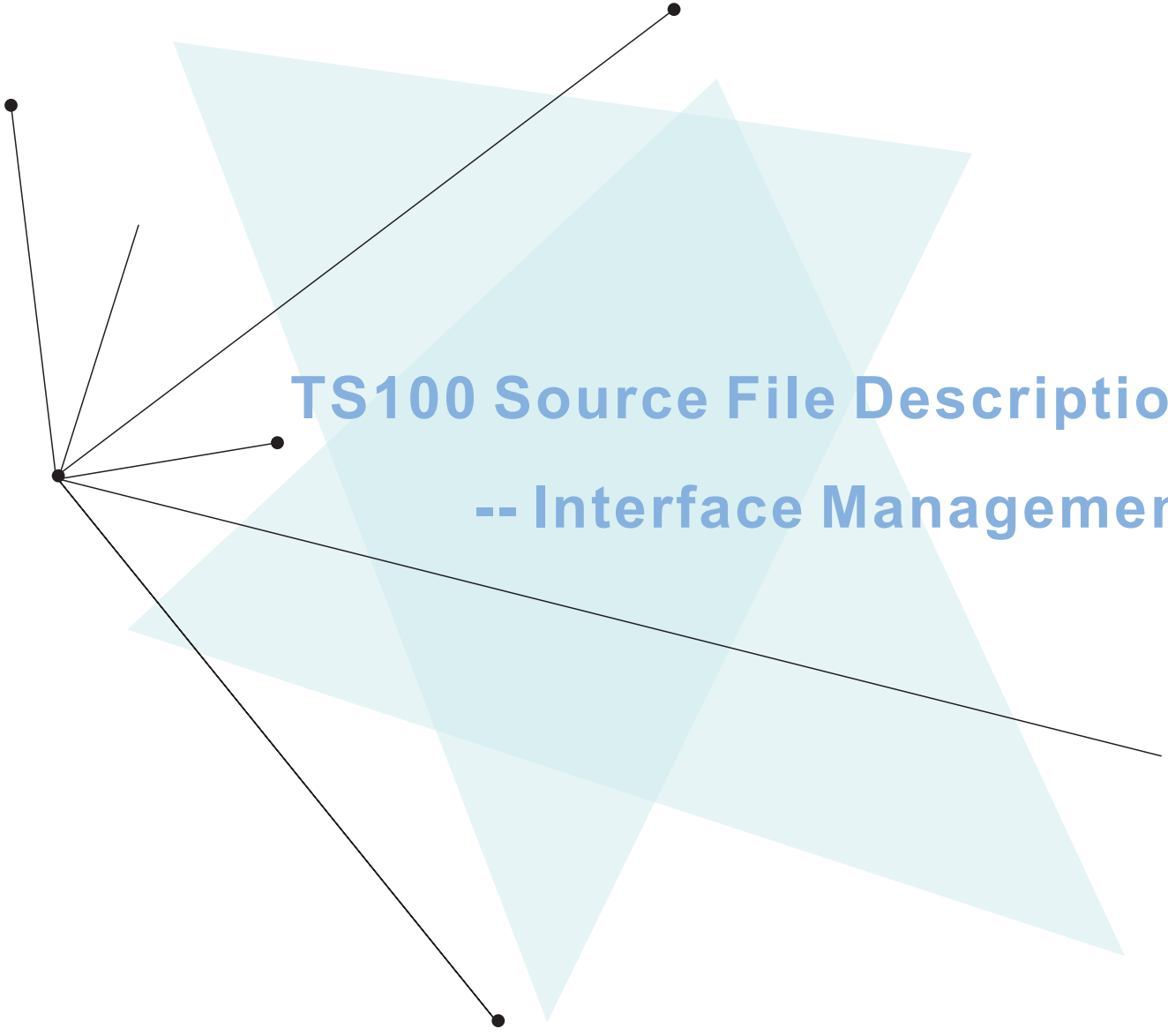


TS100 Source File Description

-- Interface Management

Content

1. Display Function P1

2. Character Library Making P3

3. Horizontal Scrolling Display P6

4. Vertical Scrolling Display P7

5. Functions of Each Interface P9

1. Display Function

The resolution of TS100's display is 16*96. To display something on the OLED screen, we just need to light up the corresponding bits following its rules.

```
/*******************************************************************************
函数名: Oled_DrawArea
函数作用:显示一个区域
输入参数: x0:起始横坐标
          y0:起始纵坐标(0,8,16,24)
          wide:显示内容宽度
          high:显示内容高度
          ptr:显示的内容库指针
返回参数:下一库指针
*******************************************************************************/
u8* Oled_DrawArea(u8 x0,u8 y0,u8 wide, u8 high,u8* ptr)
{
    u8 m,n,y;

    n = y0 + high;
    if(y0 % 8 == 0) m = y0 / 8;
    else            m = y0 / 8 + 1;

    if(n % 8 == 0)  y = n / 8;
    else            y = n / 8 + 1;

    for(; m < y; m++) {
        Set_ShowPos(x0,m);
        ptr = Data_Command(wide,ptr);
    }
    return ptr;
}

/*******************************************************************************
函数名: Clean_Char
函数作用:清除长度wide位置为k处的屏幕
输入参数:    清除位置wide为清除宽度
返回参数:NULL
*******************************************************************************/
void Clean_Char(int k,u8 wide)
{
```

Oled.c, Line 145

Input parameters: x0: the starting abscissa (0-96).
y0: the starting ordinate (0,8,16,24).
wide: the width of the displayed content.
high: the height of the displayed content.
ptr: the displayed content library (font library) pointer;
Return parameter: the next character library pointer.

For example, if we want to display CONFIG, we need to find the const u8 Config[] array in WordLib.h, and then assign config to the temporary pointer ptr and display characters one by one.

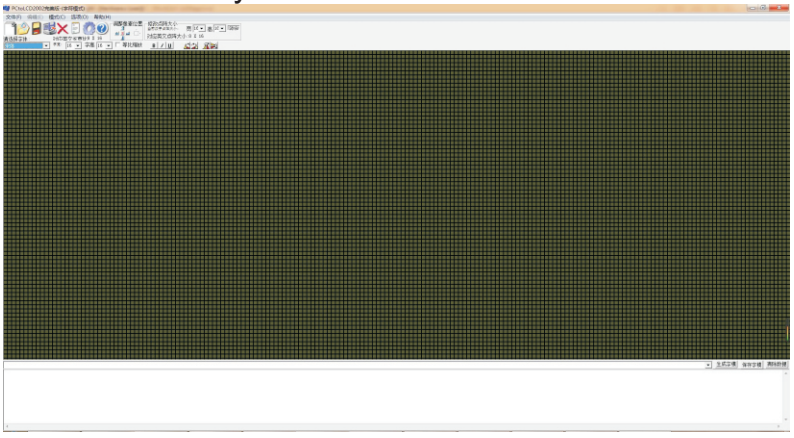
```
262 函数作用:显示CONFIG
263 输入参数:NULL
264 返回参数:NULL
265 ********************************************************************************/
266 void Show_Config(void)
267 {
268     u8* ptr;
269     u8 j;
270
271     ptr = (u8*)Config;
272     for(j = 0; j < 6; j++) {
273         ptr = Oled_DrawArea(j*16,0,16,16,ptr);
274     }
275 }
276
277 /********************************************************************************
278 函数名: Show_TempDown
279 函数作用:显示温度下降: 实际温度>>>目标温度
280 输入参数:temp 实际温度,dst_temp 目标温度
281 返回参数:NULL
282 ********************************************************************************/
283 void Show_TempDown(s16 temp,s16 dst_temp)
284 {
285     static u8 guide_ui = 0;
286     char str[8];
287
288     memset(str,0x0,6);
289     sprintf(str,"%d",temp);
290     if(gTemperatureshowflag == 0) {
291         if(temp > 99) str[3] = 'C';
292         else {
293             str[2] = 'C';
294             str[3] = ' ';
295         }
296     } else {
297         if(temp > 99) str[3] = 'E';
298         else {
299             str[2] = 'E';
300             str[3] = ' ';
301         }
302     }
303
304     Display_Str10(1,str);
```

UI.c, Lines 271 to 274

ptr = Oled_DrawArea(j*16,0,16,16,ptr) means that: The first 16*16 character of the current character library ptr will be displayed. The displayed location is (j*16, 0). CONFIG is displayed at 0, 16, 32, 48, 64, and 80. Oled_DrawArea() returns to the first address in the character library of the next character.

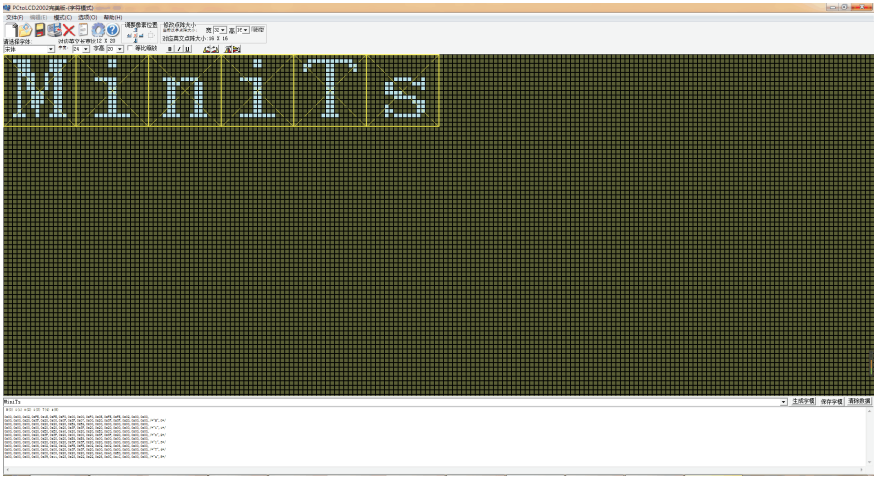
2. Character Library Making

Subtitle is the text displayed on the screen, it is not only an important part of the screen also a necessity for supplement and decoration. Check below for details about how to create the Character Library.



Open Pctolcd2002 and click Settings. And then set the picture according to the following figure (common cathode coding, column and line mode):

Next, we can begin to make our font library.
We use the string MiniTs as an example. There are six letters. The width and height of each of them are 16. Enter MiniTs in the input box. The string is now displayed as 8*16 format. Click the Settings column to modify the size of the dot matrix and adjust the position and size of the characters:



After completing the above operations, click Generate Character Matrix and the following data will be obtained:

0x00,0x00,0x02,0xFE,0x1E,0xFE,0xF0,0x00,0x00,0xF0,0x0E,0xFE,0xFE,0x02,0x00,0x00,
0x00,0x00,0x20,0x3F,0x20,0x00,0x0F,0x3F,0x07,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,/*"M",0*/
0x00,0x00,0x00,0x00,0x20,0x20,0x20,0xE6,0xE6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x00,0x00,0x00,0x00,/*"i",1*/
0x00,0x00,0x00,0x20,0xE0,0xE0,0x40,0x20,0x20,0x20,0xE0,0xC0,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,0x00,/*"n",2*/
0x00,0x00,0x00,0x00,0x20,0x20,0x20,0xE6,0xE6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x00,0x00,0x00,0x00,/*"i",3*/
0x00,0x00,0x0C,0x06,0x02,0x02,0x02,0xFE,0xFE,0x02,0x02,0x02,0x06,0x0C,0x00,0x00,
0x00,0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,0x00,/*"T",4*/
0x00,0x00,0x00,0x00,0x00,0xC0,0xC0,0x20,0x20,0x20,0x20,0x40,0x40,0xE0,0x00,0x00,0x00,
0x00,0x00,0x00,0x00,0x39,0x11,0x23,0x23,0x22,0x22,0x26,0x3C,0x1C,0x00,0x00,0x00,/*"s",5*/

```
const u8 MiniTs[] = { /*16*16*/
  0x00,0x00,0x02,0xFE,0x1E,0xFE,0xF0,0x00,0x00,0xF0,0x0E,0xFE,0xFE,0x02,0x00,0x00,
  0x00,0x00,0x20,0x3F,0x20,0x00,0x0F,0x3F,0x07,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,/*"M",0*/
  0x00,0x00,0x00,0x00,0x20,0x20,0x20,0xE6,0xE6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
  0x00,0x00,0x00,0x00,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x00,0x00,0x00,0x00,/*"i",1*/
  0x00,0x00,0x00,0x20,0xE0,0xE0,0x40,0x20,0x20,0x20,0xE0,0xC0,0x00,0x00,0x00,0x00,
  0x00,0x00,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,0x00,/*"n",2*/
  0x00,0x00,0x00,0x00,0x20,0x20,0x20,0xE6,0xE6,0x00,0x00,0x00,0x00,0x00,0x00,0x00,
  0x00,0x00,0x00,0x00,0x20,0x20,0x20,0x3F,0x3F,0x20,0x20,0x20,0x00,0x00,0x00,0x00,/*"i",3*/
  0x00,0x00,0x0C,0x06,0x02,0x02,0x02,0xFE,0xFE,0x02,0x02,0x02,0x06,0x0C,0x00,0x00,
  0x00,0x00,0x00,0x00,0x00,0x00,0x20,0x3F,0x3F,0x20,0x00,0x00,0x00,0x00,0x00,0x00,/*"T",4*/
  0x00,0x00,0x00,0x00,0xC0,0xC0,0x20,0x20,0x20,0x20,0x40,0x40,0xE0,0x00,0x00,0x00,
  0x00,0x00,0x00,0x00,0x39,0x11,0x23,0x23,0x22,0x22,0x26,0x3C,0x1C,0x00,0x00,0x00,/*"s",5*/
};
```

Put the data into the array, and then we can display the contents. Similarly, you can modify the size of characters. The character size on the display side should be changed to the corresponding size.

The display effect is as follows

The corresponding code is:

```
ptr = (u8*)MiniTs;
for(j = 0; j < 6; j++) {
   ptr = Oled_DrawArea(j*16,0,16,16,ptr);
}
```

3. Horizontal Scrolling Display

The following situations need horizontal scrolling display: the system prompt button, the setting of initial states, etc.In fact, we can implement scrolling display by constantly changing the location of each character. Next, let's analyze the scrolling display code of the prompt button press:

```
202         ptr0 = Oled_DrawArea(0,0,96,16,(u8*)Maplib);
203     } else if(i == 3) {//第4 步
204         for(j = 0 ; j < 6; j++) {
205             k = 84;
206             while(k >= posi) {
207                 ptr0 = (u8*)Maplib1 + j*28;
208                 Clean_Char(k+7,14);
209                 ptr0 = Oled_DrawArea(k,0,14,16,ptr0);
210                 k-=7;
211                 Delay_Ms(10);
212             }
213             posi += 14;
214         }
215         posi = 0;
216     }
217     i++;
218     if(i == 4)  i = 0;
219 }
220 /*****************************************************************************
221 函数名: Show_Ver
222 函数作用:显示版本
223 输入参数:ver 版本号flag (0 :滚动显示 )(1不滚动)
224 返回参数:NULL
225 *****************************************************************************/
226 void Show_Ver(u8 ver[],u8 flag)
227 {
228     u8 *ptr;
229     int k,i;
230     u8 temp0,temp1,temp2;
231
232     if(ver[2] >= 0x30 && ver[2] < 0x3a)   temp1 = ver[2] - 0x30;
233     if(ver[3] >= 0x30 && ver[3] < 0x3a)   temp2 = ver[3] - 0x30;
234     if(ver[0] >= 0x30 && ver[0] < 0x3a)   temp0 = ver[0] - 0x30;
235
```

UI.c, Line 204

This function traverses six characters, calculates the desired display locations when each character approaches to designated locations with a step of 7 pixels, and clears the number on the previous position.

When traverse and stepping combines with each other, a continuous horizontal scrolling display is formed for human eyes. Similarly, for the horizontal display of other characters, we just need to replace the character library in ptr0 = (u8*) Maplib1 + j*28.

4. Vertical Scrolling Display

Vertical scrolling display includes: version information of TS100 during the boot process, temperature scrolling at temperature setting, and the temperature indicating icon during temperature control. Vertical scrolling is slightly more complicated than the horizontal scrolling.

UI.c, Line 558

```
550  }
551  /*********************************************************************
552  函数名: Show_ReverseChar
553  函数作用:竖向动态显示字符
554  输入参数: ptr:字节库    num:个数
555                width:宽度    direction :方向 (0 up, 1 down)
556  返回参数:NULL
557  *********************************************************************/
558  void Show_ReverseChar(u8* ptr,u8 num,u8 width,u8 direction)
559  {
560      static u32 j = 0,m = 0,po_j[3] = {0,0,0},po_m[3] = {0,0,16};
561      u32 i,k;
562
563      if(direction == 0) { //up
564          if(gUp_flag == 0) { //前一状态不是加热
565              j = 0;
566              m = 0;
567              gUp_flag     = 1;
568              gDown_flag   = 0;
569              gLevel_flag  = 0;
570          } else {
571              j = po_j[0];
```

Input parameters:
ptr: byte library.
num: the number of characters.
width: the width of a character.
direction: the scrolling direction (0 indicates up. 1 indicates down. You can also define other directions, you just need to change the initial values of po_j and po_m).
For example:
Show_ReverseChar((u8*)Ver_s,8,12,2); indicates that 8 characters (num = 8) with a width of 12 (width = 12) are scrolling up (direction = 2), and the character library is (ptr = Ver_s)Ver_s.

A simple example shows the normal character display order on the OLED is reversed in storing order. As shown in Table 1 and Table 2, 0 means the digital LED is off. 1 means it is on.

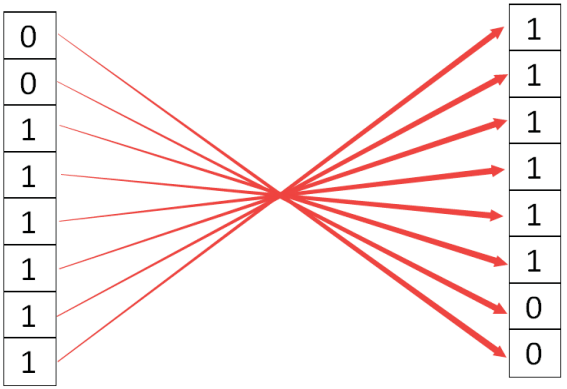


Table 1: characters display order on the OLED

Table 2: storing order in the character library

4. Vertical Scrolling Display

Follow the below steps to adjust a normal display.

UI.c, Line 589

First step: However, the configuration in the character library is different and shown in Table 2. Therefore, first of all, Show_ReverseChar() converts the upper 8 bits and lower 8 bits in the character library (Table 2) to the normal

```
584         }
585     } else {
586         j = po_j[2];
587         m = po_m[2];
588     }
589     for(i = 0; i < width * 2 * num; i++)    gTemp_array[i] = Reverse_Bin8(*(ptr + i)); //逆向8位
590
591     for(k = 0; k < width * 2 * num; k += width * 2)
592         for(i = 0; i < width ; i++) {
593             gTemp_array_u16[i + k] = ((gTemp_array[i + k] & 0x00FF) << 8) | gTemp_array[i + k + width] ;//上半部下半部与成u16 便于移位
594             if(direction == 1) {
595                 if(j == 0)  gTemp_array_u16[i + k] <<= m;//下面空，上面显示
596                 else        gTemp_array_u16[i + k] >>= j;//上面空，下面显示
597             } else { //上
598                 if(m == 0)  gTemp_array_u16[i + k] <<= j;//下面空，上面显示
599                 else        gTemp_array_u16[i + k] >>= m;//上面空，下面显示
600             }
```

Second step: Next, combine the upper 8 bits with the normal lower 8 bits to form 16-bit data

UI.c, Line 593

UI.c, Lines 594 to 603

Third step: According to the scrolling direction, shift the 16-bit

UI.c, Line 601

Fourth step: Divide the 16-bit data after shift into two 8-bit parts

```
591     for(k = 0; k < width * 2 * num; k += width * 2)
592         for(i = 0; i < width ; i++) {
593             gTemp_array_u16[i + k] = ((gTemp_array[i + k] & 0x00FF) << 8) | gTemp_array[i + k + width] ;//上半部下半部与成u16 便于移位
594             if(direction == 1) {
595                 if(j == 0)  gTemp_array_u16[i + k] <<= m;//下面空，上面显示
596                 else        gTemp_array_u16[i + k] >>= j;//上面空，下面显示
597             } else { //上
598                 if(m == 0)  gTemp_array_u16[i + k] <<= j;//下面空，上面显示
599                 else        gTemp_array_u16[i + k] >>= m;//上面空，下面显示
600             }
601             gTemp_array[i + k] = (gTemp_array_u16[i + k] & 0xFF00) >> 8;
602             gTemp_array[i + k + width] = gTemp_array_u16[i + k] & 0x00FF;
603         }
604
605     for(i = 0; i < width * 2 * num; i++)    gTemp_array[i] = Reverse_Bin8(gTemp_array[i]); //移位后再逆向
606
607     if(m == 0 && j == 16) { //全显示，换显示'头数'
608         j = 0;
609         m = 16;
610     }
611     if(m == 0)  j++;
612     else        m--;
613
614     if(direction == 0) { //up
615         po_j[0] = j;
616         po_m[0] = m;
617     } else if(direction == 1) {
618         po_j[1] = j;
619         po_m[1] = m;
620     } else {
621         po_j[2] = j;
622         po_m[2] = m;
623     }
624 }
625
626 /*******************************************************************************
627  Show_TempReverse 竖向动态显示温度字符
628  word_num:     个数
629  word_width:   宽度
630  direction :   方向 (0 up, 1 down)
631  *******************************************************************************/
```

Fifth step: At last, carry out a reverse operation: Convert the normal display data in Table 1 to the OLED display

UI.c, Line 605

Finally, the global parameter gTemp_array[] after shift is obtained. When the shift is accumulated step by step, the content in the gTemp_array[] library will be displayed, and a vertical screen scrolling will be implemented.

The principle of other vertical scrolling is the same. The difference is that the size, length, as well as the initial location of rolling characters may be different, then the scrolling effects will be different.

Based on the above character library making, now we have three different display modes. From now on, you can rewrite TS100's interfaces and make your own interfaces. Let me introduce calling positions of some interfaces to you so that you can quickly locate the interfaces you need to modify.

```
936     case IDLE:
937         if(gCont == 1) {
938             gCont = 0;
939             Clear_Screen();
940         }
941
942         if(ver_flag == 0) {
943             Display_BG();
944             Show_Ver(device_info.ver,0);
945             ver_flag = 1;
946         } else if(UI_TIMER == 0 && G6_TIMER != 0) {
947             Show_Notice();
948             UI_TIMER = 50;
949         }
950         if(G6_TIMER == 0) {//屏保
951             id_cnt++;
952             if(id_cnt == 50)Sc_Pt(bk--);
953             if(bk == 0) Oled_DisplayOff();
954         }
955         if((Get_MmaShift() == 1) || (Get_gKey() != NO_KEY)) {
956             G6_TIMER = device_info.idle_time;
957             bk = 0x33;
958             Sc_Pt(bk);
959             Oled_DisplayOn();
960         }
961         break;
962     case TEMP_CTR:
963         if(gCont == 0) {
964             gCont = 1;
965             Set_LongKeyFlag(1);
966             Clear_Screen();
967         }
968
969         ht_flag = Get_HtFlag();
970
971         if(ht_flag != 1) {
972             if(td_flag == 1)  Clear_Screen();
973             td_cnt  = 0;
974             td_flag = 0;
975         }
976
977         if(td_cnt == 75 && td_flag == 0) {
978             Clear_Screen();
979             td_flag = 1;
```

UI.c, Line 943

The function Display_BG() used to display the logo in the booting process is in UI.c, line 943. The logo display is normal. To change the displayed logo, you can change the name library (ptr = (u8*)Mini).

The iron will display its firmware version information after the logo. The version information can be displayed either vertically or normally. This display function Show_Ver(device_info.ver,0) is located in line 944, UI.c. device_info.ver is the version number array. The latter parameter indicates the display mode.

UI.c, Line 944

Next, the iron displays the key-pressing prompt information and 'press'. The relate function Show_Notice() is in line 947, UI.c. Show_Notice is divided into 4 parts. All of them are very clear, and you can change them manually.

UI.c, Line 947

```
994             if((temp_val > device_info.t_work) && (temp_val - device_info.t_work < 18))        te
995             else if((temp_val <= device_info.t_work) && (device_info.t_work - temp_val < 18)) te
996             if(Get_TemperatureShowFlag() == 1) {
997                 temp_val = TemperatureShow_Change(0,temp_val);
998             }
999
1000            Display_Temp(1,temp_val/10);
1001            Show_HeatingIcon(ht_flag,Get_MmaActive());//0升温1降温2恒温
1002            td_cnt++;
1003        }
1004        break;
1005    case TEMP_SET:
1006        Temp_SetProc();                                    //设置
1007        break;
1008    case CONFIG:
1009        if(gCont == 1) {
1010            gCont = 0;
1011            Clear_Screen();
1012        }
1013        switch(Get_gKey()) {
1014        case KEY_CN|KEY_V1:
1015            config_show = 1;
1016            break;
1017        case KEY_CN|KEY_V2:
1018            config_show = 2;
1019            break;
1020        case KEY_CN|KEY_V3:
1021            config_show = 0;
1022            break;
1023        default:
1024            break;
1025        }
1026        if(config_show == 0) {
1027            Show_Config();
1028        } else if(config_show == 1) {
1029            Display_BG();
1030        } else if(config_show == 2) {
1031            Show_Ver(device_info.ver,1);
1032        }
1033        if(config_show != 3) {
1034            Set_gKey(NO_KEY);
1035            config_show = 3;
1036        }
1037        break;
1038    case THERMOMETER:
1039        if(gCont == 0) {
1040            gCont = 1;
1041            Clear_Screen();
1042        }
1043        if(gCalib_flag != 0) {
```

```
1043    if(gCalib_flag != 0) {
1044        //显示校准完成
1045        Show_Cal(gCalib_flag);
1046        gCalib_flag = 0;
1047        Clear_Screen();
1048    }
1049    if(UI_TIMER == 0) {
1050        temp_val = Get_Temp(0);
1051        if(Get_TemperatureShowFlag() == 1)
1052            temp_val = TemperatureShow_Change(0,temp_v
1053        }
1054        Display_Temp(1,temp_val/10);
1055        Display_Str(6,"F");
1056        UI_TIMER = 20;
```

UI.c ,Lines 1000 to 1001

In constant temperature status, the temperature display and heating up status display are implemented in lines 1000 and 1001 UI.c.temp_val is 10 times of the actual temperature. ht_flag indicates the heating status. Get_MmaActive() indicates the iron's movement.

Similarly, the function displays this temperature in Thermometer Mode and Sleep Mode. In Thermometer Mode, the temperature display is in line 1054, UI.c. The Show_Cal() in line 1045 is a calibrated-result displaying function. It displays the calibrated results according to the current display flags.

In configuration mode, the system displays CONFIG (UI.c, line 1027, Show_Config()), LOGO (UI.c, line 1029,Display_BG()), and the version number (UI.c, line 1031,Show_Ver()). The display method is the same as the above method. The difference is that the called locations or parameters are different.

```
1056            UI_TIMER = 20;
1057        }
1058        break;
1059    case ALARM:
1060        if(gCont == 0) {
1061            gCont = 1;
1062            Clear_Screen();
1063        }
1064        if(gCont == 1 && UI_TIMER == 0) {
1065            Show_Warning();
1066            UI_TIMER = 50;
1067        }
1068        break;
1069    case WAIT:
1070        temp_val = Get_TempVal();
1071        if((temp_val > device_info.t_standby) && (temp_val - device_info.t_standby <= 18))
1072        else if((temp_val <= device_info.t_standby) && (device_info.t_standby - temp_val <
1073
1074        ht_flag = Get_HtFlag();
```

UI.c ,Line 1065

The Show_Warning () function (UI.c, line 1065) is a warning interface calling function. It calls different different character libraries according to different warning types to display different warnings.

```
976
977        if(td_cnt == 75 && td_flag == 0) {
978            Clear_Screen();
979            td_flag = 1;
980        }
981
982        if(td_flag && UI_TIMER == 0) {
983            temp_val = Get_TempVal();
984            dst_temp = device_info.t_work;
985            if(Get_TemperatureShowFlag() == 1) {
986                temp_val = TemperatureShow_Change(0,temp_val);
987                dst_temp = TemperatureShow_Change(0,dst_temp);
988            }
989            Show_TempDown(temp_val,dst_temp);
990            UI_TIMER = 50;
991        }
992        if((TEMPSHOW_TIMER == 0) && (!td_flag)) {
993            temp_val = Get_TempVal();
994            if((temp_val > device_info.t_work) && (temp_val - device_info.t_work < 18))
995            else if((temp_val <= device_info.t_work) && (device_info.t_work - temp_val < 1
996            if(Get_TemperatureShowFlag() == 1) {
997                temp_val = TemperatureShow_Change(0,temp_val);
998            }
```

UI.c ,Line 989

The Show_TempDown(temp_val,dst_temp) function (UI.c, line 989) displays a dynamic picture: "current temperature >>> target temperature". This function is called when the temperature drop is relatively large for a long time.