIT0
A1	s19 与 c1	32H 的 BIT1
A2	s19 与 c0	32H 的 BIT2
A3	s19 与 c3	32H 的 BIT3
A4	s19 与 c4	32H 的 BIT4
B0	s18 与 c0	31H 的 BIT0
B1	s18 与 c1	31H 的 BIT1
B2	s18 与 c0	31H 的 BIT2
B3	s18 与 c3	31H 的 BIT3
B4	s18 与 c4	31H 的 BIT4
C0	s17 与 c0	30H 的 BIT0
C1	s17 与 c1	30H 的 BIT1
C2	s17 与 c0	30H 的 BIT2
C3	s17 与 c3	30H 的 BIT3
C4	s17 与 c4	30H 的 BIT4
D0	s16 与 c0	2fH 的 BIT0
D1	s16 与 c1	2fH 的 BIT1
D2	s16 与 c0	2fH 的 BIT2
D3	s16 与 c3	2fH 的 BIT3
D4	s16 与 c4	2fH 的 BIT4
E0	s15 与 c0	2fH 的 BIT0
E1	s15 与 c1	2fH 的 BIT1
E2	s15 与 c0	2fH 的 BIT2
E3	s15 与 c3	2fH 的 BIT3
E4	s15 与 c4	2fH 的 BIT4

现在假如要显示 S，根据这个字母的形状特征得出要点亮的部分应为下图中黑体部分，如图所示：



因此得出 A0~A4、B0~B4、C0~C4、D0~D4、E0~E4 各点的状态如下：

点	交叉	RAM 位置
A0	s19 与 c0	32H 的 BIT0=1
A1	s19 与 c1	32H 的 BIT1=0
A2	s19 与 c0	32H 的 BIT2=0
A3	s19 与 c3	32H 的 BIT3=1
A4	s19 与 c4	32H 的 BIT4=0
B0	s18 与 c0	31H 的 BIT0=1
B1	s18 与 c1	31H 的 BIT1=0
B2	s18 与 c0	31H 的 BIT2=1
B3	s18 与 c3	31H 的 BIT3=0

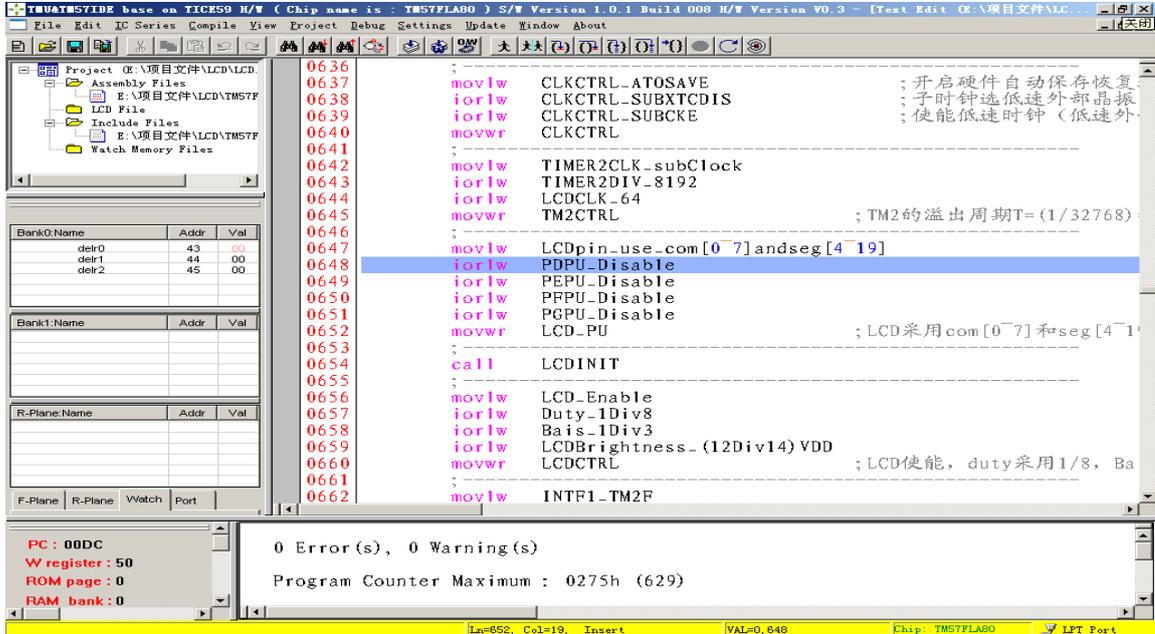
B4	s18 与 c4	31H 的 BIT4=1
C0	s17 与 c0	30H 的 BIT0=1
C1	s17 与 c1	30H 的 BIT1=0
C2	s17 与 c0	30H 的 BIT2=1
C3	s17 与 c3	30H 的 BIT3=0
C4	s17 与 c4	30H 的 BIT4=1
D0	s16 与 c0	2fH 的 BIT0=1
D1	s16 与 c1	2fH 的 BIT1=0
D2	s16 与 c0	2fH 的 BIT2=1
D3	s16 与 c3	2fH 的 BIT3=0
D4	s16 与 c4	2fH 的 BIT4=1
E0	s15 与 c0	2fH 的 BIT0=0
E1	s15 与 c1	2fH 的 BIT1=1
E2	s15 与 c0	2fH 的 BIT2=0
E3	s15 与 c3	2fH 的 BIT3=0
E4	s15 与 c4	2fH 的 BIT4=1

这样，当需要显示 S 时，只需按如上要求将这 25 个点的状态设置好既可。到此，后面的程序部分就请用户参见 DEMO 程序中的 WRITE_dat_1 子程序了。

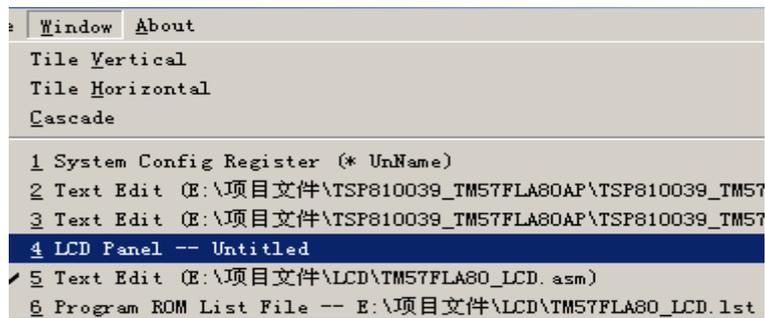
04. LCD 模拟面板图的制作及使用方法

1. LCD 模拟面板图的制作

1) 打开工程界面，如下图所示：

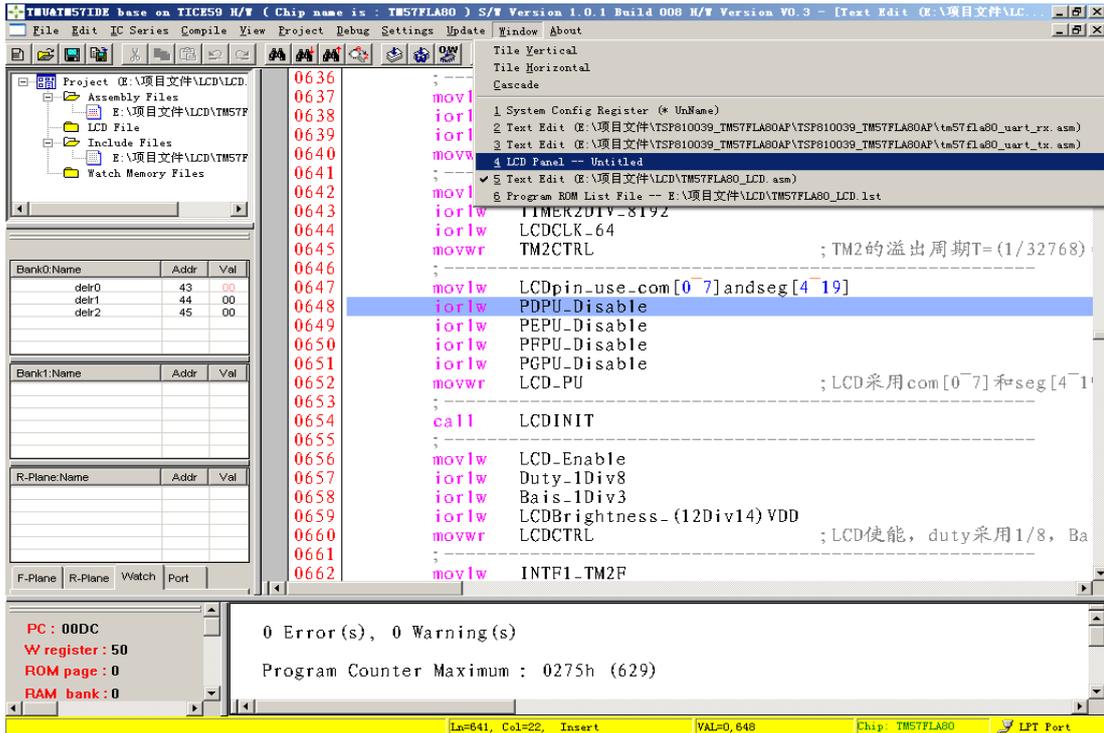


2) 用鼠标左键点选

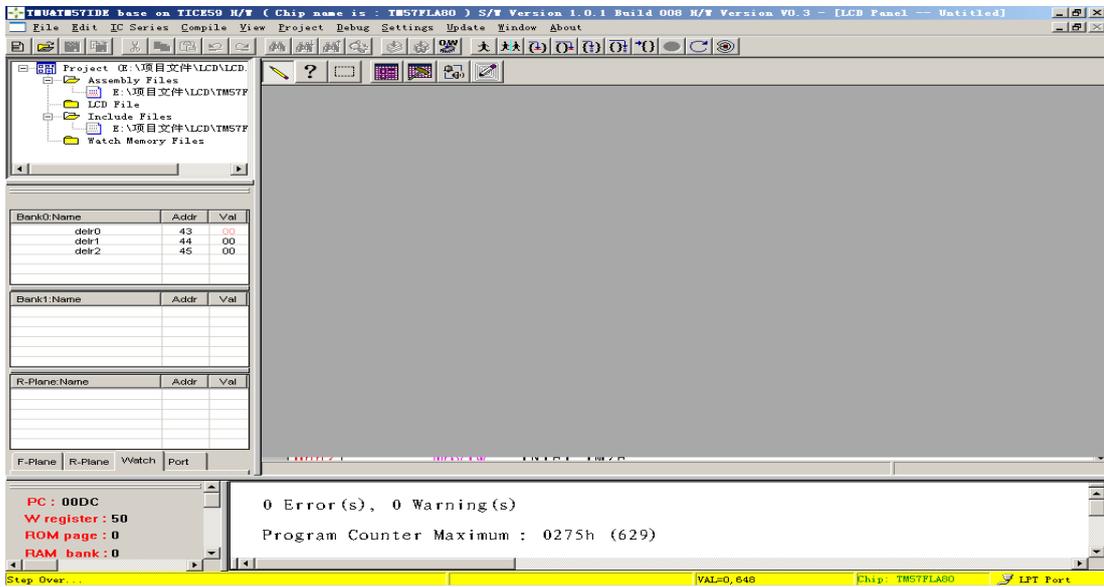


，如下

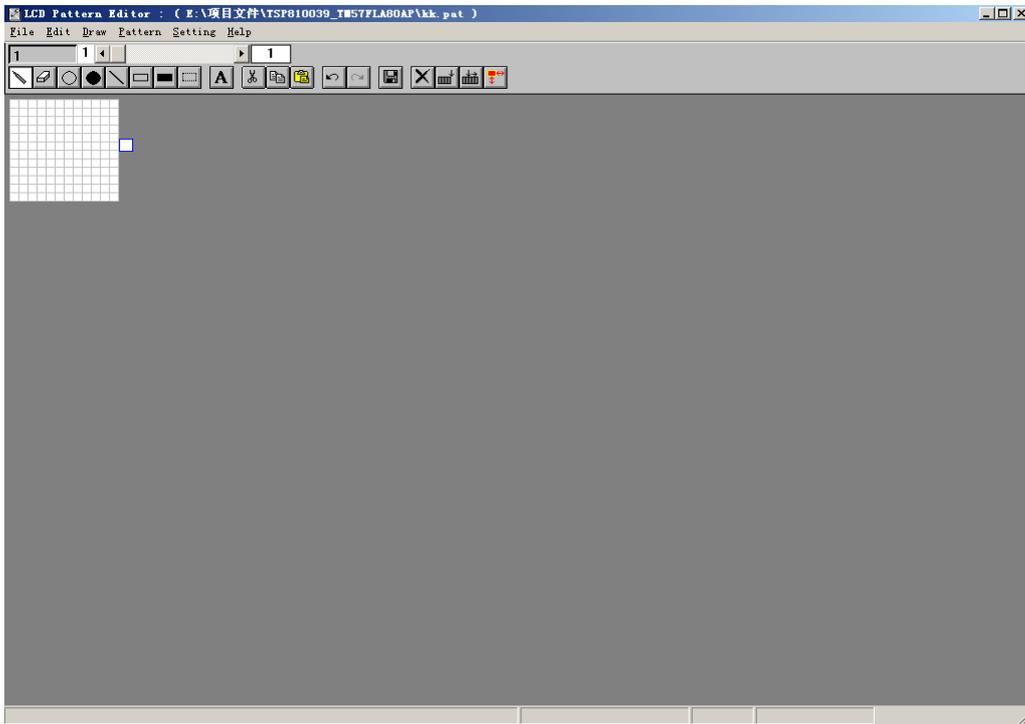
图：



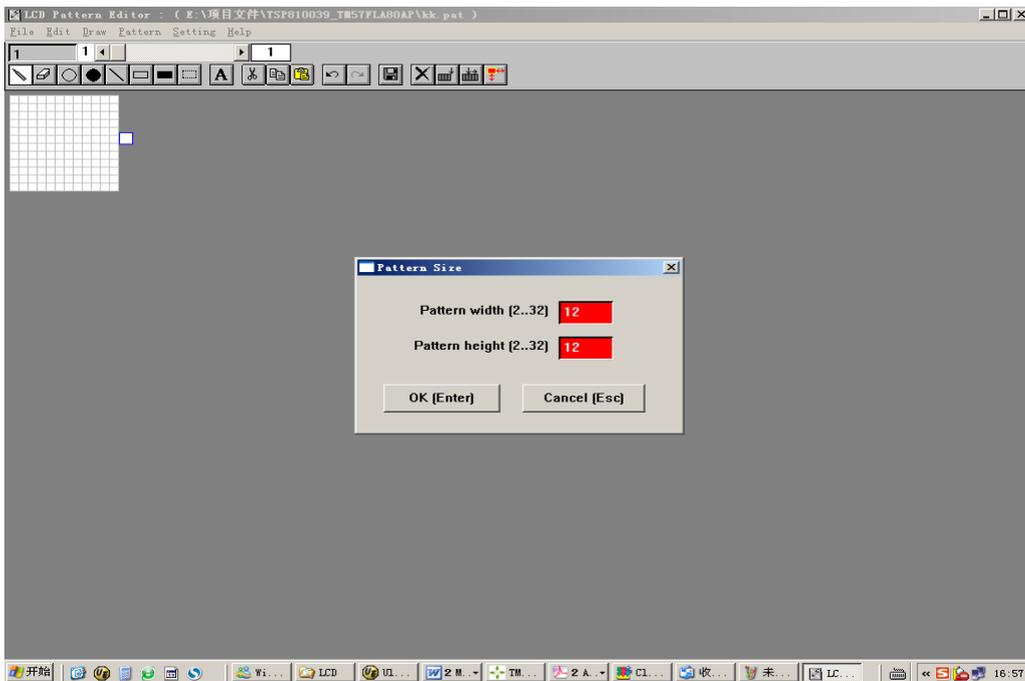
3) 然后单击鼠标左键，工程界面将打开 LCD Panel 操作界面，见下图：



4) 鼠标单击  按钮，进入 LCD Panel 库编辑界面，如下图：



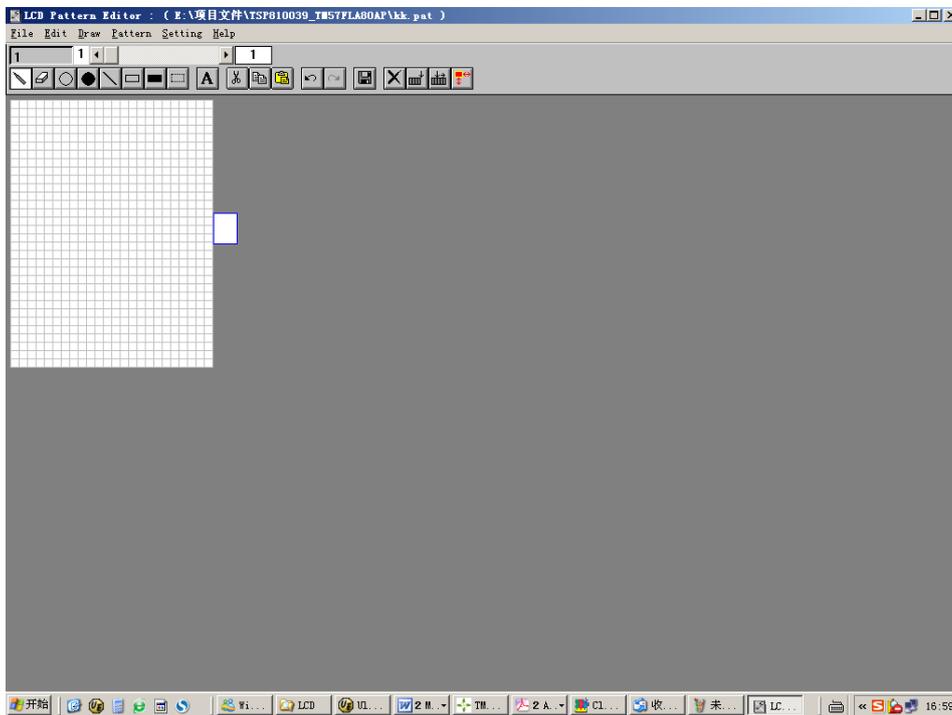
5) 鼠标单击  按钮，将弹出如下对话框：



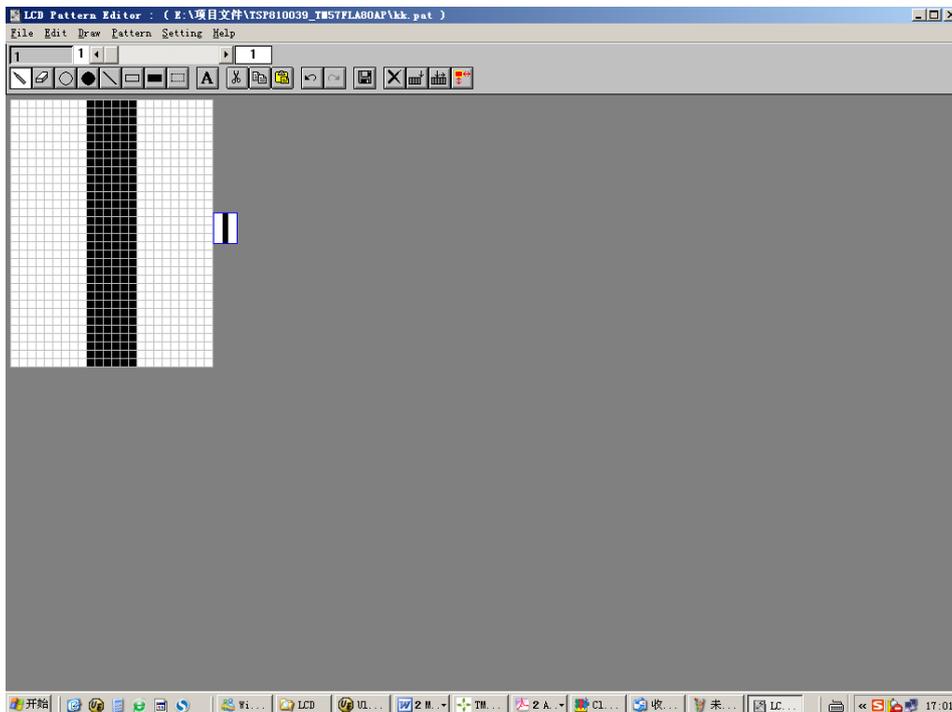
- 6) 对话框里的格点数据 width 和 height 均为分别改为 24 和 32，然后单击



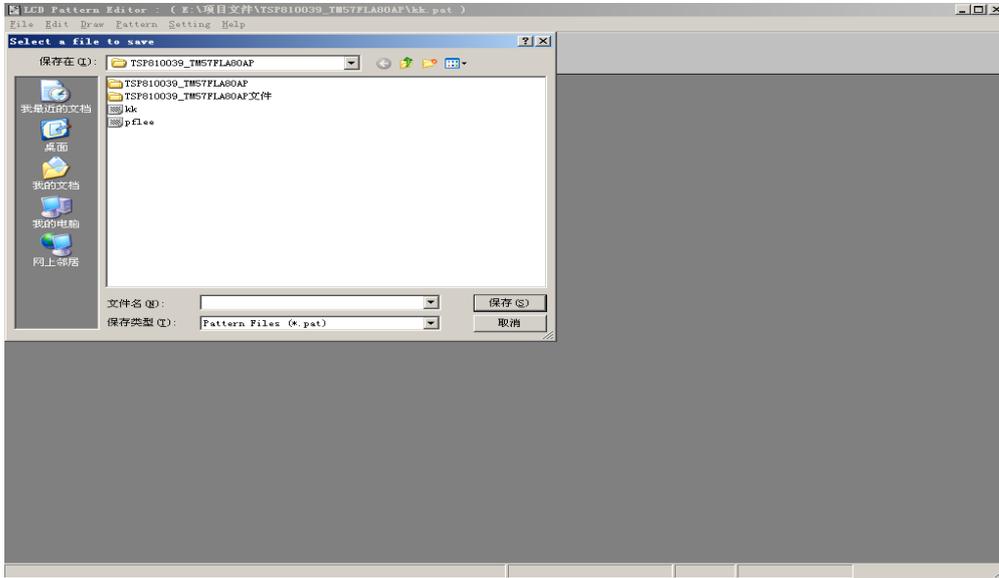
按钮。观察界面的变化，如下图：



- 7) 鼠标单击  按钮，用户将可在库编辑界面里编辑所需要的图形，编辑的图形如下：

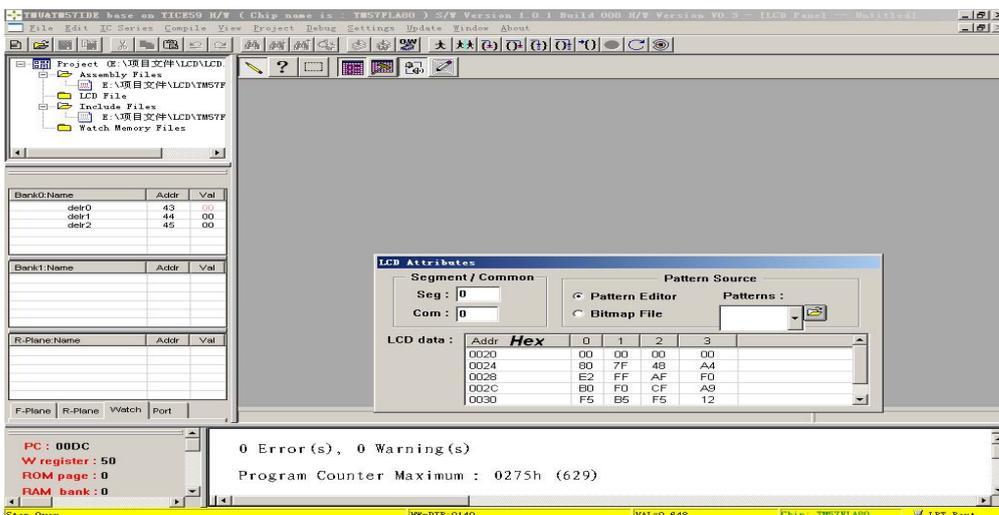


- 8) 单击编辑界面的  保存按钮，那么所画的图形将保存下来。如果是首次编辑，那么单击  保存按钮时将会弹出如下对话框：

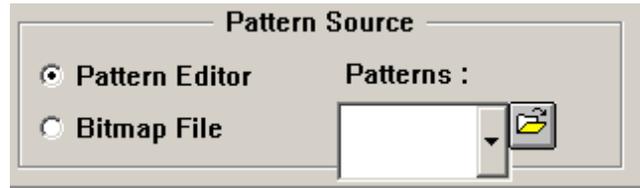


- 9) 在上述对话框里选择文件存储的路径及文件名，单击保存后，这个所画的图形就存储在制定的文件里了。
- 10) 如果需要再画一个图形，只要单击  按钮，并重复 7、8、9 步即可，当然如果需要删除图形，只要单击  按钮。

- 11) 单击  按钮，将弹出如 LCD attribute 话框：



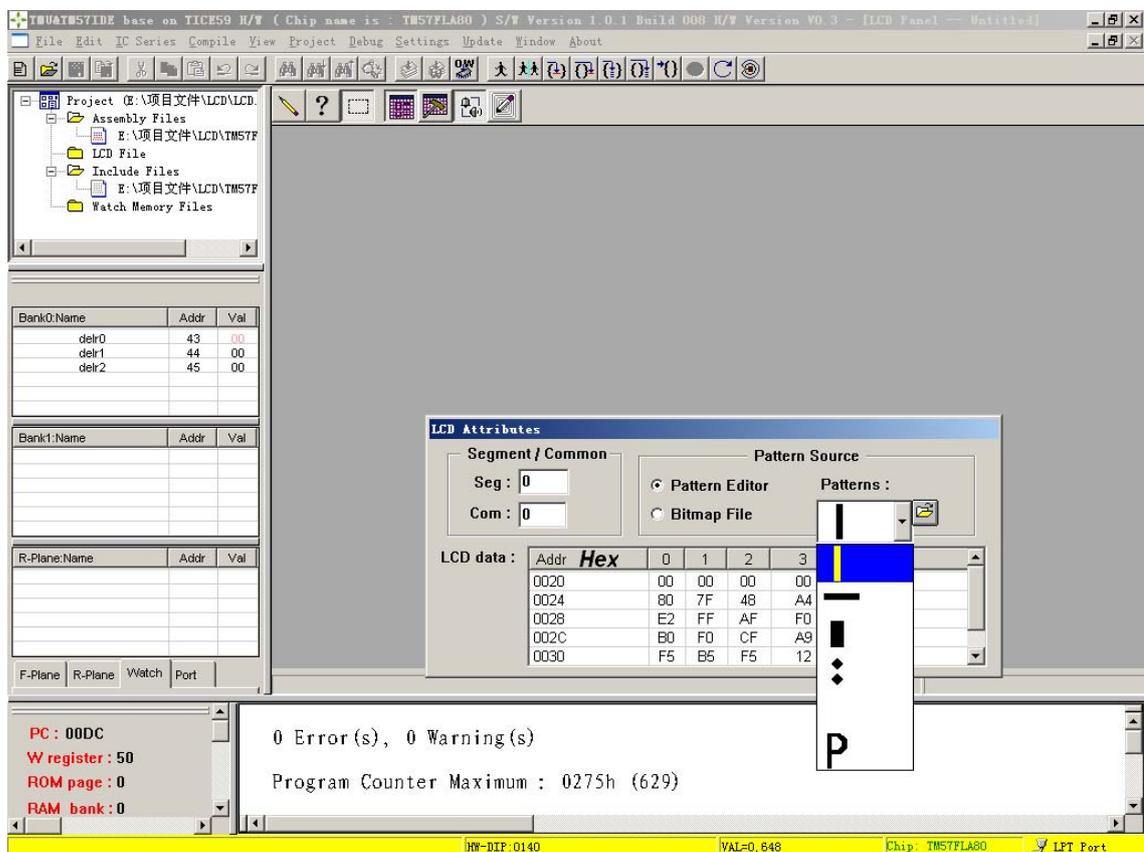
12)选中“Pattern Editor的项目”



，使其处于所示的状态，然后单击按钮，将先前保存的图形文件载

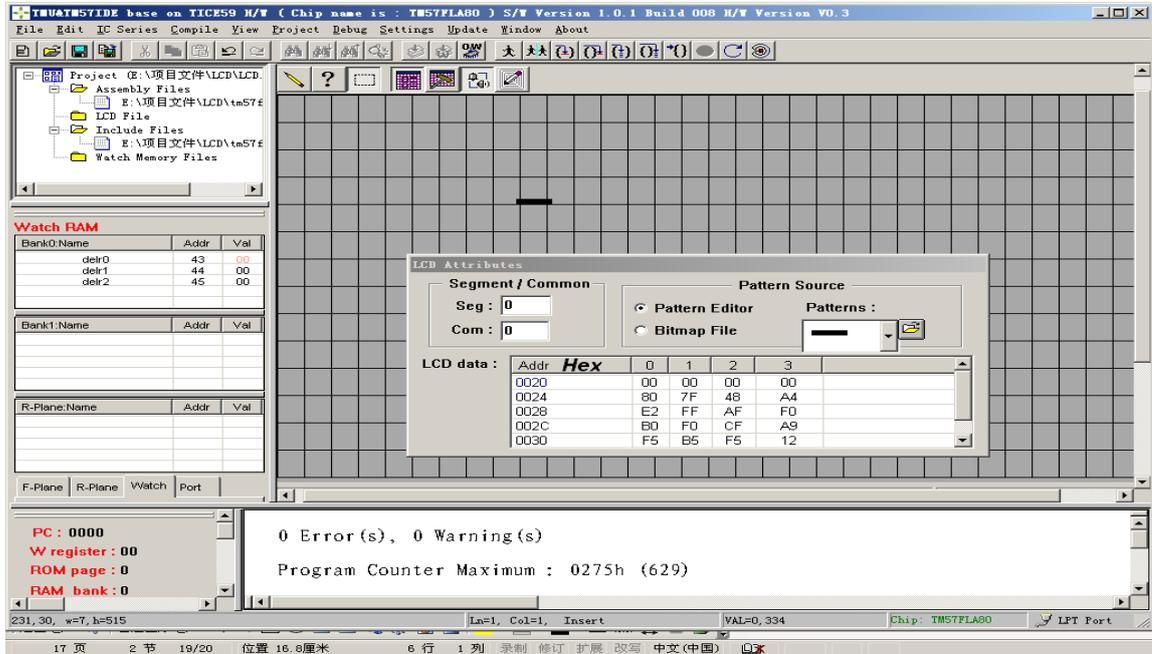
入编辑器，如果被载入的文件里有图形，那么在左边的

里面将会显示出来，可以通过单击按钮实现翻页。如下图所示：

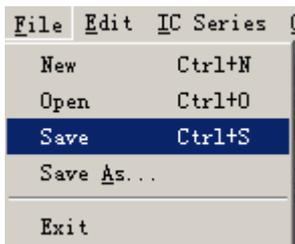


15)单击鼠标左键确定所选的图形。

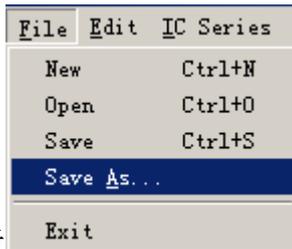
16)单击按钮，使这个图标处于选中状态。在栅格区适当位置单击一下鼠标左键。将选择的图形放入到栅格区。如下图：



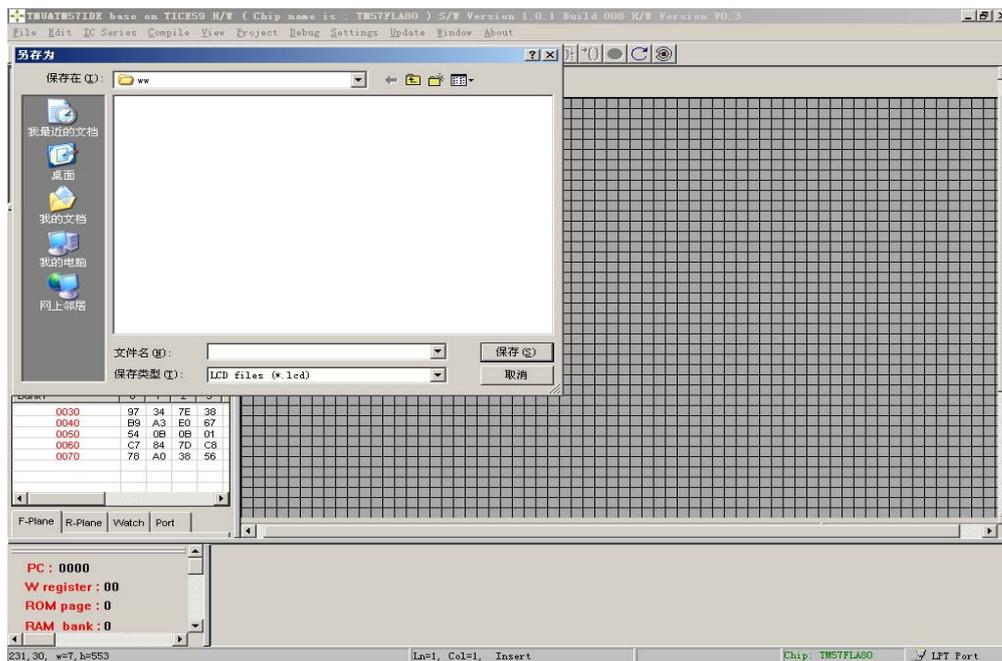
17)如果需要调整这个图形的位置，单击按钮，然后移动鼠标到需要调整的图形的上方，鼠标的图形会变成四个箭头朝外的图形，这表示该图形处于待拖动状态。按下鼠标左键不放开，拖动图形到需要的位置，然后放开鼠标。该图形的位置就调整好了。



18)单击，存储所建的 LCD 模拟面板图，如果是第一次存储，那么将会弹出如下图所示的对话框，

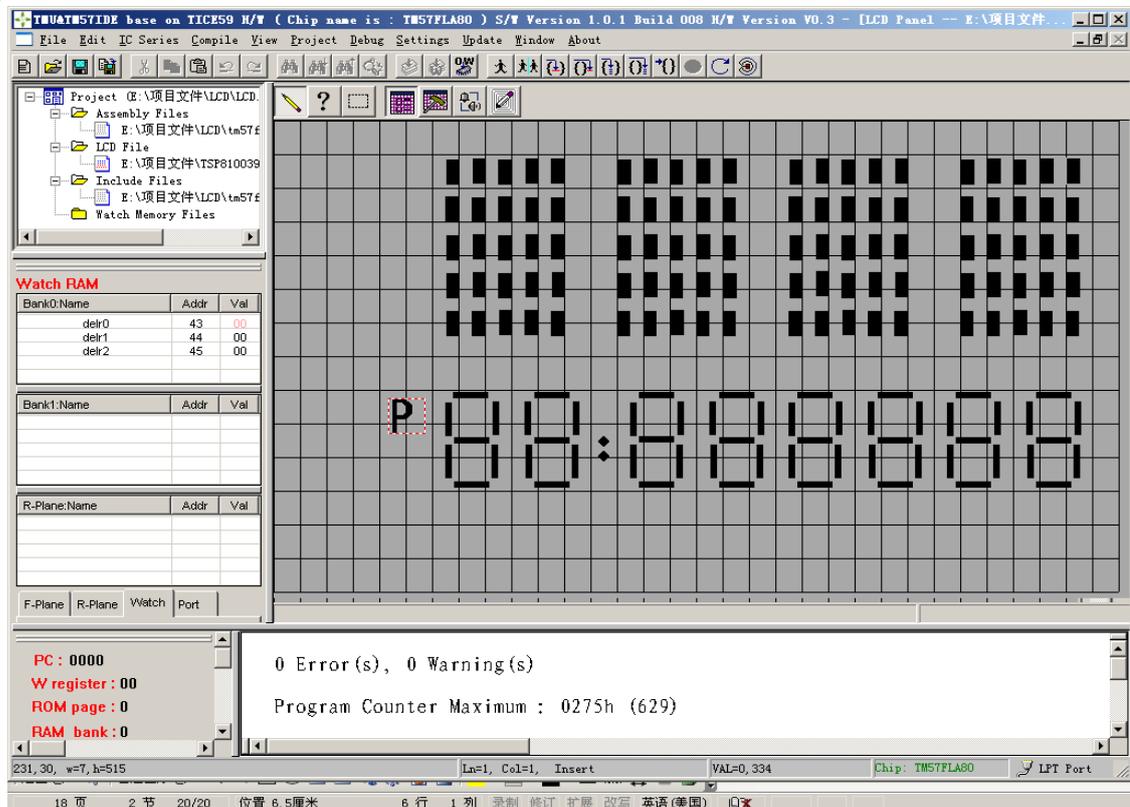


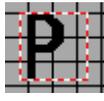
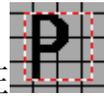
当然如果单击，那都会弹出如下的对话框。

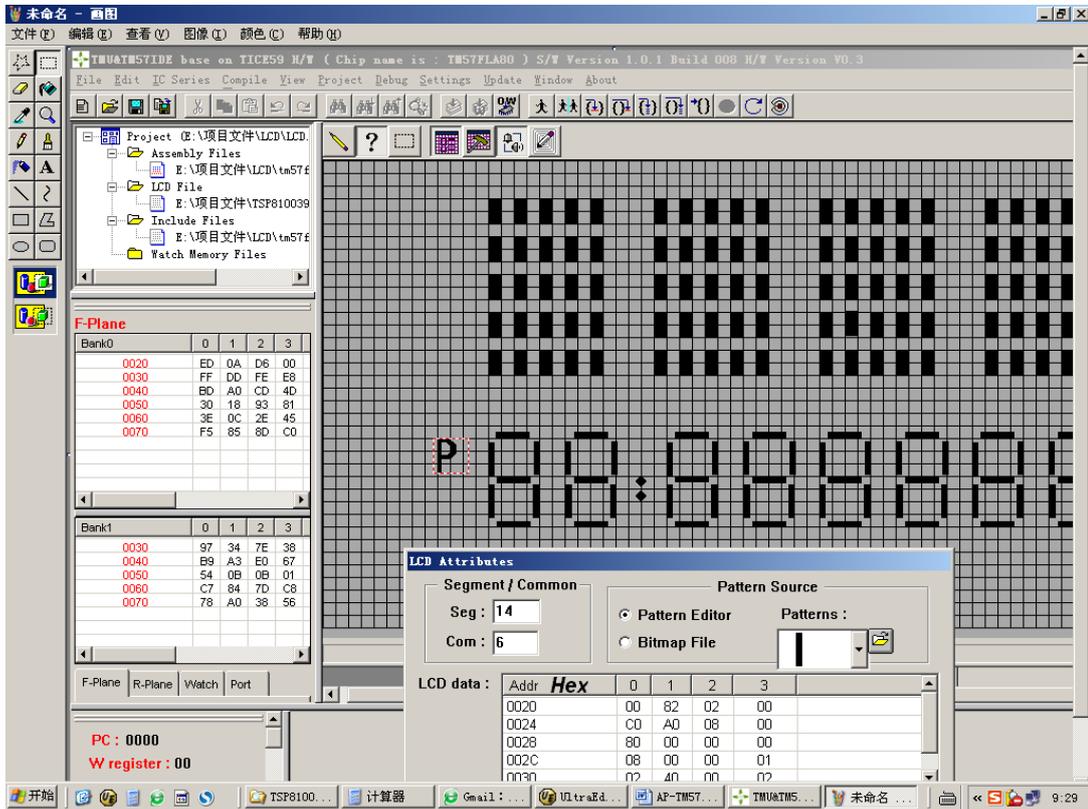


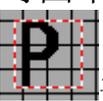
在对话框中可以选择 LCD 模拟面板图存储的路径及名称，然后点击保存按钮，以便保存所建的 LCD 模拟面板图。

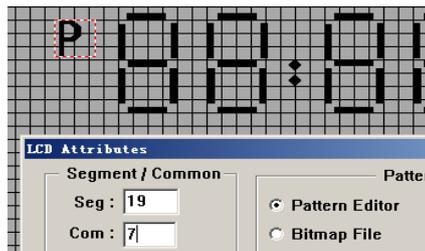
19) 重复 14~18 步，将栅格区布置成如下图所示的样式。

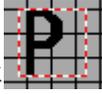


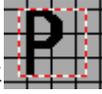
20)单击  按钮，使其处于选中状态。移动鼠标到“P”上方，单击鼠标左键，“P”字符会呈  状态。然后在  上方双击鼠标左键或直接单击  按钮，图面将呈现如下的状态：



此时图中选项框的 Seg 和 Com 右边的数据分别是  字符所对应的 SEG 及 COM 值，根据 LCD 面板图，我们将这两个数据修改为 19 和 7，修改后的图面如下：





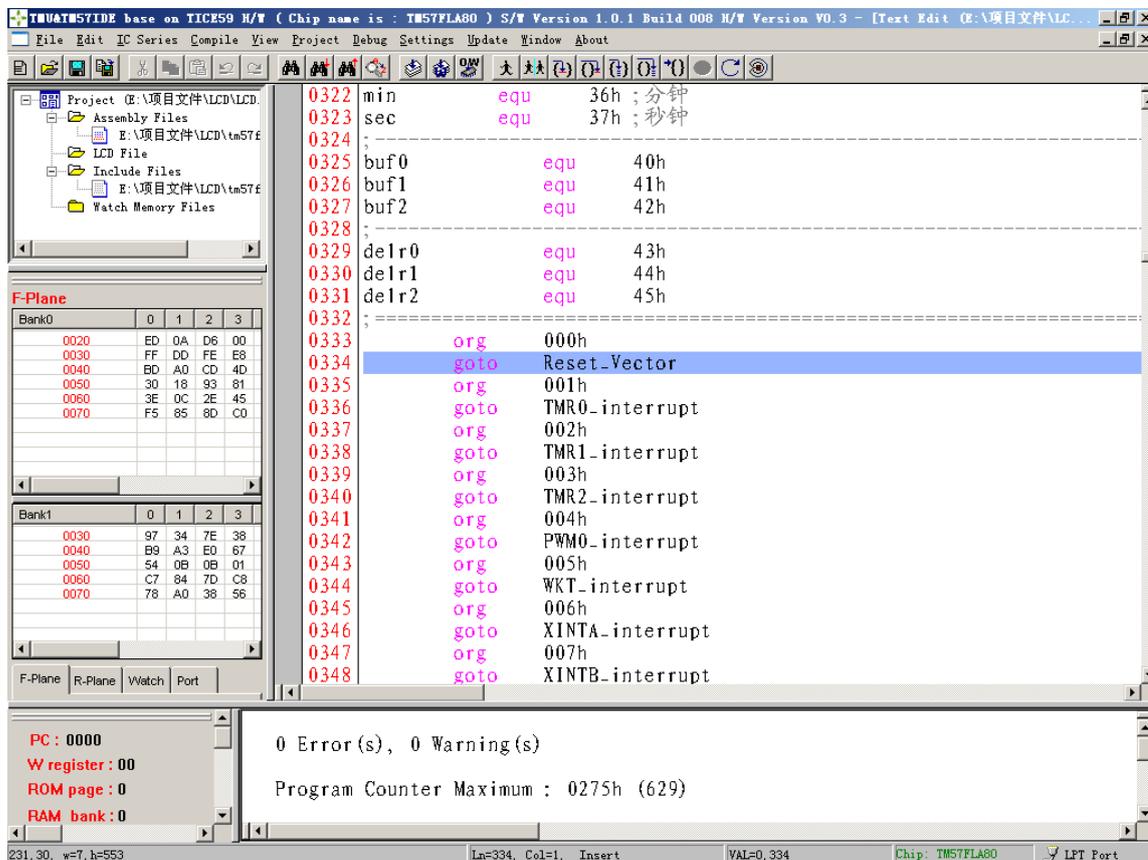
这样字符的 SEG,COM 属性就调整好了。

21)重复 20 步的动作，将栅格区其余 24 个黑色图形的各自对应的 Seg 和 Com 值设好。

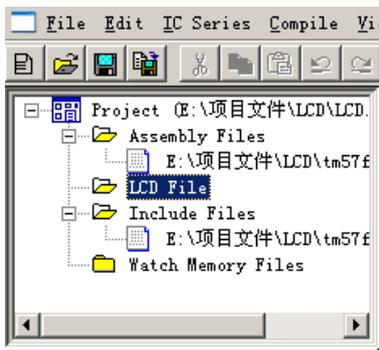
最后，单击按钮，退出属性编辑状态。这样一个完整的 LCD 模拟面板图就建好了。

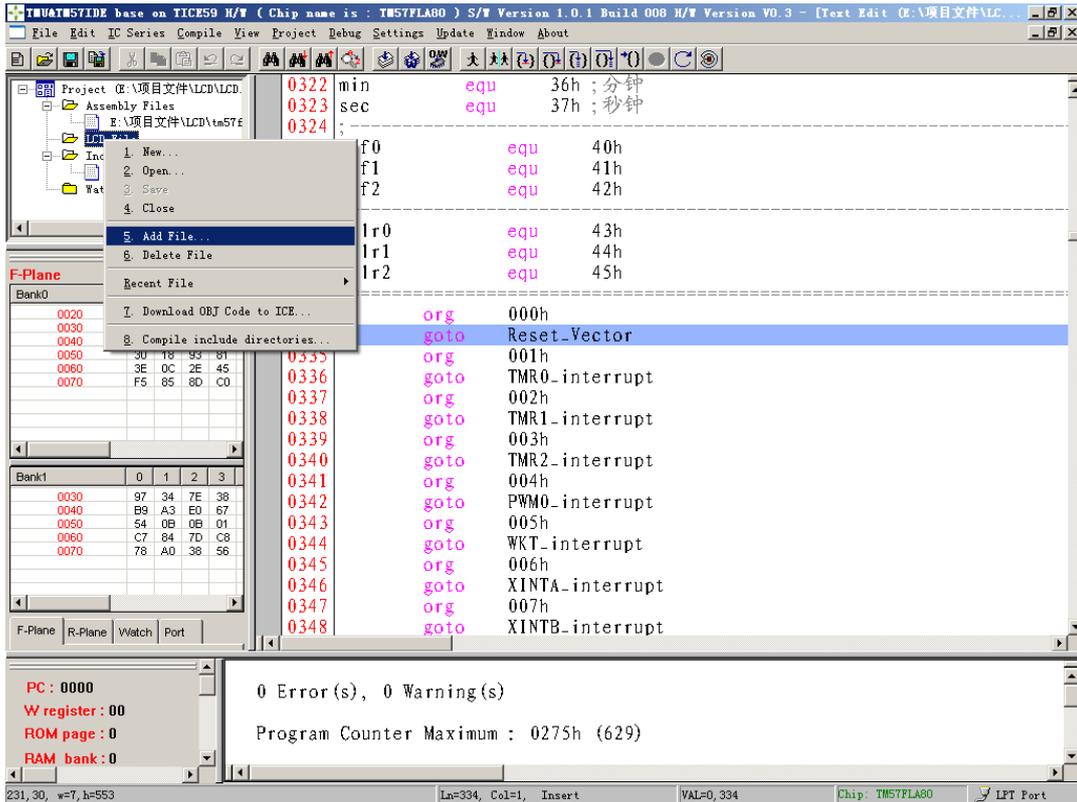
2. LCD 模拟面板图的使用方法

1) 打开 ICE59 工程环境，如下图所示：

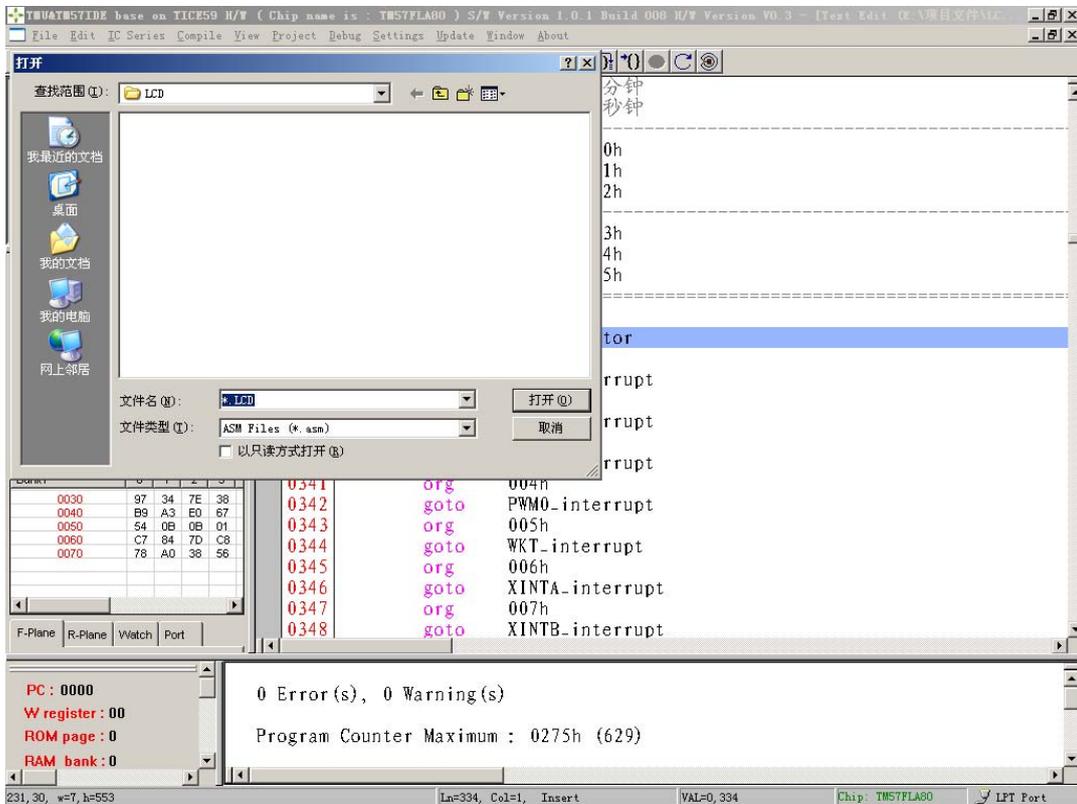


2) 鼠标单击 LCD File，然后单击右键，如下图：

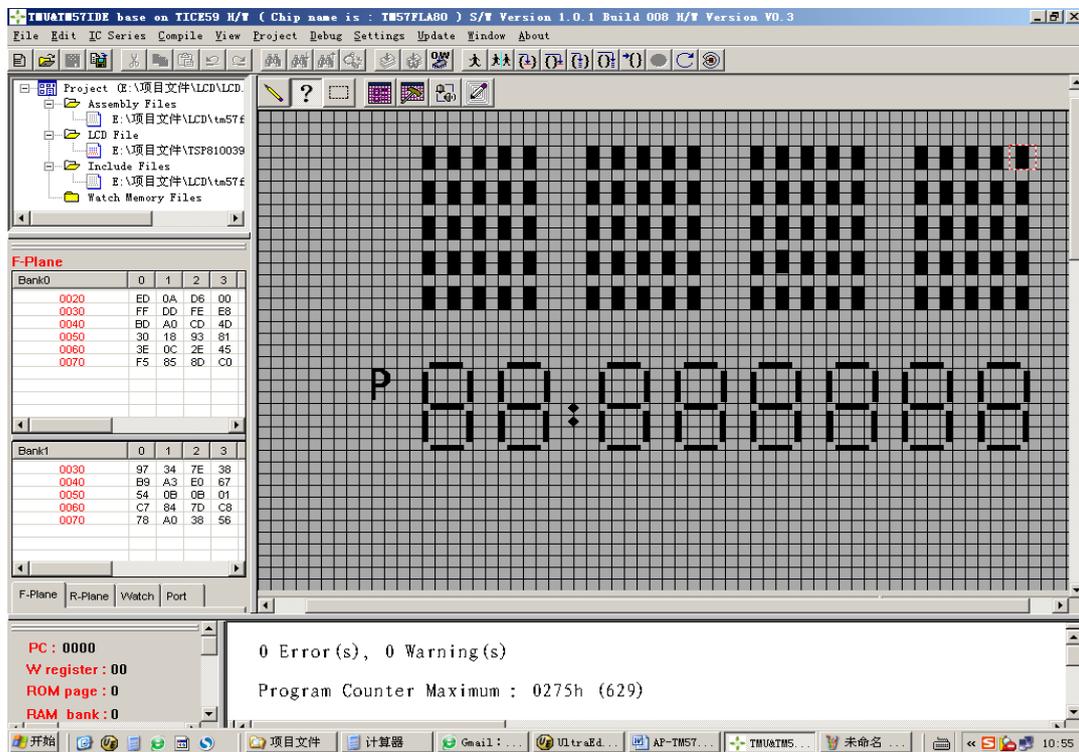




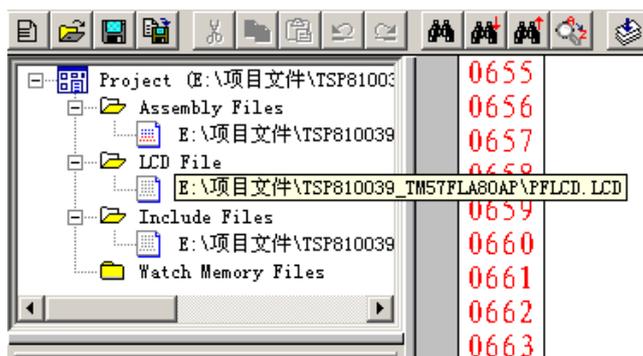
3) 移动鼠标到“5 Add File”按钮并单击，弹出下图对话框：



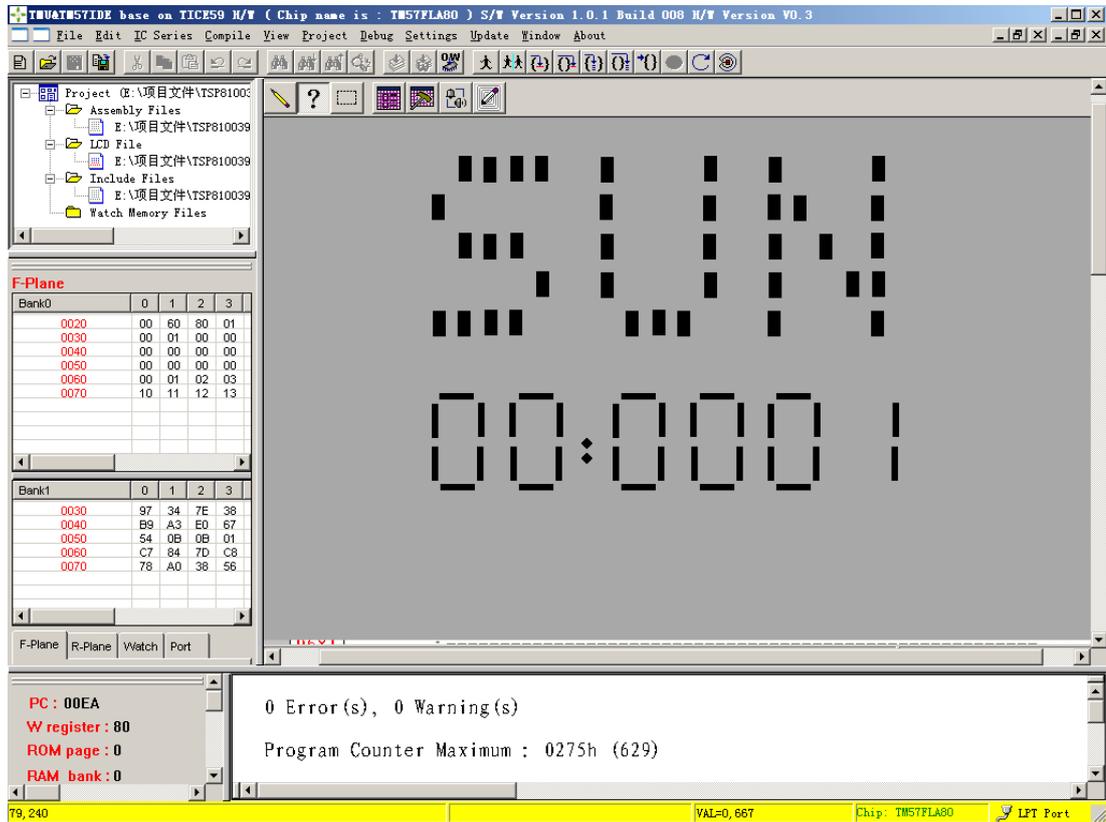
- 4) 将先前制作的 LCD 模拟面板图载入工程环境。并将其打开，显示如下：



- 5) 将工程软件经过编译下载后全速运行几秒钟停下来，双击。



打开 LCD 文件，LCD 模拟面板图显示如下：



注意，LCD 模拟面板图，只能在程序运行停止时更新显示，在程序运行过程中不会更新显示。