



2556/2796 添加中文菜单 OSD

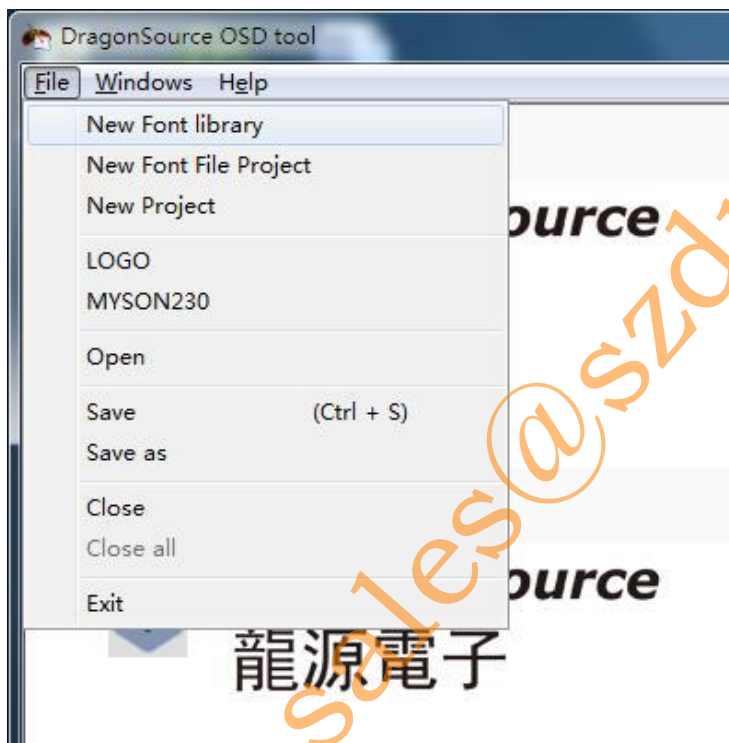
—2556/2796 为例

一. 使用软件工具 OSDEdit.exe 制作字库

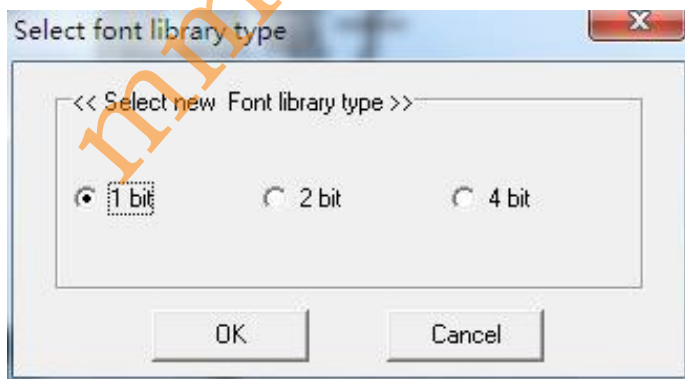
1) 打开软件 OsdEditor.exe



2) file->New Font library



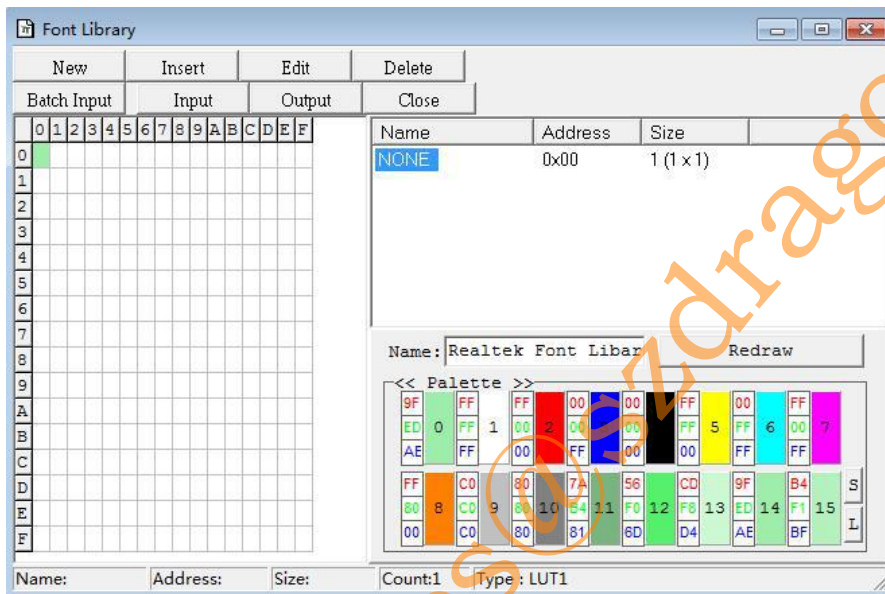
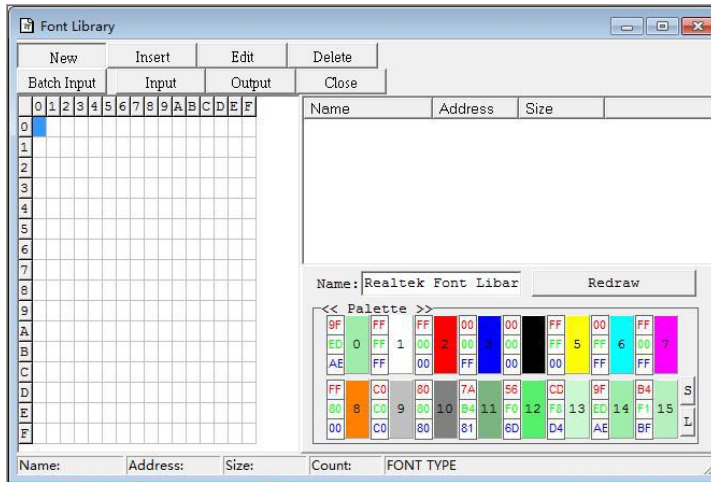
3) 选择 1bit



4) 选择 New, 一般要空一格, 即 (0,0) 不写字, draw 一个空白字, 宽 1 高 1 即可实现; 从 (1,0)



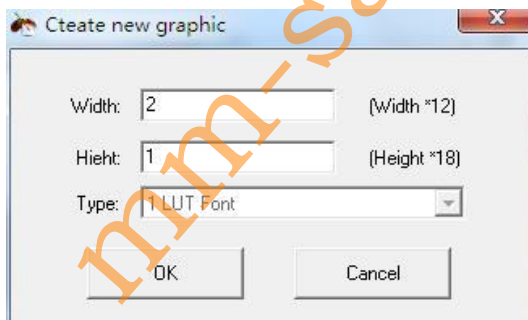
开始写字。



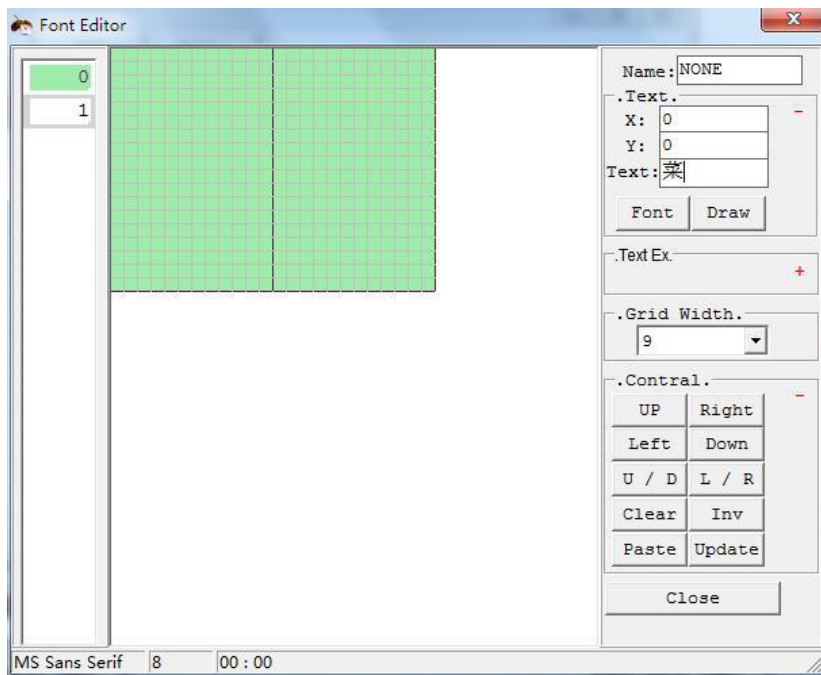
5) 空了一个小白格后其他取模设置:

width: 2

hieght: 1



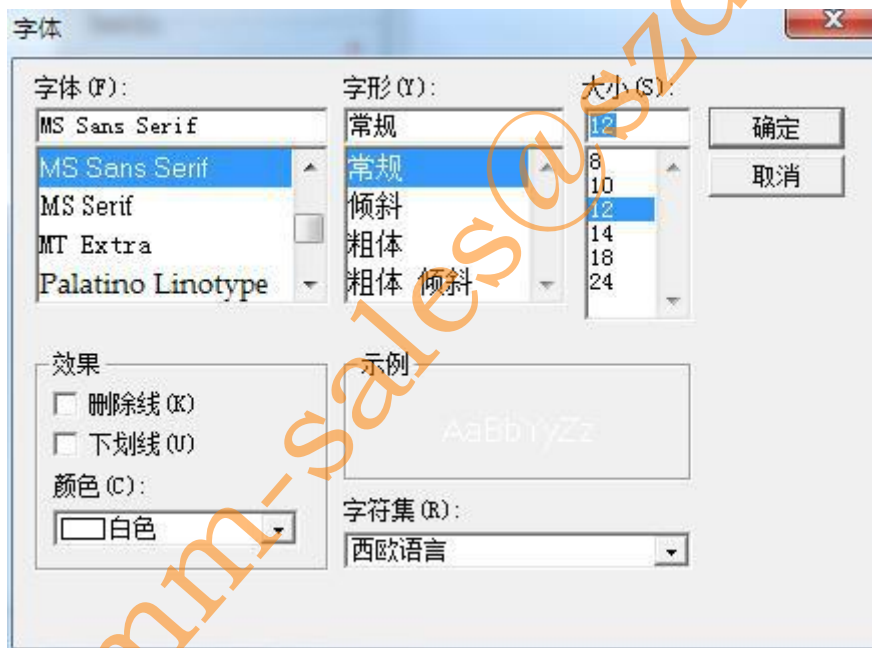
6) 字体编辑 Font Editor



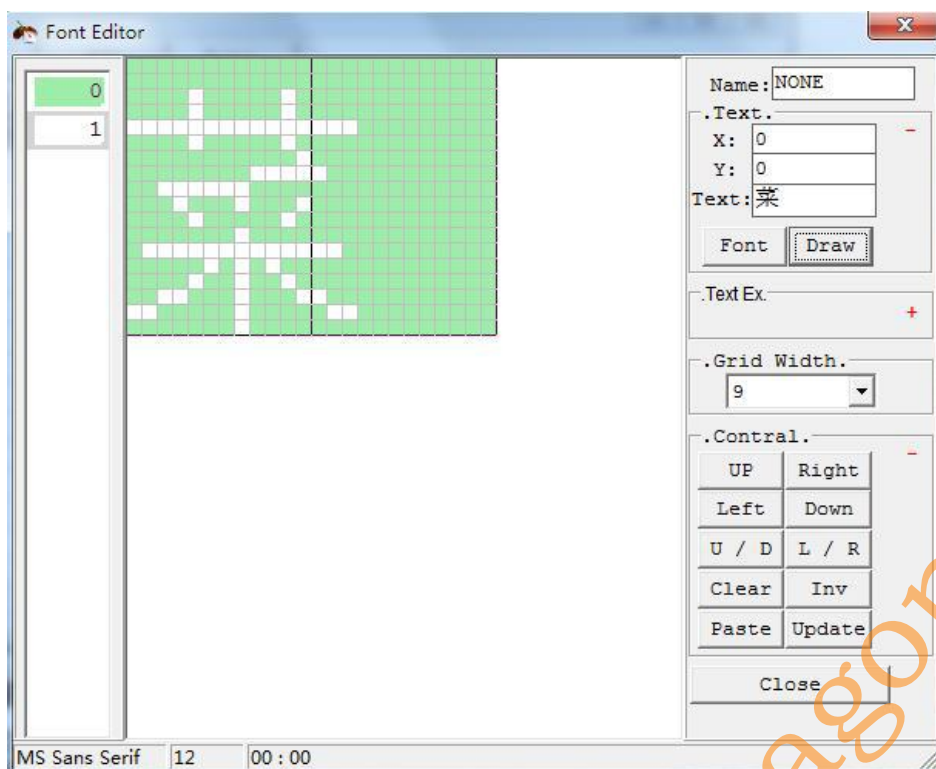
在 Text 处输入要增加的中文文字，如输入“菜”字；

7) 点击 Font 设置中文的字体大小：

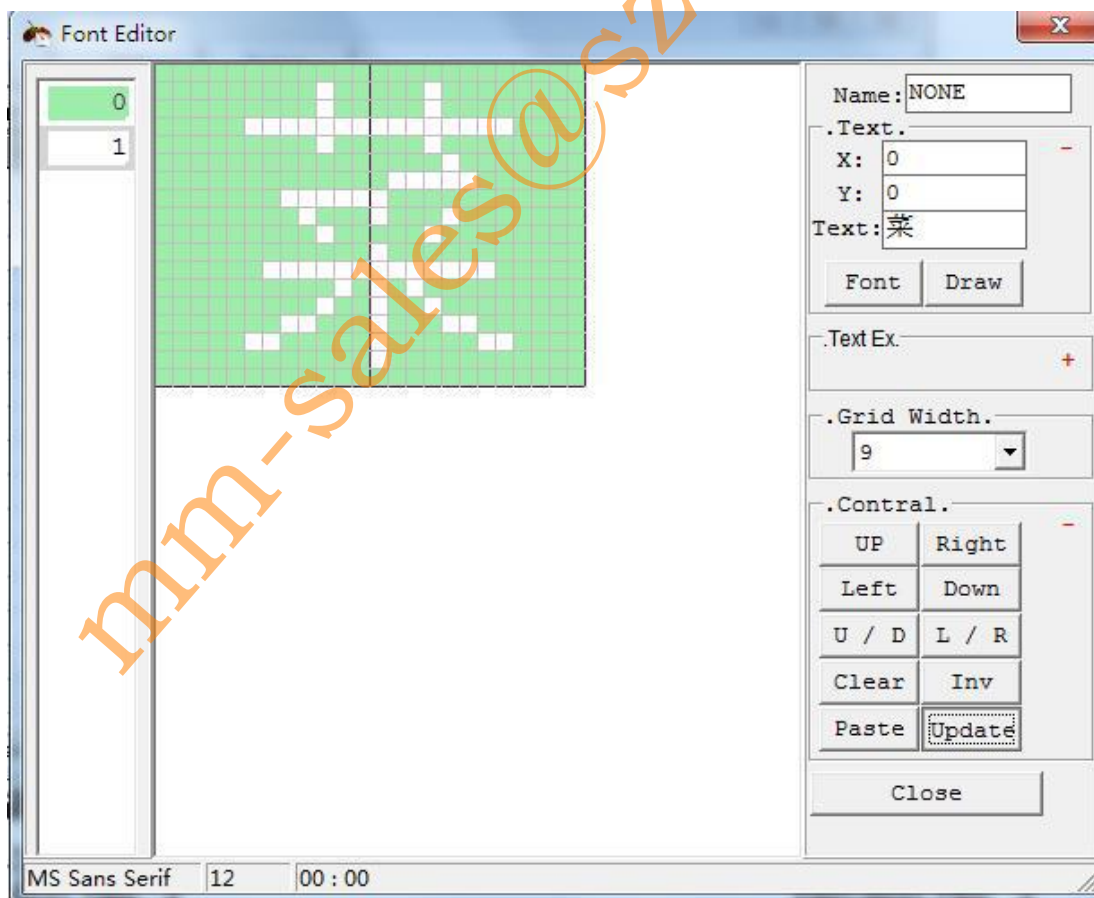
字体一般默认，字号大小选择 12 号，点击确定。



8) 点击 draw，即可在左边绿色框看到写入的中文：

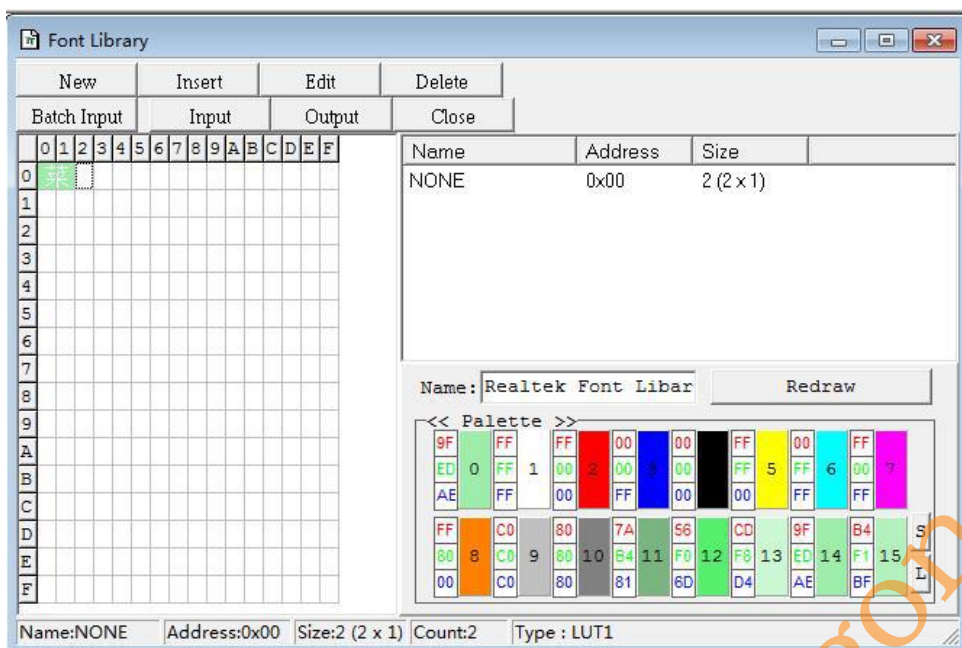


9) 通过右下角的 Control 里面的按钮对字体进行“(up,down,left,right)上下左右”等移动，字移动居中位置后点击“update”

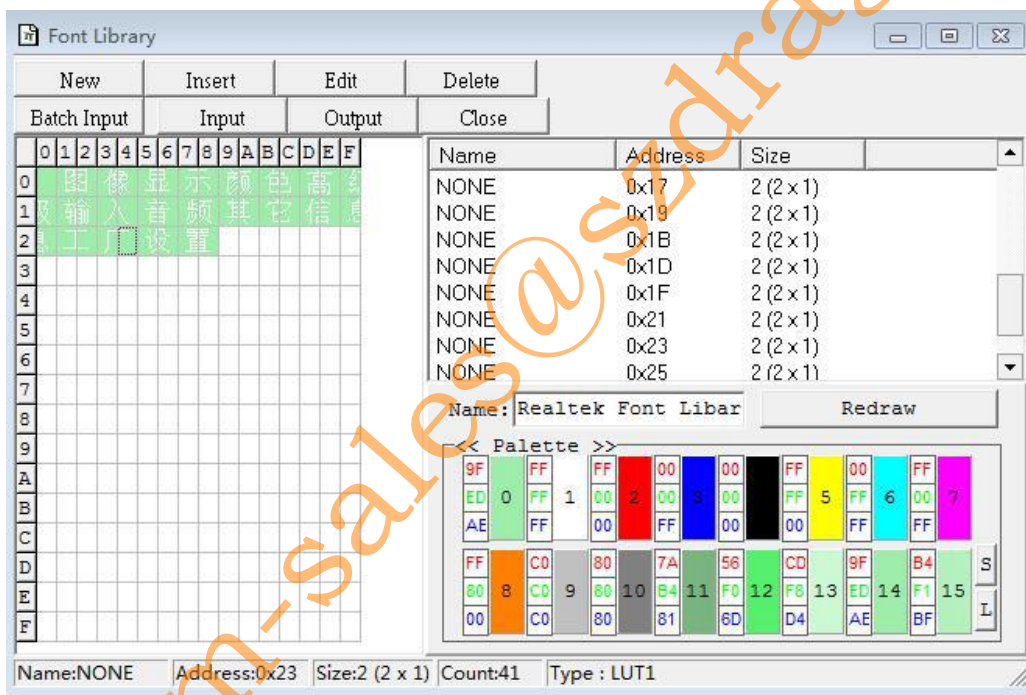




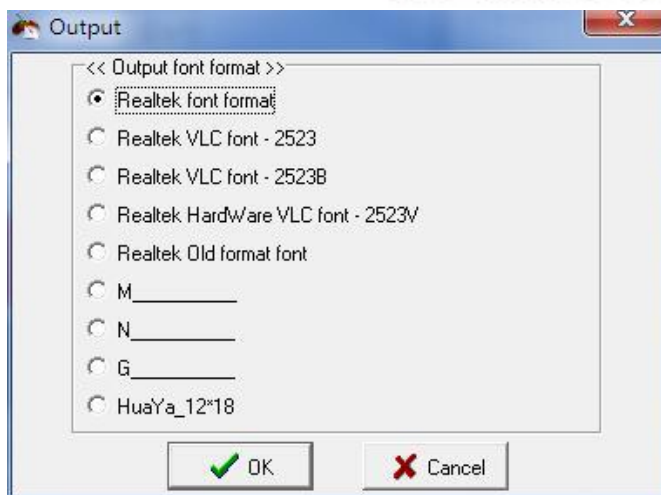
10) 写完后点击 “Close”



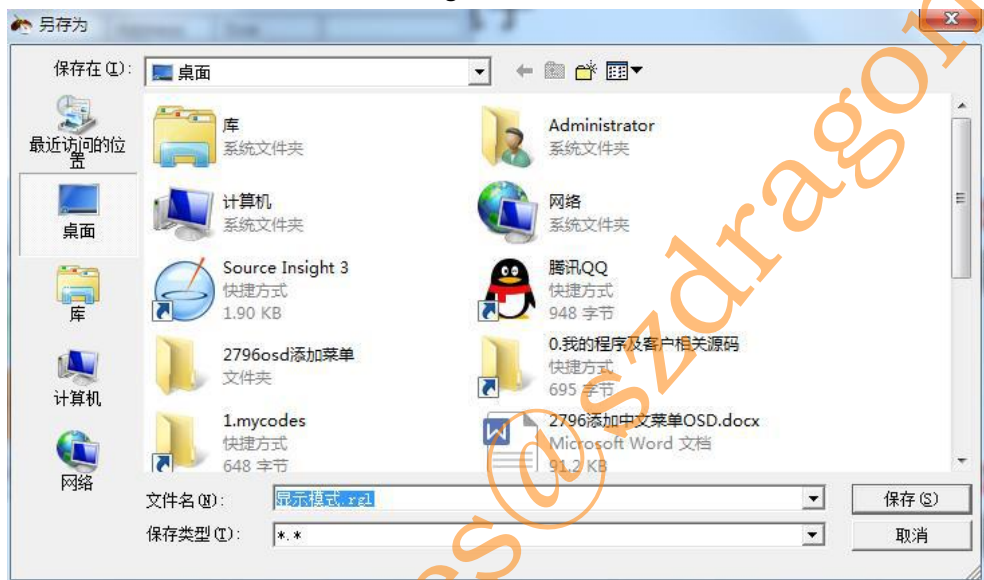
11) 以此上述步骤继续添加文字



12) output->第一个 Realtek font format-》 OK

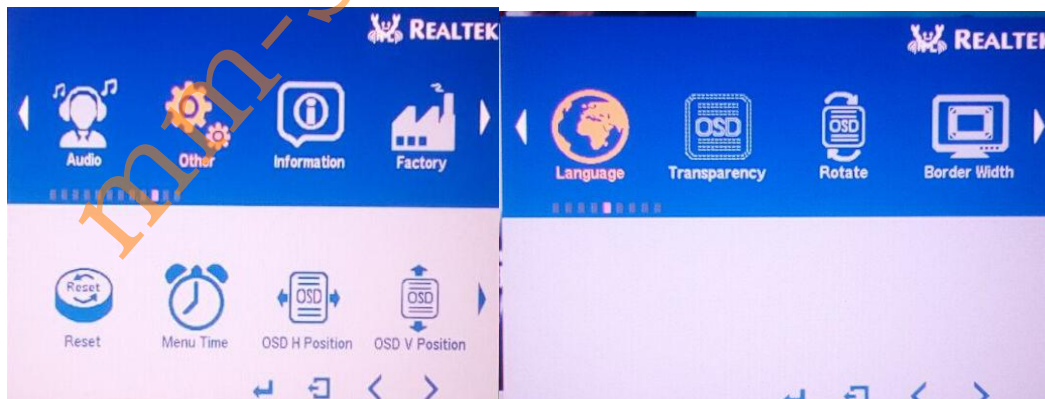


13) 将中文字库保存为“显示模式.rgl”文件

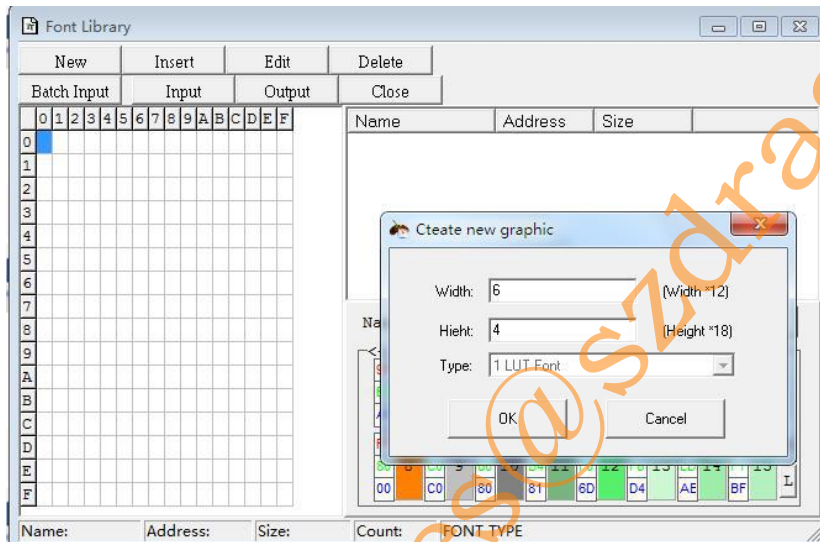
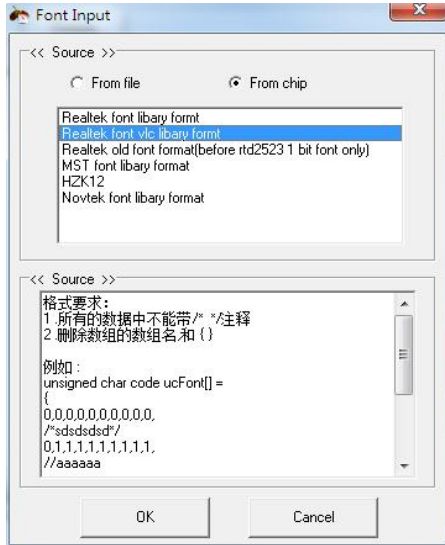


二. 程序菜单部分

1) 在菜单选项“Other”的 language 中添加中文选项

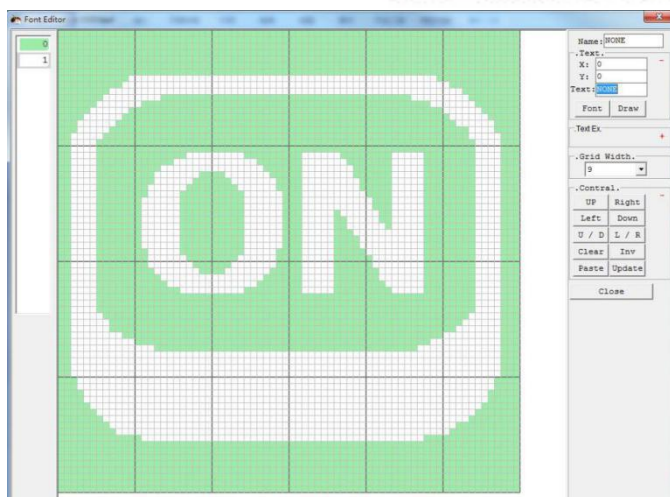


2) 用 OsdEditor.exe 画出要中英文两个图标-将数据复制

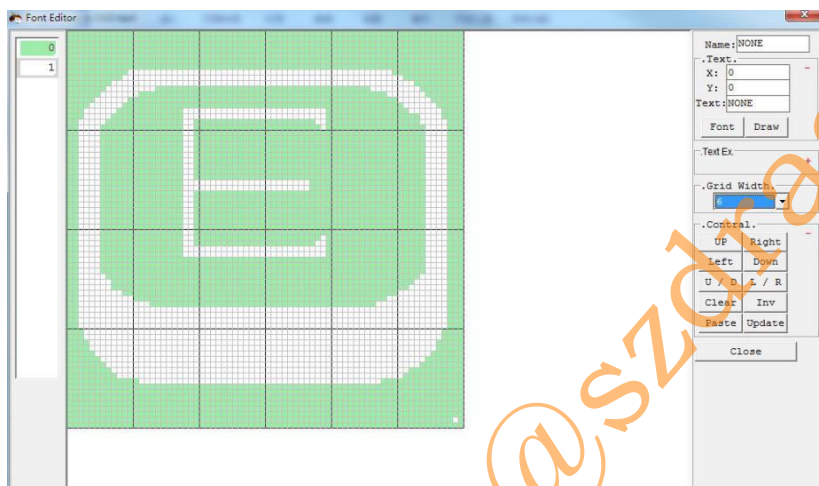


New -> 设置 width: 6 ; Height: 4

3) 采用 RTD20140sdFontDynamicIcon_2. c-菜单下层图标数据
这里在 ON 图标的基础上进行修改。

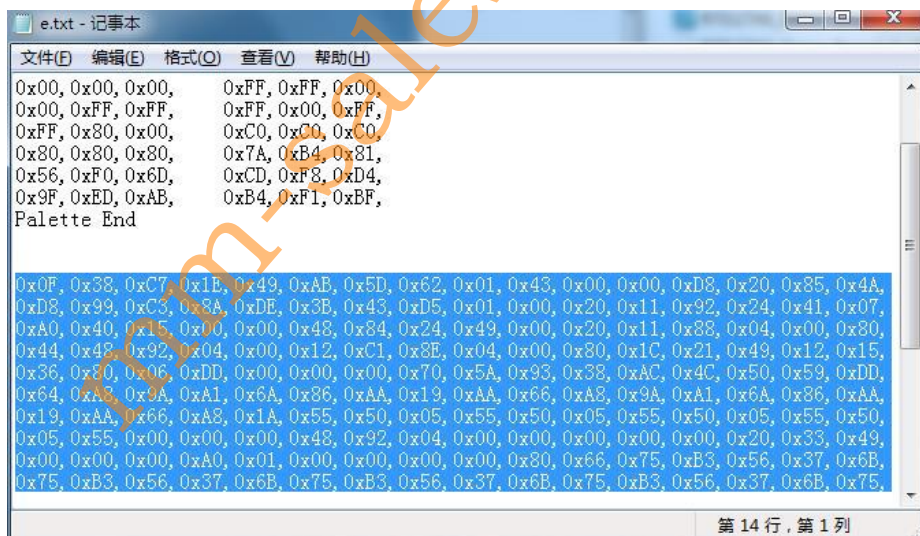


4) 手动修改图标如下图:



5) 点击“output”，保存为“e.txt”

6) 打开 e.txt 文件，复制蓝色部分数据:



7) 在 RTD20140sdFontDynamicIcon_2. c-菜单下层图标数据

添加该新图标的数-tFONT1_ICON_language_english[]



```
RTD2014OsdFontDynamicIcon_2.c
03813: BYTE code FONT1_ICON_language_english[] =
03814: {
03815: 0x0f,0x3e,0xc7,0x81,0x96,0x24,0xd5,0xab,0x01,0x5d,0x00,0x20,
03816: 0x07,0x2b,0xe1,0xa4,0x73,0x24,0xa6,0x4e,0x9a,0x55,0x3d,0xac,
03817: 0x35,0xc3,0x5a,0x03,0x00,0x49,0x92,0x24,0x49,0x42,0x24,0x00,
03818: 0x00,0x00,0x00,0x00,0x24,0x49,0x92,0x24,0x49,0x02,0x00,0x00,
03819: 0x00,0x00,0x49,0x92,0x24,0x49,0x42,0x24,0x00,0x00,0x00,0x00,
03820: 0x90,0x24,0x49,0x92,0x24,0x44,0x02,0x00,0x00,0x00,0x00,0xc8,
03821: 0x68,0x02,0x09,0xb6,0x02,0xc4,0x36,0x29,0x6e,0x56,0x75,0x53,
03822: 0x05,0x4d,0x0d,0x6b,0xc4,0xb0,0xd6,0xc0,0x6b,0xc4,0xb0,0xd6,
03823: 0x0c,0x6b,0xc4,0xd4,0x59,0x6b,0xc6,0xb3,0xd6,0xc0,0x6b,0xc4,
03824: 0x0d,0x6b,0xc0,0x00,0x00,0x00,0x12,0xb3,0x4a,0x6e,0x0a,0x62,
03825: 0x0e,0xab,0x08,0x77,0xd5,0x31,0x00,0x00,0x00,0x00,0xab,0xc0,
03826: 0xbb,0xa0,0x00,0x00,0x00,0x00,0x00,0xc4,0xa5,0xb9,0x26,0x95,
03827: 0x73,0x6d,0x9d,0x73,0x01,0x00,0x00,0x00,0x80,0x65,0x38,0x3b,
03828: 0xab,0x55,0x54,0x55,0x75,0x53,0x05,0x4d,0x0d,0x6b,0xc4,
03829: 0x54,0x75,0x53,0xd5,0x4d,0x55,0xf7,0xa8,0xea,0x5e,0x5d,0xd5,
03830: 0x9d,0xaa,0x61,0xad,0x99,0x63,0xad,0x99,0x55,0x6b,0xc4,0x58,
03831: 0x6b,0xc4,0x00,0xd6,0xd6,0x6b,0xc4,0x00,0xd6,0xd6,0x6b,0xc4,
03832: 0xb0,0xd6,0xa8,0x5a,0xb5,0x55,0x4d,0xeb,0xa,0x6c,0x94,0x34,
03833: 0x00,0x00,0x00,0x60,0x16,0xf4,0x60,0x56,0x02,0x64,0x00,0x00,
03834: 0x00,0x80,0x6b,0xe4,0x93,0xeb,0xd6,0xb3,0xc4,0x75,0x6d,0x7d,
03835: 0xe5,0x02,0x00,0x00,0x00,0x80,0x92,0xaa,0x3a,0x5b,0x55,0x59,
03836: 0xa9,0x3a,0x00,0x00,0x00,0x00,0x50,0xdd,0x52,0x5d,0xfd,0xa8,
03837: 0xea,0x96,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,
03838: 0xea,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,0xae,
03839: 0xb6,0xaa,0x24,0xe3,0xa4,0x12,0x96,0x15,0x0e,0x00,0x00,0x00,
03840: 0x00,0x24,0x42,0x92,0x24,0x49,0x92,0x00,0x00,0x00,0x00,0x40,
03841: 0x22,0x24,0x49,0x92,0x24,0x09,0x00,0x00,0x00,0x00,0x24,0x42,
03842: 0x92,0x24,0x09,0x00,0x00,0x00,0x00,0x40,0x22,0x24,0x49,
03843: 0x92,0x24,0x09,0x00,0x54,0x37,0x55,0xdd,0xd4,0xea,0xa6,0x32,
03844: 0xc4,0xc4,0xde,0xa9,0x34,0x49,0xd0,0x9,0xd0,0xf,
03845: code BYTE *FONT1_MAIN_ICON_TABLE_2[] =
03846: {
03847:
03848: }
03849:
03850:
03851:
03852: code BYTE *FONT1_MAIN_ICON_TABLE_2[] =
03853: {
```

BYTE code FONT1_ICON_language_english[] =
BYTE code FONT1_ICON_language_chinese[] =

8) 在代码 void (*OperationTable[])(void) 中可以看到“language 图标”前一个是“MenuOtherOsdVPos,”

```
MenuAdvanceUpD1LaneCountAdjust
MenuAdvanceUpD6LaneCountAdjust
MenuHotKeyDccci
MenuHotKeySource
MenuHotKeyDisplayMode
MenuHotKeyInformation
MenuPanelUniformityMsgAdjust
MenuMessageDisplay
MenuSource_1P
MenuSource_2P
MenuSource_3P
MenuSource_4P
MenuSource_1P_ADJ
MenuSource_2P_ADJ
MenuSource_3P_ADJ
MenuSource_4P_ADJ
OperationTable
endif
12746:
12747: // other
12748: MenuOtherReset,
12749: MenuOtherMenuTime,
12750: MenuOtherOsdHPos,
12751: MenuOtherOsdVPos,
12752: MenuOtherLanguage,
12753: MenuOtherTransparency,
12754: MenuOtherRotate,
12755: MenuOtherBorderWidth,
12756: MenuOtherBorderColor,
12757:
12758: // display function adjust
12759: MenuDisplayFunDispRotateAdjust,
12760: MenuDisplayFunDispLatencyAdjust,
12761: MenuDisplayFunFreezeAdjust,
```

9) 在 RTD2014Osd.c 中找到函数 void MenuOtherOsdVPos(void); 在右键选项中加入:
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);

```
RTD2014Osd.c
MenuAdvanceFreeSync
if(_DP_INST_SUPPORT == _ON)
endif
MenuAdvanceUpD1LaneCount
MenuAudioVolume
MenuAudioMute
MenuAudioStandAlone
MenuAudioSource
MenuAudioSoundMode
MenuOtherReset
MenuOtherMenuTime
MenuOtherOsdHPos
MenuOtherOsdVPos
MenuOtherLanguage
MenuOtherTransparency
MenuOtherRotate
MenuOtherBorderWidth
MenuOtherBorderColo
MenuDisplayFunDispRotateAdjust
MenuDisplayFunDispLatencyAdjust
MenuDisplayFunFreezeAdjust
MenuDisplayFunLatencyAdjust
MenuDisplayFunPositionV
MenuDisplayFunPositionH
MenuDisplayFunPositionVAdjust
MenuDisplayFunPositionHAdjust
MenuDisplayFunTransparencyAdjust
MenuDisplayFunSizeAdjust
MenuPictureBacklightAdjust
MenuPictureBrightnessAdjust
MenuPictureContrastAdjust
MenuPictureSharpnessAdjust
MenuDisplayFunPosAdjust
05278: OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
05279: OsdDispMainMenuIconPage(_UP, _ICON_PAGE_MAIN_2);
05280: OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_OTHER_0);
05281: break;
05282: default:
05283: break;
05284: }
05285: }
05286: }
05287: void MenuOtherOsdVPos(void)
05288: {
05289: switch(GET_KEYMESSAGE())
05290: {
05291: case _MENU_KEY_MESSAGE:
05292: g_usBackValue = GET_OSD_VPOS();
05293:
05294: SET_OSD_STATE(_MENU_OTHER_OSD_VPOS_ADJUST);
05295: OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _INSUBSET);
05296: OsdDispSliderAndNumber(_MENU_OTHER_OSD_VPOS_ADJUST, GET_OSD_VPOS());
05297: break;
05298:
05299: case _RIGHT_KEY_MESSAGE:
05300: OsdDispClearSliderAndNumber();
05301: SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
05302: OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
05303: OsdDispMainMenuIconPage(_UP, _ICON_PAGE_OTHER_1);
05304: OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);
05305:
05306: break;
05307:
05308: case _LEFT_KEY_MESSAGE:
05309: SET_OSD_STATE(_MENU_OTHER_OSD_VPOS);
05310: break;
```

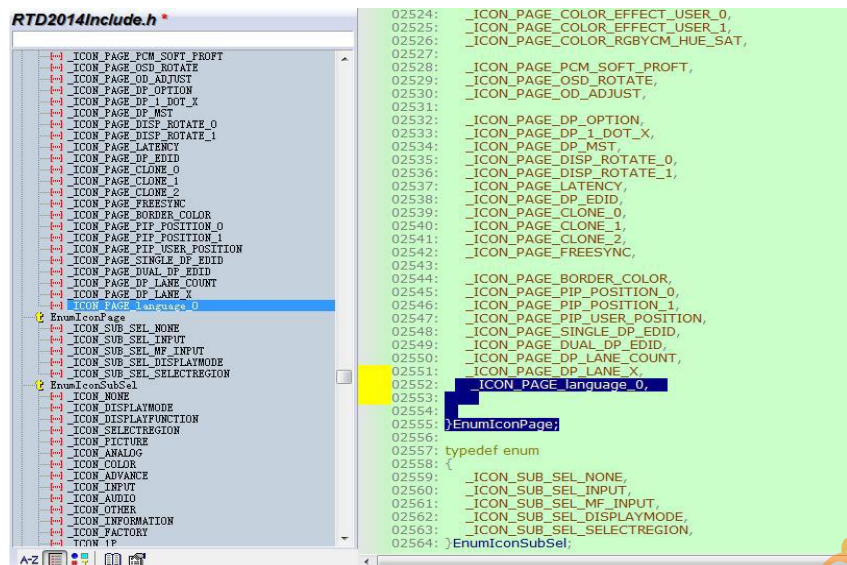
在 language 图标后一个函数 void MenuOtherTransparency(void)的左键也加入:
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);

因为没有_ICON_PAGE_language_0, 所以得到图标定义处去添加:



RTD2014Include.h->枚举 EnumIconPageIndex(2556 板子) 或者 EnumIconPage (2796 板子)

末尾添加 _ICON_PAGE_language_0



9) 接着到画图标的地方添加:

RTD2014OsdDisplay.c->void OsdDispMainMenuIconPage(BYTE ucUpDown, BYTE ucState)中

```
case _ICON_PAGE_language_0:
    OsdDispMainOptionIcon((ucUpDown + 0), _ICON_LANGUAGE_ENGLISH, pOsdItemColor[0]);
    OsdDispMainOptionIcon((ucUpDown + 1), _ICON_LANGUAGE_CHINESE, pOsdItemColor[1]);
    OsdDispMainOptionIcon((ucUpDown + 2), _ICON_NONE, pOsdItemColor[2]);
    OsdDispMainOptionIcon((ucUpDown + 3), _ICON_NONE, pOsdItemColor[3]);
    break;

/*
OsdDispMainMenuIconString((ucUpDown + 0), _ICON_LANGUAGE_english, pucOsdItemColor[0]);
OsdDispMainMenuIconString((ucUpDown + 1), _ICON_LANGUAGE_chinese, pucOsdItemColor[1]);
OsdDispMainMenuIconString((ucUpDown + 2), _ICON_NONE, pucOsdItemColor[2]);
OsdDispMainMenuIconString((ucUpDown + 3), _ICON_NONE, pucOsdItemColor[3]);
*/
break;
```




```
RTD2014OsdDisplay.c
...
OsdDispMainMenuIconString((ucUpDown + 2), _ICON_NONE, pOsdItemColor[2]);
OsdDispMainMenuIconString((ucUpDown + 3), _ICON_NONE, pOsdItemColor[3]);
break;
...
case _ICON_PAGE_OTHER_0:
OsdDispMainMenuIconString((ucUpDown + 0), _ICON_RESET, pOsdItemColor[0]);
OsdDispMainMenuIconString((ucUpDown + 1), _ICON_MENU_TIME, pOsdItemColor[1]);
OsdDispMainMenuIconString((ucUpDown + 2), _ICON_OSD_HPOS, pOsdItemColor[2]);
OsdDispMainMenuIconString((ucUpDown + 3), _ICON_OSD_VPOS, pOsdItemColor[3]);
break;
...
case _ICON_PAGE_OTHER_1:
OsdDispMainMenuIconString((ucUpDown + 0), _ICON_LANGUAGE, pOsdItemColor[0]);
OsdDispMainMenuIconString((ucUpDown + 1), _ICON_TRANSPARENCY, pOsdItemColor[1]);
OsdDispMainMenuIconString((ucUpDown + 2), _ICON_ROTATE, pOsdItemColor[2]);
OsdDispMainMenuIconString((ucUpDown + 3), _ICON_BORDER_WIDTH, pOsdItemColor[3]);
break;
...
case _ICON_PAGE_language_0:
OsdDispMainMenuIconString((ucUpDown + 0), _ICON_LANGUAGE_chinese, pOsdItemColor[0]);
OsdDispMainMenuIconString((ucUpDown + 1), _ICON_LANGUAGE_english, pOsdItemColor[1]);
OsdDispMainMenuIconString((ucUpDown + 2), _ICON_NONE, pOsdItemColor[2]);
OsdDispMainMenuIconString((ucUpDown + 3), _ICON_NONE, pOsdItemColor[3]);
break;
...
case _ICON_PAGE_OTHER_2:
OsdDispMainMenuIconString((ucUpDown + 0), _ICON_BORDER_COLOR, pOsdItemColor[0]);
OsdDispMainMenuIconString((ucUpDown + 1), _ICON_BORDER_STYLE, pOsdItemColor[1]);
OsdDispMainMenuIconString((ucUpDown + 2), _ICON_BORDER_WIDTH, pOsdItemColor[2]);
OsdDispMainMenuIconString((ucUpDown + 3), _ICON_BORDER_COLOR, pOsdItemColor[3]);
break;
...

```

10) 接着: void OsdDispMainMenuIconString(BYTE ucIconPos, WORD usIcon, BYTE ucColor)函数中查看画图标函数-> void OsdFontVLCDynamicLoadIcon(BYTE ucIconPos, WORD usIcon); 看函数的走向:

```
RTD2014OsdFontDynamicIcon_1.c
...
void OsdFontVLCDynamicLoadIcon(BYTE ucIconPos, WORD usIcon)
{
    BYTE ucOsdRotateStatus = _OSD_ROTATE_DEGREE_0;
    if((_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE) || (_OSD_ROTATE_FUNCTION == GET_OSD_ROTATE_STATUS))
    {
        ucOsdRotateStatus = GET_OSD_ROTATE_STATUS();
    }
   ScalerOsdHardwareVLC((FONT1_MAIN_ICON_TABLE_1[usIcon - 1], VLC_TABLE_1[ucOsdRotateStatus]);
}
...
void OsdFontVLCDynamicLoadIcon(BYTE ucIconPos, WORD usIcon)
{
    if((usIcon >= _ICON_INPUT_NO_PORT) && (usIcon < _ICON_INPUT_END))
    {
        //Input Icon
        OsdFontVLCDynamicLoadInputIcon(ucIconPos, usIcon);
    }
    else if(usIcon == _ICON_INFORMATION_HINT) //for information hint
    {
        ScalerOsdHardwareVLC((FONT1_MAIN_ICON_TABLE_1[_ICON_INFORMATION_HINT], VLC_TABLE_1[ucOsdRotateStatus]);
    }
    else
    {
        //Main Icon
        if(usIcon < _ICON_ON)
        {
            OsdFontVLCDynamicLoadMainIcon_1(ucIconPos, usIcon);
        }
        else
        {
            OsdFontVLCDynamicLoadMainIcon_2(ucIconPos, (usIcon - _ICON_ON));
        }
    }
}
...

```

RTD2014Include.h->枚举 EnumIconIndex 在末尾加入: -放入两个语言图标:

typedef enum

```
{
    ...
    _ICON_2_LANE,
    _ICON_4_LANE,
    ...
    _ICON_LANGUAGE_english,
    _ICON_LANGUAGE_chinese,
    ...
    OPTION_ICON_END,
}
```

} EnumIconIndex;

11) 接着进一步寻找图标的存放数组:

void OsdFontVLCDynamicLoadMainIcon_2(BYTE ucIconPos, WORD usIcon);



```
RTD2014OsdFontDynamicIcon_2.c
...
03971:
03972: //*****
03973: // FUNCTION DEFINITIONS
03974: //*****
03975: //-----
03976: // Description : Osd Load Hardware VLC Font
03977: // Input Value :
03978: // Output Value :
03979: //-----
03980: void OsdFontVLCLoadMainIcon_2(BYTE ucIconPos, WORD usIcon)
03981: {
03982:     BYTE ucOsdRotateStatus = _OSD_ROTATE_DEGREE_0;
03983:     if((_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE) || (_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE))
03984:     {
03985:         ucOsdRotateStatus = GET_OSD_ROTATE_STATUS();
03986:     }
03987:     ScaledOsdHardwareVLC(tFONT1_MAIN_ICON_TABLE_2[usIcon], VLC_TABLE_SIZE(tFONT1_MAIN_ICON_TABLE_2));
03988: }
```

12) RTD2014OsdFontDynamicIcon_2.c 中发现字符数组名字存放在一个枚举中: -

code BYTE *tFONT1_MAIN_ICON_TABLE_2[] = //两个图标数组

//on_off page

tFONT1_ICON_ON,

tFONT1_ICON_OFF,

末尾添加:

tFONT1_ICON_2_LANE,

tFONT1_ICON_4_LANE

//C 图标在前

//tFONT1_ICON_language_chinese,

//tFONT1_ICON_language_english,

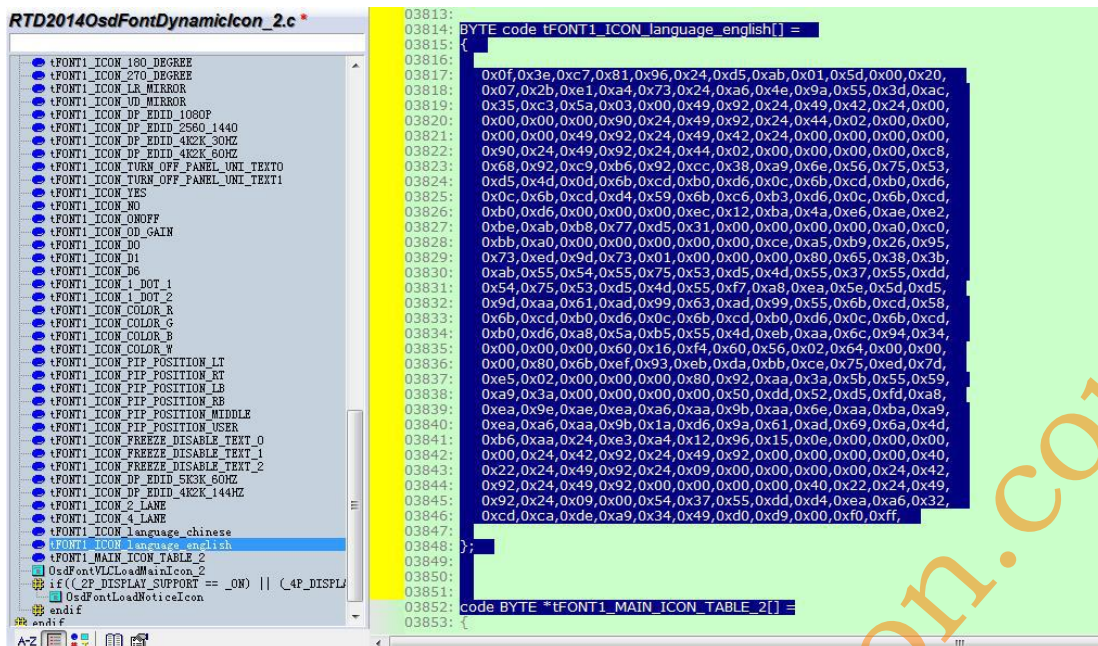
//决定图标的前后顺序-这个 E 图标在前

tFONT1_ICON_language_english,

tFONT1_ICON_language_chinese,

```
RTD2014OsdFontDynamicIcon_2.c
...
03931: tFONT1_ICON_OD_GAIN,
03932:
03933: tFONT1_ICON_D0,
03934: tFONT1_ICON_D1,
03935: tFONT1_ICON_D6,
03936: tFONT1_ICON_1_DOT_1,
03937: tFONT1_ICON_1_DOT_2,
03938:
03939: tFONT1_ICON_COLOR_R,
03940: tFONT1_ICON_COLOR_G,
03941: tFONT1_ICON_COLOR_B,
03942: tFONT1_ICON_COLOR_W,
03943:
03944: tFONT1_ICON_PIP_POSITION_LT,
03945: tFONT1_ICON_PIP_POSITION_RT,
03946: tFONT1_ICON_PIP_POSITION_LB,
03947: tFONT1_ICON_PIP_POSITION_RB,
03948: tFONT1_ICON_PIP_POSITION_MIDDLE,
03949: tFONT1_ICON_PIP_POSITION_USER,
03950:
03951: tFONT1_ICON_FREEZE_DISABLE_TEXT_0,
03952: tFONT1_ICON_FREEZE_DISABLE_TEXT_1,
03953: tFONT1_ICON_FREEZE_DISABLE_TEXT_2,
03954: tFONT1_ICON_DP_EDID_5K3K_60HZ,
03955: tFONT1_ICON_DP_EDID_4K2K_144HZ,
03956: tFONT1_ICON_2_LANE,
03957: tFONT1_ICON_4_LANE,
03958: tFONT1_ICON_language_chinese,
03959: tFONT1_ICON_language_english,
03960:
03961:
03962: };
03963:
03964: //*****
```

并在此文件 RTD2014OsdFontDynamicIcon_2.c 程序最上面将图标数组数据添加:



14) 在 RTD2014Osd.c 函数中找到

void MenuOtherLanguage(void);--画菜单图标

void MenuOtherLanguageAdjust(void);--对应语言功能调节

将中英文两个图标功能加进去:

1>修改 **void MenuOtherLanguage(void);**

void MenuOtherLanguage(void)

```
{
    switch(GET_KEYMESSAGE())
    {
        case _MENU_KEY_MESSAGE:
            g_usBackupValue = GET_OSD_LANGUAGE();
            SET_OSD_STATE(_MENU_OTHER_LANGUAGE_ADJUST);
            OsdDispMainMenuCursor(GET_OSD_STATE(),GET_OSD_STATE_PREVIOUS(),_INSUBSET);
            break;

            case _RIGHT_KEY_MESSAGE:
                OsdDispMainMenuIconPage(_DOWN,_ICON_PAGE_CLEAR);//
                OsdDispClearSelectColor(_DOWN);//

                SET_OSD_STATE(_MENU_OTHER_TRNAPARENCY);
                OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(),_OUTSUBSET);
                OsdDispSliderAndNumber(_MENU_OTHER_TRNAPARENCY,GET_OSD_TRANSPARENCY_STATUS());
                break;

            case _LEFT_KEY_MESSAGE:
                OsdDispMainMenuIconPage(_DOWN,_ICON_PAGE_CLEAR);//
                OsdDispClearSelectColor(_DOWN);//
```



```
SET_OSD_STATE(_MENU_OTHER_OSD_VPOS);
OsdDispMainMenuCursor(GET_OSD_STATE(),GET_OSD_STATE_PREVIOUS(),_OUTSUBSET);
OsdDispMainMenuIconPage(_UP,_ICON_PAGE_OTHER_0);
OsdDispSliderAndNumber(_MENU_OTHER_OSD_VPOS, GET_OSD_VPOS());
break;

case _EXIT_KEY_MESSAGE:
    OsdDispClearSliderAndNumber();
    SET_OSD_STATE(_MENU_OTHER);
OsdDispMainMenuCursor(GET_OSD_STATE(),GET_OSD_STATE_PREVIOUS(),_OUTSUBSET);
OsdDispMainMenuIconPage(_UP,_ICON_PAGE_MAIN_2);
OsdDispMainMenuIconPage(_DOWN,_ICON_PAGE_OTHER_0);
break;

default:
    break;
}
}
```

2>修改 void MenuOtherLanguageAdjust(void);-里面的函数修改在后面再进行。

3>后一个图标的左键加入：

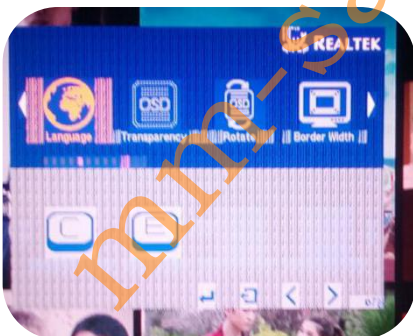
void MenuOtherTransparency(void)

case _LEFT_KEY_MESSAGE:

```
OsdDispMainMenuIconPage(_DOWN,_ICON_PAGE_CLEAR);//
OsdDispClearSelectColor(_DOWN);//
```

```
OsdDispClearSliderAndNumber();
SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
OsdDispMainMenuIconPage(_DOWN,_ICON_PAGE_language_0);
break;
```

4>菜单修改后图片为：



发现图标乱码，原因是 void MenuOtherLanguageAdjust(void)里面还未修改，受里面的影响。不过在后面的添加中发现是用了写字符串函数来画图标才显示乱码，问题解决详见后面 bug1.



15) RTD20140sdFontProp. c->OsdPropPutStringCenter () 函数修改, 添加中文菜单的地址
--2556 函数中应用

```
//-----  
// Description :  
// Input Value : None  
// Output Value : None  
//-----  
void OsdPropPutStringCenter(BYTE ucRow, BYTE ucCol, BYTE ucWidth, BYTE ucFptsSelect, BYTE ucString, BYTE ucColor, BYTE  
ucLanguage)  
{  
    bit bEndFlag = 0;  
    BYTE pucTemp[27] = {0};  
    bit bMode = _CENTER;  
    BYTE *pucFontTable = 0;  
    BYTE *pucArray = 0;  
    BYTE ucByte0 = 0x6C; // palette index 0, 1bit 256 ~ 511  
    BYTE pucDataTemp[16] = {0}; // replace pData  
  
    #if(_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE)  
        if((GET_OSD_ROTATE_STATUS() == _OSD_ROTATE_DEGREE_90) ||  
            (GET_OSD_ROTATE_STATUS() == _OSD_ROTATE_DEGREE_270))  
        {  
            ucByte0 = 0x6E;  
        }  
    #endif  
    pucArray = OsdPropGetStringTableAddress(ucString);  
  
    if(ucLanguage == _CHINESE_T)  
    {  
        // pucFontTable = tFONT_CHINESE; //中文字库数据数组  
  
        pucArray = OsdPropGetStringTableAddress_ChnS(ucString); //自造函数, 获取单文字地址  
        pucFontTable = OsdGetChnTFontPage(ucString); //自造函数  
  
    }  
    else  
    {  
        pucFontTable = tFONT_EUROPE; //英文字库数据数组  
        //-----  
        pucArray = OsdPropGetStringTableAddress(ucString);  
        //-----  
    }  
  
    //////////////////////////////////////  
    if(pucFontTable == 0)  
        pucFontTable = tFONT_EUROPE;  
    if(pucArray == 0)  
        pucArray = OsdPropGetStringTableAddress(ucString); //  
  
    /*
```



```
pucDataTemp[0] = ucLanguage;

while(pucDataTemp[0] != _ENGLISH)
{
    if(*pucArray == _END_)
    {
        pucDataTemp[0]--;
    }
    pucArray++;
}
*/

pucDataTemp[1] = OsdPropGetFontPointer(ucFptsSelect);

// reorder
SCALEROSD_FONT_ROTATE_SETTING_SET(SCALEROSD_FONT_ROTATE_SETTING_GET() | _BIT6);
SCALEROSD_FONT_ROTATE_SETTING_SET(SCALEROSD_FONT_ROTATE_SETTING_GET() & (~(_BIT7 | _BIT5)));

pucDataTemp[2] = 0;
pucDataTemp[3] = 0;
pucDataTemp[4] = 0;
pucDataTemp[7] = 0;
pucDataTemp[10] = 0;

while(*(pucArray + pucDataTemp[2]) != _END_)
{
    switch(*(pucArray + pucDataTemp[2]))
    {
        case _END_:
            break;
        default:
            pucDataTemp[3] = *(pucArray + pucDataTemp[2]);
            if(ucLanguage == _ENGLISH)
            {
                pucDataTemp[4] = pucDataTemp[4] + tOSD_CHARWIDTH_EUROPE[pucDataTemp[3]][1];
            }
            else if(ucLanguage == _CHINESE_T)
            {
                //pucDataTemp[4] = pucDataTemp[4] + tOSD_CHARWIDTH_CHINESE[pucDataTemp[3]][1];

                if(pucFontTable == tFONT_EUROPE)
                    pucDataTemp[4] = pucDataTemp[4] + tOSD_CHARWIDTH_EUROPE[pucDataTemp[3]][1];
                else
                    pucDataTemp[4] = pucDataTemp[4] + 12;
            }
            else
            {
                pucDataTemp[4] = pucDataTemp[4] + 12;
            }
            pucDataTemp[2] ++;
            break;
    }
}
```




```
    }  
}  
pucDataTemp[5] = ucCol + (((((WORD)((ucWidth) * 12) - pucDataTemp[4]) / 2)) / 12);  
pucDataTemp[0] = (((((WORD)((ucWidth) * 12) - pucDataTemp[4]) / 2)) % 12);  
  
/*//for Right mode  
ucCol = (ucColEnd - (ucCharWidth/12) -1);  
ucBlinking = 12 - (ucCharWidth % 12);  
*/  
  
pucDataTemp[4] = 0;  
pucDataTemp[2] = 0;  
  
if(pucDataTemp[0] == 0)  
{  
    bMode = _NORMAL;  
}  
else  
{  
    bMode = _CENTER;  
}  
  
while(*(pucArray + pucDataTemp[2]) != _END_)  
{  
    switch(*(pucArray + pucDataTemp[2]))  
    {  
        case _END_:  
            bEndFlag = 1;  
            break;  
  
        default:  
  
            if(bMode == _CENTER)  
            {  
                pucDataTemp[6] = _;  
            }  
            else  
            {  
                pucDataTemp[6] = *(pucArray + pucDataTemp[2]);  
            }  
  
            if(ucLanguage == _ENGLISH)  
            {  
                if(bMode == _CENTER)  
                {  
                    pucDataTemp[4] = pucDataTemp[0];  
                }  
                else  
                {  
                    pucDataTemp[4] = tOSD_CHARWIDTH_EUROPE[pucDataTemp[6]][1];  
                }  
            }  
  
            else if(ucLanguage == _CHINESE_T)  
            {  
                if(bMode == _CENTER)  
                {  

```



```
pucDataTemp[4] = pucDataTemp[0];
}
else
{
// pucDataTemp[4] = tOSD_CHARWIDTH_CHINESE[pucDataTemp[6]][1];
////////////////////////////////////
if(pucFontTable == tFONT_EUROPE)
pucDataTemp[4] = tOSD_CHARWIDTH_EUROPE[pucDataTemp[6]][1];
else
pucDataTemp[4] = 12;
////////////////////////////////////
}
}
else
{
pucDataTemp[4] = 12;
}
pucDataTemp[2] ++;
break;
}

if(pucDataTemp[7] == 0)
{
pucDataTemp[7] = pucDataTemp[4];
}

SCALEROSD_FONT_ROTATE_SETTING_SET(SCALEROSD_FONT_ROTATE_SETTING_GET() | _BIT3); // first bits shift direction:
left shift
SCALEROSD_FONT_ROTATE_SETTING_SET(SCALEROSD_FONT_ROTATE_SETTING_GET() & (~_BIT2)); // left bits shift direction:
righ shift
SCALEROSD_FONT_ROTATE_1_2_BITS_SHIFT_SET(((pucDataTemp[4] - pucDataTemp[7]) << 4) | pucDataTemp[7]); // first bits
shift and second bits shift

pucDataTemp[8] = pucDataTemp[2];
pucDataTemp[9] = pucDataTemp[7];

if(bMode == _CENTER)
{
pucDataTemp[8] = 0;
pucDataTemp[6] = 0;
pucDataTemp[2] = 0;
bMode = _NORMAL;
}
for(pucDataTemp[0] = 0; pucDataTemp[0] < 9; pucDataTemp[0]++)
{
SCALEROSD_FONT_ROTATE_SETTING_SET(SCALEROSD_FONT_ROTATE_SETTING_GET() | _BIT0); // restart from last three
bytes

PDATA_WORD(7) = pucDataTemp[6] * 27 + pucDataTemp[0] * 3;

SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7)));
SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7) + 1));
SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7) + 2));

pucDataTemp[2] = pucDataTemp[8];
```



```
pucDataTemp[7] = pucDataTemp[9];

if((pucDataTemp[7] == 12) || (*(pucArray + pucDataTemp[2]) == _END_))
{
    SCALEROSD_FONT_ROTATE_INPUT_SET(0x00);
    SCALEROSD_FONT_ROTATE_INPUT_SET(0x00);
    SCALEROSD_FONT_ROTATE_INPUT_SET(0x00);
}
else
{
    pucDataTemp[3] = *(pucArray + pucDataTemp[2]);
    if(ucLanguage == CHINESE_T)
    {
        // pucDataTemp[4] = tOSD_CHARWIDTH_CHINESE[pucDataTemp[3]][1];
        if(pucFontTable == tFONT_EUROPE)
            pucDataTemp[4] = tOSD_CHARWIDTH_EUROPE[pucDataTemp[3]][1];
        else
            pucDataTemp[4] = 12;

    }
    else
    {
        pucDataTemp[4] = tOSD_CHARWIDTH_EUROPE[pucDataTemp[3]][1];
    }
    pucDataTemp[2]++;

    PDATA_WORD(7) = pucDataTemp[3] * 27 + pucDataTemp[0] * 3;

    SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7)));
    SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7) + 1));
    SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7) + 2));

    pucDataTemp[7] += pucDataTemp[4];
}

while((pucDataTemp[7] < 12) && (*(pucArray + pucDataTemp[2]) != _END_))
{
    SCALEROSD_FONT_ROTATE_3_L_BITS_SHIFT_SET((pucDataTemp[7] << 4) | pucDataTemp[7]); // third bits shift and
left bits shift

    switch(*(pucArray + pucDataTemp[2]))
    {
        case _END_:
            bEndFlag = 1;
            break;

        default:
            pucDataTemp[3] = *(pucArray + pucDataTemp[2]);

            if(ucLanguage == _ENGLISH)
            {
                pucDataTemp[4] = tOSD_CHARWIDTH_EUROPE[pucDataTemp[3]][1];
            }
    }
}
```



```
else if(ucLanguage == _CHINESE_T)
{
//pucDataTemp[4] = tOSD_CHARWIDTH_CHINESE[pucDataTemp[3]][1];
/////

if(pucFontTable == tFONT_EUROPE)
pucDataTemp[4] = tOSD_CHARWIDTH_EUROPE[pucDataTemp[3]][1];
    else
pucDataTemp[4] = 12;
        }
    else
    {
        pucDataTemp[4] = 12;
    }
    pucDataTemp[2] ++;
    break;
}

pucDataTemp[7] += pucDataTemp[4];

PDATA_WORD(7) = pucDataTemp[3] * 27 + pucDataTemp[0] * 3;

SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7)));
SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7) + 1));
SCALEROSD_FONT_ROTATE_INPUT_SET(*(pucFontTable + PDATA_WORD(7) + 2));
}

pucTemp[pucDataTemp[0] * 3 + 2] = SCALEROSD_FONT_ROTATE_ONPUT_GET();
pucTemp[pucDataTemp[0] * 3 + 1] = SCALEROSD_FONT_ROTATE_ONPUT_GET();
pucTemp[pucDataTemp[0] * 3] = SCALEROSD_FONT_ROTATE_ONPUT_GET();
}

if(pucDataTemp[7] <= 12)
{
    pucDataTemp[7] = 0;
}
else
{
    pucDataTemp[7] = pucDataTemp[7] - 12;
    pucDataTemp[2] --;
}

// Write the pro font data to sram
// COsdFxLoadFontDataAddrCal(ucBankupFontPointer);

// PDATA_WORD(7) = (WORD)pucDataTemp[1] * 9;
PDATA_WORD(7) = ((WORD)pucDataTemp[1] + 0x100) * 9;
PDATA_WORD(7) += g_usFontTableStart;

#if((_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE) || (_OSD_ROTATE_FUNCTION == _OSD_ROTATE_HARDWARE))
    if(GET_OSD_ROTATE_STATUS() != _OSD_ROTATE_DEGREE_0)
    {
        ScalerOsdFontRotateCtrl(GET_OSD_ROTATE_STATUS(), _ENABLE, _ENABLE);
        ScalerOsdCompressionCtrl(_OSD_DECODE_NON_COMPRESSED, _OSD_COMPRESSION_ENABLE);
    }
else
```




```
#endif
{
    ScalerOsdFontRotateCtrl(GET_OSD_ROTATE_STATUS(), _DISABLE, _DISABLE);
    ScalerOsdCompressionCtrl(_OSD_DECODE_COMPRESSED, _OSD_COMPRESSION_DISABLE);
}

if(PDATA_WORD(7) > 4095)
{
    ScalerOsdScrambleLoadFontAddrHsbSet();
}
else
{
    ScalerOsdScrambleLoadFontAddrHsbClr();
}

ScalerOsdAddrSet(_OSD_SRAM, _OSD_BYTEALL, PDATA_WORD(7), _DISABLE);

ScalerOsdBurstWriteDataPort(pucTemp, 27, GET_CURRENT_BANK_NUMBER(), _BURSTWRITE_DATA_OSD, _BURSTWRITE_FROM_XRAM);

// OSD HW needs time to process it and then write decompressed data into SRAM.
// The value 1ms is enough at all.
ScalerTimerDelayXms(1);

ScalerOsdCompressionCtrl(_OSD_DECODE_COMPRESSED, _OSD_COMPRESSION_DISABLE);

#if(_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE) || (_OSD_ROTATE_FUNCTION == _OSD_ROTATE_HARDWARE)
    if(GET_OSD_ROTATE_STATUS() != _OSD_ROTATE_DEGREE_0)
    {
        ScalerOsdFontRotateCtrl(GET_OSD_ROTATE_STATUS(), _DISABLE, _ENABLE);
    }
#endif

ScalerOsdScrambleLoadFontAddrHsbClr();

#if(_OSD_ROTATE_FUNCTION == _OSD_ROTATE_SOFTWARE)
    if(GET_OSD_ROTATE_STATUS() == _OSD_ROTATE_DEGREE_90)
    {
        PDATA_WORD(7) = (WORD)((g_ucOsdWidth - ucRow - 1) + g_ucOsdWidth * (pucDataTemp[5] + pucDataTemp[10]));
    }
    else if(GET_OSD_ROTATE_STATUS() == _OSD_ROTATE_DEGREE_270)
    {
        PDATA_WORD(7) = (WORD)(g_ucOsdHeight - 1 - pucDataTemp[5] - pucDataTemp[10]) * g_ucOsdWidth + ucRow;
    }
    else
#endif
#endif
{
    PDATA_WORD(7) = (WORD)ucRow * g_ucOsdWidth + pucDataTemp[5] + pucDataTemp[10];
}

// ScalerOsdCommandByte(PDATA_WORD(7) + g_usFontSelectStart, _OSD_BYTE1, pucDataTemp[1]);
ScalerOsdCommandAllByte(PDATA_WORD(7) + g_usFontSelectStart, ucByte0, pucDataTemp[1], ucColor);

pucDataTemp[10]++; // Record the length of prop string

// Increase the sram address pointer
pucDataTemp[1] = OsdPropSetFontPointer(ucFptsSelect, pucDataTemp[1]);
} // End of while(*(pucArray + stringcnt) != _END_)

if(ucFptsSelect == _PFONT_PAGE_0)
```



```
{
    g_ucFontPointer0 = pucDataTemp[1];
}
else if(ucFptsSelect == _PFONT_PAGE_1)
{
    g_ucFontPointer1 = pucDataTemp[1];
}
else if((ucFptsSelect == _PFONT_PAGE_2) ||
        ((ucFptsSelect >= _PFONT_PAGE_ITEM_1) && (ucFptsSelect <= _PFONT_PAGE_ITEM_8)))
{
    g_ucFontPointer2 = pucDataTemp[1];
}
}
```

16) 在此文件末尾添加函数：-注意要到文件首部声明。

BYTE *OsdGetChnTFontPage(BYTE ucString)

```
{
    BYTE *pArray = 0;

    switch(ucString)
    {
        case _STRING_PICTURE:
        case _STRING_BRIGHTNESS:
        case _STRING_CONTRAST:
        case _STRING_SHARPNESS:
        case _STRING_COLOR_TEMP:
        case _STRING_OTHER:
        case _STRING_RESET:
        case _STRING_DISP_ROTATE:
        case _STRING_VOLUME:
        case _STRING_LANGUAGE:
        case _STRING_INPUT:
            break;
        case _STRING_INFORMATION: //蓝色部分是我大概给了几个要翻译为中文的菜单
            pArray = tFONT_CHINESE; //这个中文地址就是刚刚第二步生成的码值文档 1234，可参考英文
            break;
    }
    return pArray;
}
```

完成以上项编译会报错，

1) OsdPropGetStringTableAddress_Chns 没有定义；

OsdPropGetStringTableAddress_Chns 参照 OsdPropGetStringTableAddress 仿照修改添加到文件末尾；

```
//-----
// Description :Chinese Address-中文 OSD 的排序中文地址
// Input Value : None
// Output Value : None
```



```
//-----  
BYTE* OsdPropGetStringTableAddress_Chns(BYTE ucString)  
{  
    BYTE *pucArray = 0;  
    switch(ucString)  
    {  
        case _STRING_PICTURE:  
            pucArray = tSTRING_PICTURE_CHS;  
            break;  
        case _STRING_DISPLAY:  
            pucArray = tSTRING_DISPLAY_CHS;  
            break;  
        case _STRING_COLOR:  
            pucArray = tSTRING_COLOR_CHS;  
            break;  
        case _STRING_ADVANCE:  
            pucArray = tSTRING_ADVANCE_CHS;  
            break;  
        case _STRING_INPUT:  
            pucArray = tSTRING_INPUT_CHS;  
            break;  
        case _STRING_AUDIO:  
            pucArray = tSTRING_AUDIO_CHS;  
            break;  
        case _STRING_OTHER:  
            pucArray = tSTRING_OTHER_CHS;  
            break;  
        case _STRING_INFORMATION:  
            pucArray = tSTRING_INFORMATION_CHS;  
            break;  
        case _STRING_FACTORY:  
            pucArray = tSTRING_FACTORY_CHS;  
            break;  
    }  
    return pucArray;  
}
```

17)RTD2014OsdFontProp.c 中添加中文数组地址

用 OSDEdit.ext 制作字库-svn_2556_osdfont.rgl

OsdPropGetStringTableAddress_Chns 中，中文字符串怎么添加，如英文与中文的区别（我举 picture 为例），



BYTE code tSTRING_PICTURE[] =

```
{  
    _P_,_i_,_c_,_t_,_u_,_r_,_e_,_END_,英文  
};
```

BYTE code tSTRING_PICTURE_CHS[] =

```
{  
    0x11,0x12,0x01,0x02,_END_,//你可能比较纳闷这中文数据怎么来的,  
};
```

打开 osd edit

点击 file 下的 new project ,

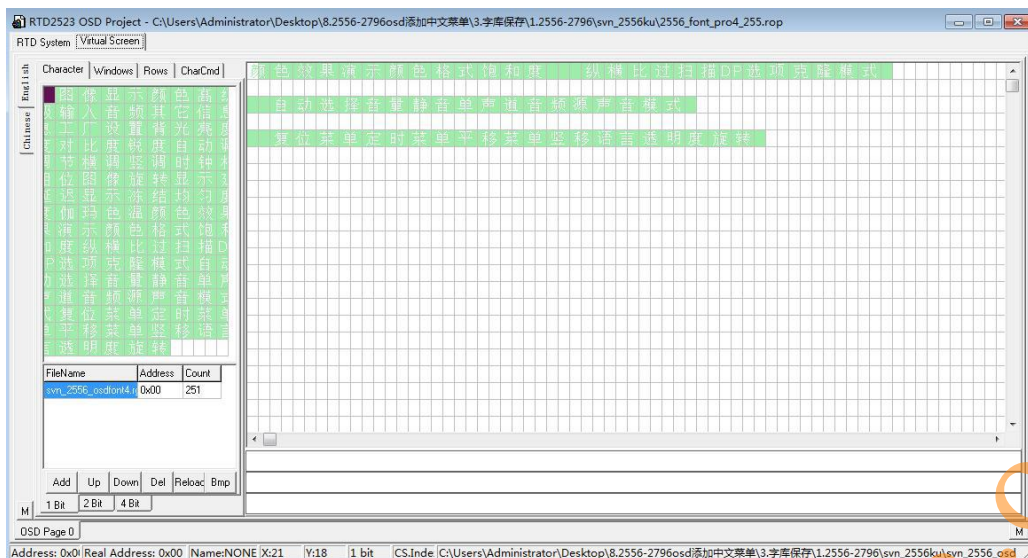
然后点击左下角的 add 添加前面自己制作生成的字库 “svn_2556_osdfont.rgl” ,

如图 1: 添加成功后如图 2 双击图像 (picture) 然后鼠标圈中图像即可得到 0x11, 0x12, 0x01, 0x02, 即为 picture 的中文添加到程序中 BYTE code tSTRING_PICTURE_CHS[] = { 0x11, 0x12, 0x01, 0x02, _END_ }; 其他也是相同的方法添加;



图 1





18) RTD2014OsdFontProp.c 中

将中文的地址数据写入 RTD2014OsdFontProp.c 文件中

```
BYTE code tSTRING_FAIL[] =
```

```
{ _F_, _a_, _i_, _l_, _END_,};
```

```
////////////////////////////////////
```

```
//中文
```

```
BYTE code tSTRING_PICTURE_CHS[] =
```

```
{
```

```
0x01,0x02,0x03,0x04, _END_,
```

```
};
```

```
BYTE code tSTRING_DISPLAY_CHS[] =
```

```
{
```

```
0x05,0x06,0x07,0x08, _END_,
```

```
};
```

```
BYTE code tSTRING_COLOR_CHS[] =
```

```
{
```

```
0x09,0x0A,0x0B,0x0C, _END_,
```

```
};
```

```
BYTE code tSTRING_ADVANCE_CHS[] =
```

```
{
```

```
0x0D,0x0e,0x0f,0x10, _END_,
```

```
};
```

```
BYTE code tSTRING_INPUT_CHS[] =
```

```
{
```

```
0x11,0x12,0x13,0x14, _END_,
```

```
};
```

```
BYTE code tSTRING_AUDIO_CHS[] =
```

```
{
```

```
0x15,0x16,0x17,0x18, _END_,
```

```
};
```

```
BYTE code tSTRING_OTHER_CHS[] =
```

```
{
```

```
0x19,0x1a,0x1b,0x1c, _END_,
```

```
};
```

```
BYTE code tSTRING_INFORMATION_CHS[] =
```

```
{
```

```
0x1d,0x1e,0x1f,0x20, _END_,
```



```
};
BYTE code tSTRING_FACTORY_CHS[] =
{
    0x21,0x22,0x23,0x24,0x25,0x26,0x27,0x28, _END_,
};
//////////
```



19. 在 **OsdDispMainMenuIconString** 函数末尾中添加中文判断的条件:

```
void OsdDispMainMenuIconString(BYTE ucIconPos, WORD usIcon, BYTE ucColor)
```

```
{
    BYTE ucRow = 4;
    BYTE ucCol = 4;
    BYTE ucFontPage = _PFONT_PAGE_0;
    WORD usIconLoad = 0;

    /*
    if((usIcon >= _ICON_A0_PORT) && (usIcon <= _ICON_D6_PORT))
    {
        OsdPropPutStringCenter(ucRow, ucCol, WIDTH(10), ucFontPage, (usIcon - _ICON_INPUT_END + _STRIN_END), COLOR(ucColor, _CP_BG), _ENGLISH);
    }
    else
    {
        OsdPropPutStringCenter(ucRow, ucCol, WIDTH(10), ucFontPage, usIcon - 1, COLOR(ucColor, _CP_BG), _ENGLISH);
    }
    */

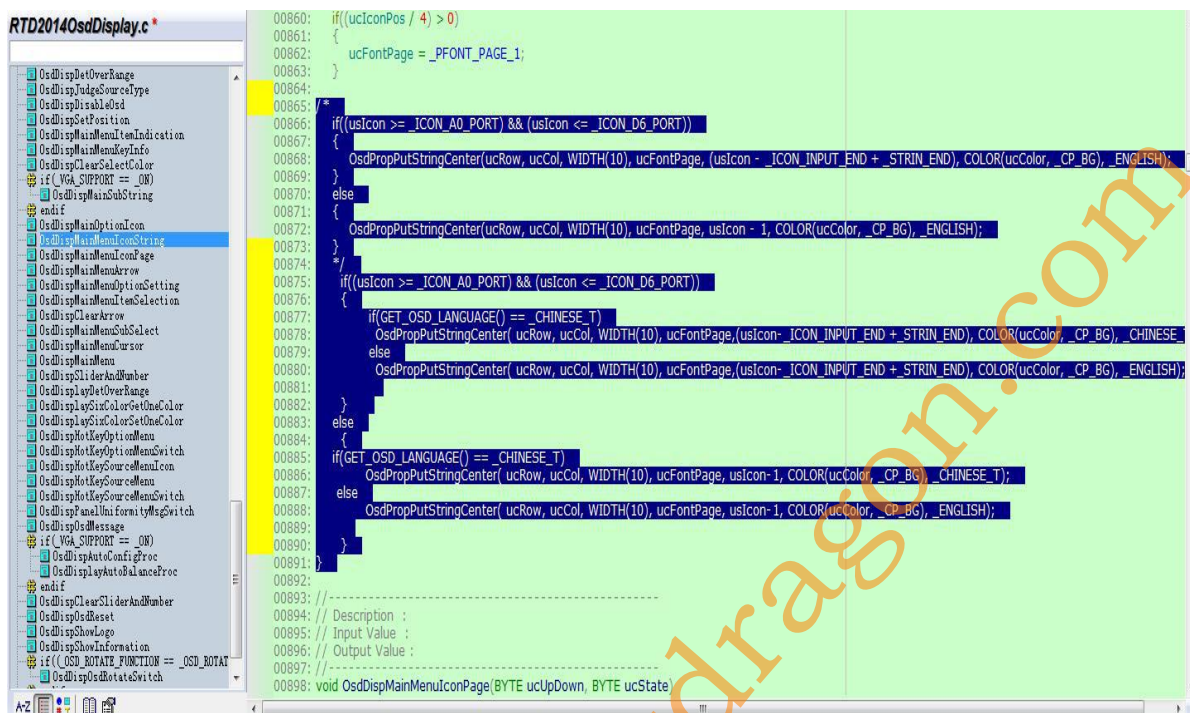
    //在函数末尾:
    if((usIcon >= _ICON_A0_PORT) && (usIcon <= _ICON_D6_PORT))
    {
        if(GET_OSD_LANGUAGE() == _CHINESE_T)
            OsdPropPutStringCenter(ucRow, ucCol, WIDTH(10), ucFontPage, (usIcon - _ICON_INPUT_END + _STRIN_END), COLOR(ucColor, _CP_BG), _CHINESE_T);
        else
            OsdPropPutStringCenter(ucRow, ucCol, WIDTH(10), ucFontPage, (usIcon - _ICON_INPUT_END + _STRIN_END), COLOR(ucColor, _CP_BG), _ENGLISH);
    }
    else
    {
        if(GET_OSD_LANGUAGE() == _CHINESE_T)
            OsdPropPutStringCenter(ucRow, ucCol, WIDTH(10), ucFontPage, usIcon - 1, COLOR(ucColor, _CP_BG), _CHINESE_T);
        else
            OsdPropPutStringCenter(ucRow, ucCol, WIDTH(10), ucFontPage, usIcon - 1, COLOR(ucColor, _CP_BG),

```



_ENGLISH);

```
}
}
```



20. 在 **MenuOtherLanguageAdjust** () 这个函数中还要添加在菜单中左右选择中英文，

void MenuOtherLanguageAdjust(void)

```
{
    switch(GET_KEYMESSAGE())
    {
        case _MENU_KEY_MESSAGE:
            //case _UP_KEY_MESSAGE:
                if(g_usBackupValue != GET_OSD_LANGUAGE())
                {
                    SET_OSD_EVENT_MESSAGE(_OSDEVENT_SAVE_NVRAM_OSDUSERDATA_MSG);
                }
                SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
                OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _UNSELECT, GET_OSD_LANGUAGE());
                // OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);

                break;

        case _RIGHT_KEY_MESSAGE:
        case _LEFT_KEY_MESSAGE:
            OsdDispSliderAndNumber(_MENU_OTHER_LANGUAGE_ADJUST, GET_OSD_LANGUAGE());
    }
}
```



```
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_CLEAR);
//OsdDispMainMenuIconPage(_UP, _ICON_PAGE_OTHER_0);//进入到选项就跳到第一页去了
OsdDispMainMenuIconPage(_UP, _ICON_PAGE_OTHER_1);
//OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_LANGUAGE);//自定义图标
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);

OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _SELECT, GET_OSD_LANGUAGE());
OsdDispMainMenuItemSelection((3), _OSD_PRE_SELECT);
//OsdDispMainMenuItemSelection((3), _OSD_SELECT);
//OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _UNSELECT, GET_OSD_LANGUAGE());

break;

/*
case _MENU_KEY_MESSAGE:
    if(g_usBackupValue != GET_OSD_LANGUAGE())
    {
        g_usAdjustValue = g_usBackupValue;
        SET_OSD_LANGUAGE(g_usAdjustValue);
    }
    SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
    OsdDispMainMenuItemSelection((3), _OSD_SELECT);
// OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
break;

*/

case _EXIT_KEY_MESSAGE:
    /*
        if(g_usBackupValue != GET_OSD_LANGUAGE())
        {
            g_usAdjustValue = g_usBackupValue;
            SET_OSD_LANGUAGE(g_usAdjustValue);
        }
        // OsdDispMainMenuItemSelection((3), _OSD_SELECT);
        SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
        OsdDispMainMenuCursor(GET_OSD_STATE(),
GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);//选中图标为橙黄色
        // OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _SELECT, GET_OSD_LANGUAGE());
        // OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OSD_PRE_SELECT);
    */
    ///////////
    if(g_usBackupValue != GET_OSD_LANGUAGE())
    {
        g_usAdjustValue = g_usBackupValue;
        SET_OSD_LANGUAGE(g_usAdjustValue);
    }
```



```
    }  
    SET_OSD_STATE(_MENU_OTHER_LANGUAGE);  
    OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);  
    OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _UNSELECT, GET_OSD_VOLUME_MUTE());  
        break;  
    default:  
        break;  
    }  
}
```

2) 在枚举中添加: **-RTD2014Include.h**

-2556 中

typedef enum

```
{  
    _OPTION_DP_LANE_COUNT_SELECT,  
    _OPTION_HDR_MODE_SELECT, // _ULTRA_HDR_SUPPORT  
    _OPTION_DARK_ENHANCE_ON_OFF, // _ULTRA_HDR_SUPPORT  
    _OPTION_LANGUAGE_SELECT,  
}EnumOptionIndex;
```

--2796 则是下面

typedef enum

{

末尾添加

```
    _OPTION_LANGUAGE_SELECT,  
}EnumOptiontype;
```

3) 在 language 图标的前面的右键添加:

void MenuOtherOsdVPos(void)

{

```
    switch(GET_KEYMESSAGE())  
    {
```

```
        case _RIGHT_KEY_MESSAGE:
```

```
/*--用注释这个，菜单就花了
```

```
    OsdDispClearSliderAndNumber();
```

```
    SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
```

```
    OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_LANGUAGE);
```

```
    OsdDispMainMenuItemSelection((3), _OSD_SELECT);
```

```
OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _UNSELECT, GET_OSD_LANGUAGE());
```

```
    */
```

```
    OsdDispClearSliderAndNumber();
```

```
    SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
```

```
    OsdDispMainMenuIconPage(_UP, _ICON_PAGE_OTHER_1);
```




```
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);
// OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
//OsdDispMainMenuItemSelection((3), _OSD_SELECT);
OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _UNSELECT, GET_OSD_LANGUAGE());
OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);

break;

....
}
```

4)void MenuOtherLanguage(void)中

```
void MenuOtherLanguage(void)
{
    switch(GET_KEYMESSAGE())
    {
        case _MENU_KEY_MESSAGE:
            g_usBackupValue = GET_OSD_LANGUAGE();
            SET_OSD_STATE(_MENU_OTHER_LANGUAGE_ADJUST);
            OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _SELECT, GET_OSD_LANGUAGE());
            OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _INSUBSET);
            break;
        case _RIGHT_KEY_MESSAGE:
            OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_CLEAR); //清一页
            OsdDispClearSelectColor(_DOWN); //

            SET_OSD_STATE(_MENU_OTHER_TRNAPARENCY);
            OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
            OsdDispSliderAndNumber(_MENU_OTHER_TRNAPARENCY, GET_OSD_TRANSPARENCY_STATUS());
            break;

        case _LEFT_KEY_MESSAGE:
            OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_CLEAR); //
            OsdDispClearSelectColor(_DOWN); //

            SET_OSD_STATE(_MENU_OTHER_OSD_VPOS);
            OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
            OsdDispMainMenuIconPage(_UP, _ICON_PAGE_OTHER_0);
            OsdDispSliderAndNumber(_MENU_OTHER_OSD_VPOS, GET_OSD_VPOS());
            break;
        case _EXIT_KEY_MESSAGE:
            OsdDispClearSliderAndNumber();
            SET_OSD_STATE(_MENU_OTHER);
            OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
            OsdDispClearSelectColor(_DOWN); //清除选中的图标颜色
    }
}
```



```

OsdDispMainMenuIconPage(_UP, _ICON_PAGE_MAIN_1); //可能是 MAIN_2
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_OTHER_0);
break;
default:
break;
}
}
5) language 的后一个图标的左键修改:
void MenuOtherTransparency(void)
{
case _LEFT_KEY_MESSAGE:
/*
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_CLEAR); //
OsdDispClearSelectColor(_DOWN); //
OsdDispClearSliderAndNumber();
SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);
OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
*/

OsdDispClearSliderAndNumber();
SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
OsdDispMainMenuIconPage(_UP, _ICON_PAGE_OTHER_1);
OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_language_0);
//OsdDispMainMenuItemSelection((3), _OSD_SELECT);
OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, _UNSELECT, GET_OSD_LANGUAGE());
OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
break;
}

```

6) void **OsdDispMainMenuOptionSetting**(BYTE ucItem, BYTE ucSelectState, BYTE ucOption)函数中添加:

```

void OsdDispMainMenuOptionSetting(BYTE ucItem, BYTE ucSelectState, BYTE ucOption)
{
. . . . .
case OPTION_LANGUAGE_SELECT:
    if(ucOption == _ENGLISH)
    {
        ucCol = COL(4) + ((_ITEM_0 % 4) * 10) - 2;
    }
    else if(ucOption == _CHINESE_T)
    {
        ucCol = COL(4) + ((_ITEM_1 % 4) * 10) - 2;
    }

    OsdWindowDrawingByFontHighlight(ucWin, ucRow, ucCol, WIDTH(10), HEIGHT(5), ucWinColor,
ucForegroundColor, _CP_BG);

```



break;

```
        default:
            break;
    }

    if(ucItem == _OPTION_COLOR_PCM_TYPE)
    {
#ifdef _OCC_SUPPORT == _ON
        OsdFuncChangeIconColor1Bit(ROW(14), _ITEM_6, WIDTH(g_ucOsdWidth), HEIGHT(4), _CP_BLUE);
#endif
    }
    else if(ucItem == _OPTION_DP_MST_TYPE)
    {
    }
    else if(ucItem == _OPTION_DISP_ROTATE_TYPE)
    {
    }
    else
    {
        OsdFuncChangeIconColor1Bit(ROW(14), _ITEM_4, WIDTH(g_ucOsdWidth), HEIGHT(4), _CP_BLUE);
    }
}
```

7) RTD2014Include.h 定义 **_OPTION_LANGUAGE_SELECT**

typedef enum

{

_OPTION_LANGUAGE_SELECT,

}EnumOptionIndex;

21. **OsdDispSliderAndNumber**(**_MENU_OTHER_LANGUAGE_ADJUST, GET_OSD_LANGUAGE()**);

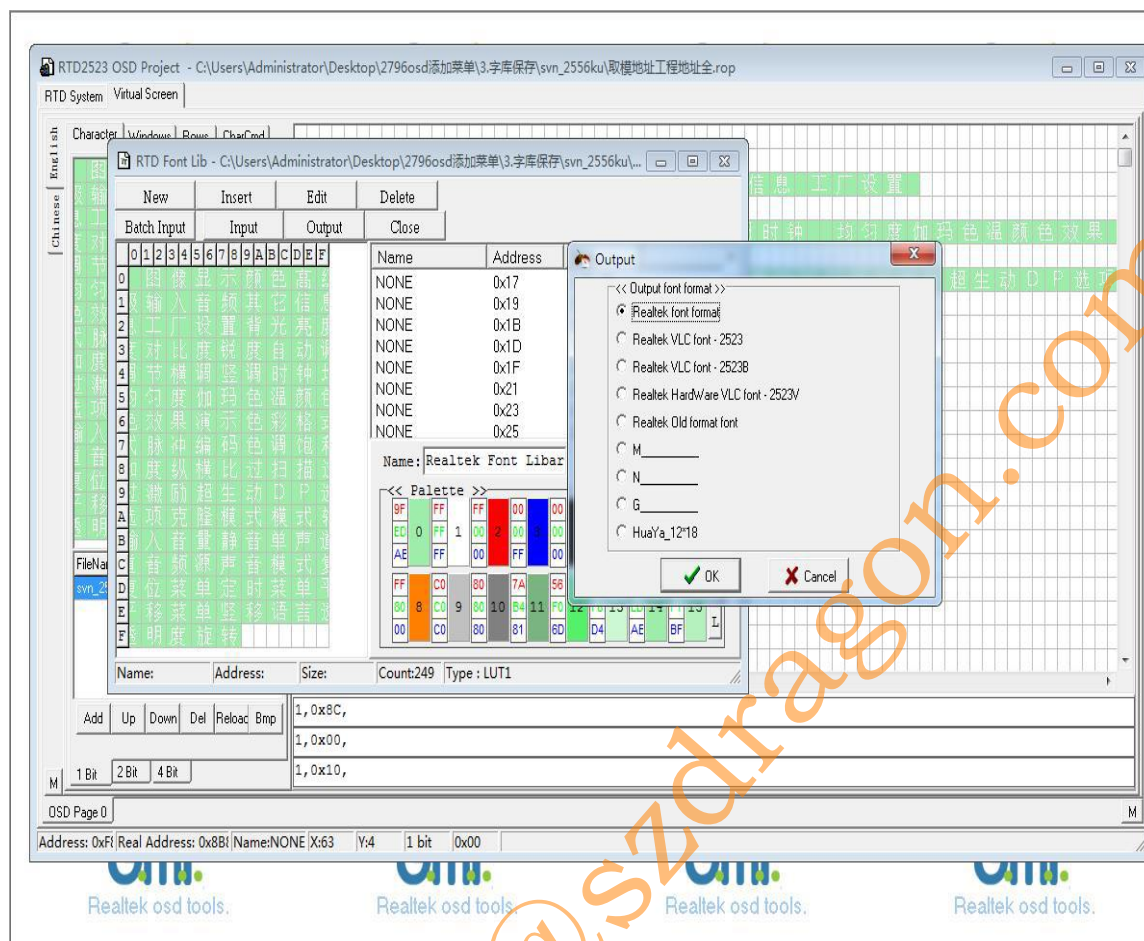
中 **_MENU_OTHER_LANGUAGE_ADJUST** 语言选择的条件, 这里举例的只有两种语言 **_ENGLISH** 和 **_CHINESE_T**:

```
case _MENU_OTHER_LANGUAGE_ADJUST:
    if((GET_KEYMESSAGE() == _LEFT_KEY_MESSAGE) || (GET_KEYMESSAGE() ==
_RIGHT_KEY_MESSAGE))
    {
        if(g_usAdjustValue == _ENGLISH)
        {
            g_usAdjustValue = _CHINESE_T;
        }
        else
        {
            g_usAdjustValue = _ENGLISH;
        }
SET_OSD_LANGUAGE(g_usAdjustValue);
    }
    break;
```

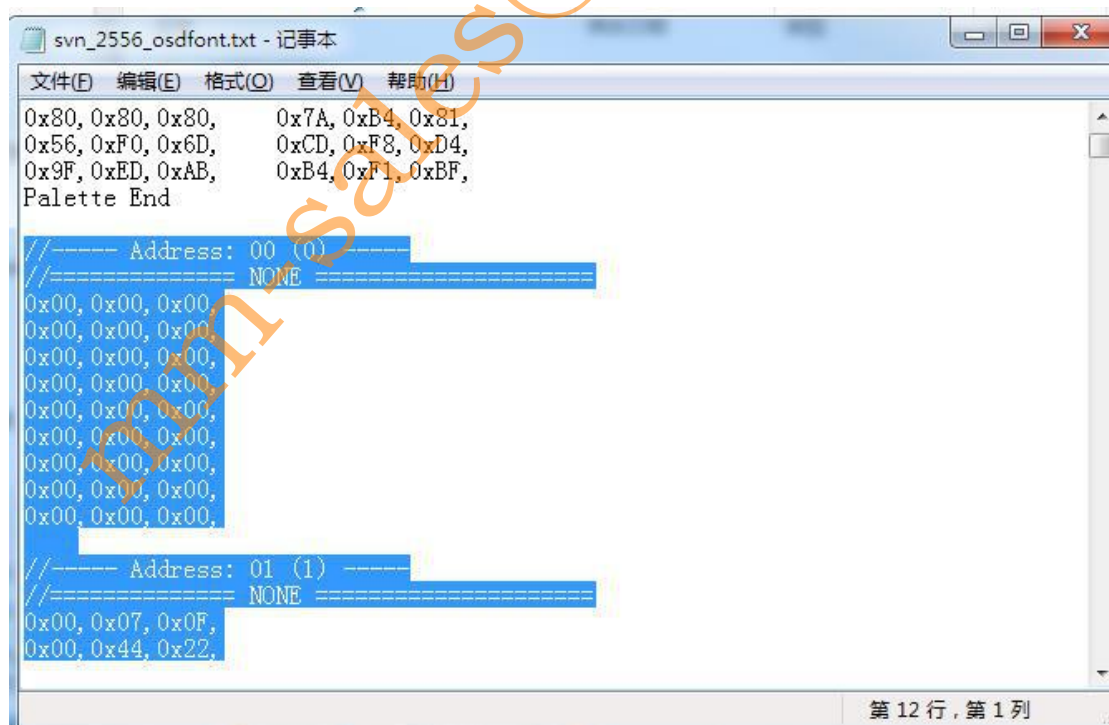


22.把所做的字库转化为数据

1) output->第一个 Realtek font format-》OK



2) 保存为 TXT 格式





3) 添加到文件 RTD2014OsdFontProp.c 文件前面

根据英文的数据仿照 BYTE code tFONT_EUROPE[];

BYTE code tFONT_CHINESE[] =

```
{
//----- Address: 00 (0) -----
//===== NONE =====
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
0x00,0x00,0x00,
//----- Address: 01 (1) -----
. . . . .
//----- Address: F7 (247) -----
//===== NONE =====
0x00,0x01,0x00,
0x00,0x11,0x00,
0x00,0x72,0xE0,
0x00,0x24,0x89,
0x00,0x70,0xE8,
0x00,0x00,0x8E,
0x00,0x72,0x88,
0x00,0x00,0x88,
0x00,0x00,0x80,

//===== NONE =====
0x01,0x00,0x00,
0x11,0x00,0x00,
0xF1,0xE0,0x00,
0x2F,0x0F,0x00,
0x24,0x00,0x00,
0xF0,0xE2,0x00,
0x42,0x48,0x00,
0x10,0x08,0x00,
0x00,0x80,0x00,
```

};

4)

////////////////////////////////////

//中文



```
BYTE code tSTRING_PICTURE_CHS[] =  
{  
    0x01,0x02,0x03,0x04, _END_,  
};
```

```
BYTE code tSTRING_DISPLAY_CHS[] =  
{  
    0x05,0x06,0x07,0x08, _END_,  
};
```

```
BYTE code tSTRING_COLOR_CHS[] =  
{  
    0x09,0x0A,0x0B,0x0C, _END_,  
};
```

.....

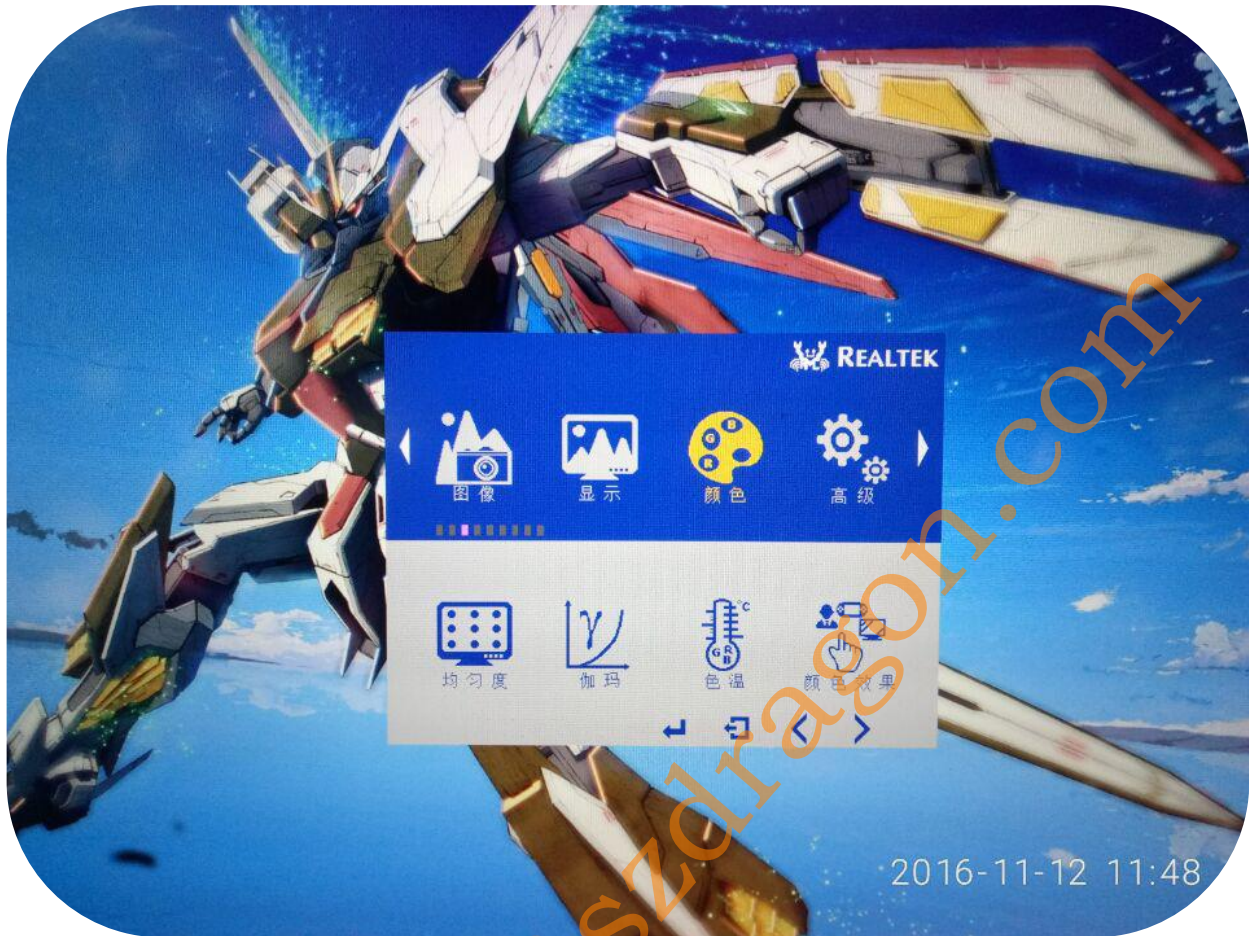
5)应用:

```
BYTE *OsdGetChnTFontPage(BYTE ucString)  
{  
    BYTE *pArray = 0;  
  
    switch(ucString)  
    {  
        case _STRING_PICTURE:  
        case _STRING_DISPLAY:  
        case _STRING_COLOR:  
        case _STRING_ADVANCE:  
        case _STRING_INPUT:  
        case _STRING_AUDIO:  
        case _STRING_OTHER:  
        case _STRING_INFORMATION:  
        case _STRING_FACTORY:  
  
        case _STRING_INFORMATION: //要翻译为中文的菜单图标都添加到上面  
            pArray = tFONT_CHINESE; //这个中文地址就是刚刚第二步生成的码值文档 1234，可参考英文  
            break;  
    }  
    return pArray;  
}
```



Dragon Source
龍源電子

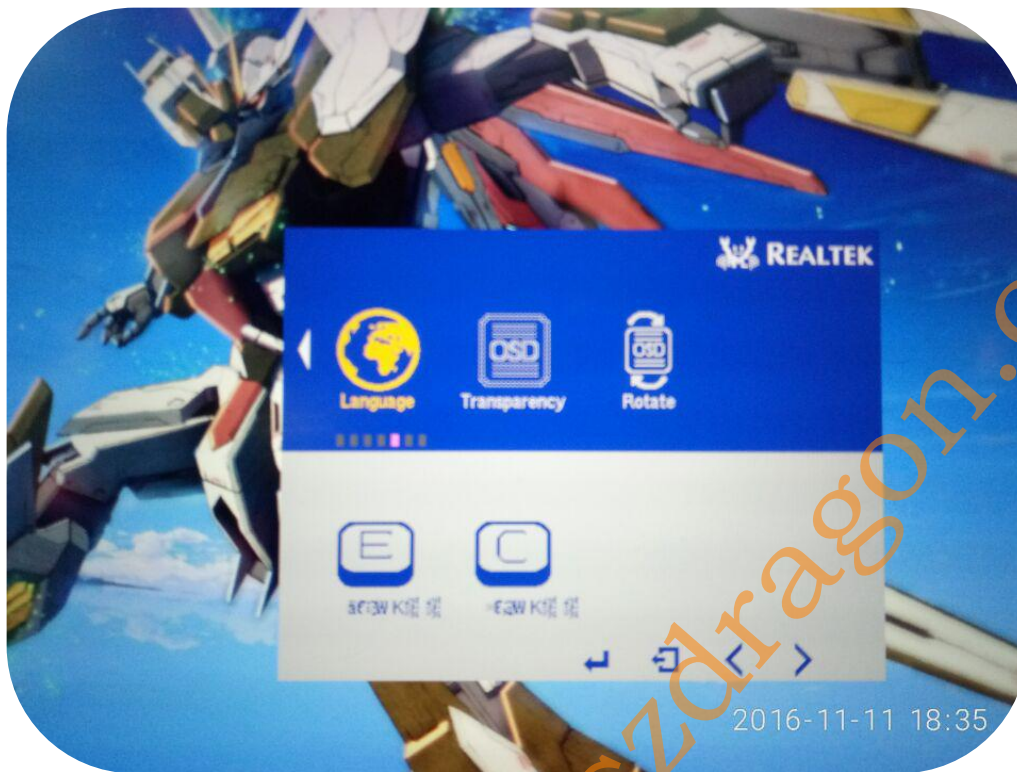
最后编译烧写成功图片：





存在的 bug:

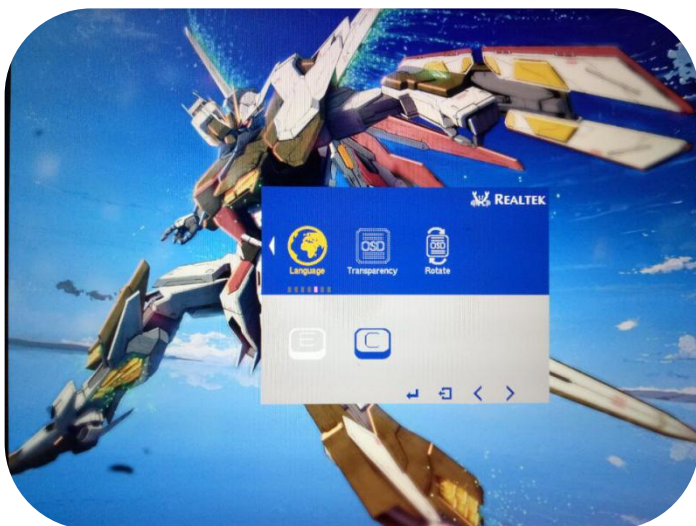
(1) bug1:中英文图标下面乱码



bug2:添加了退出键后按退出颜色状态出现异常。



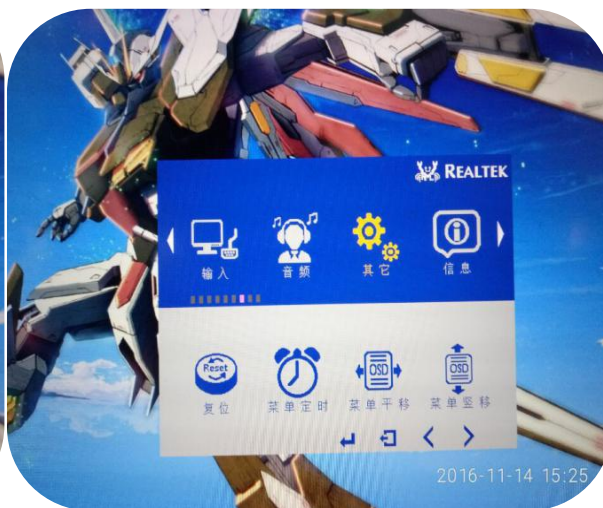
1)问题解决: -问题原因是用了写字符串的函数



RTD2014OsdDisplay.c->void OsdDispMainMenuIconPage(BYTE ucUpDown, BYTE ucState)中

```
case _ICON_PAGE_language_0:
//图标显示字符串的: OsdDispMainMenuIconString
///OsdDispMainMenuIconString((ucUpDown+0),_ICON_LANGUAGE_english,pucOsdItemColor[0]);
//OsdDispMainMenuIconString((ucUpDown + 1), _ICON_LANGUAGE_chinese,pucOsdItemColor[1]);
// OsdDispMainMenuIconString((ucUpDown + 2), _ICON_NONE,pucOsdItemColor[2]);
//OsdDispMainMenuIconString((ucUpDown + 3), _ICON_NONE, pucOsdItemColor[3]);
//图标不显示字符串: OsdDispMainOptionIcon
OsdDispMainOptionIcon((ucUpDown + 0), _ICON_LANGUAGE_english, pucOsdItemColor[0]);
OsdDispMainOptionIcon((ucUpDown + 1), _ICON_LANGUAGE_chinese, pucOsdItemColor[1]);
OsdDispMainOptionIcon((ucUpDown + 2), _ICON_NONE, pucOsdItemColor[2]);
OsdDispMainOptionIcon((ucUpDown + 3), _ICON_NONE, pucOsdItemColor[3]);
```

2)问题 2 解决: 没有退出键, 加入后发现退出上下菜单都是橙黄色, 正常的状态是上层是橙黄色, 下层是白色。-问题原因是按第二次退出键时没有清除选中的颜色状态:





```
(1) void MenuOtherLanguage(void)
{
    case _EXIT_KEY_MESSAGE:
        OsdDispClearSliderAndNumber();
        SET_OSD_STATE(_MENU_OTHER);
        OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
        OsdDispClearSelectColor(_DOWN);
        OsdDispMainMenuIconPage(_UP, _ICON_PAGE_MAIN_1);
        OsdDispMainMenuIconPage(_DOWN, _ICON_PAGE_OTHER_0);

        break;

(2) void MenuOtherLanguageAdjust(void)
{
    switch(GET_KEYMESSAGE())
    {
        case _EXIT_KEY_MESSAGE:
            /*//注释部分则会出现退出上下两个图标都是橙黄色
            if(g_usBackupValue != GET_OSD_LANGUAGE())
            {
                g_usAdjustValue = g_usBackupValue;
                SET_OSD_LANGUAGE(g_usAdjustValue);
            }
            // OsdDispMainMenuItemSelection((3), _OSD_SELECT);
            SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
            OsdDispMainMenuCursor(GET_OSD_STATE(),
            GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);//选中图标为橙黄色
            // OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, SELECT, GET_OSD_LANGUAGE());
            // OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OSD_PRE_SELECT);
            */
            if(g_usBackupValue != GET_OSD_LANGUAGE())
            {
                g_usAdjustValue = g_usBackupValue;
                SET_OSD_LANGUAGE(g_usAdjustValue);
            }

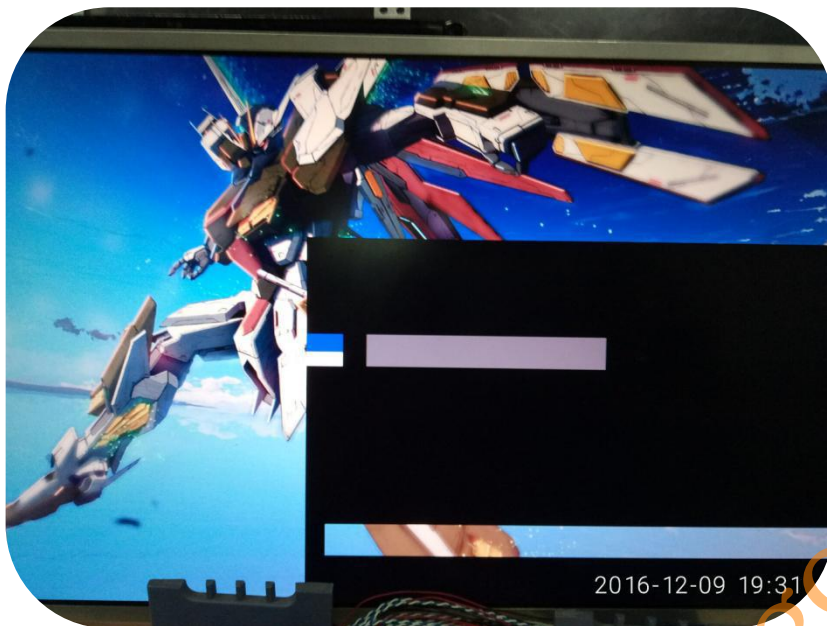
            SET_OSD_STATE(_MENU_OTHER_LANGUAGE);
            OsdDispMainMenuCursor(GET_OSD_STATE(), GET_OSD_STATE_PREVIOUS(), _OUTSUBSET);
            OsdDispMainMenuOptionSetting(_OPTION_LANGUAGE_SELECT, UNSELECT, GET_OSD_VOLUME_MUTE());

            break;
    }
}
```



20161209 问题三:

2796 添加中文问题:

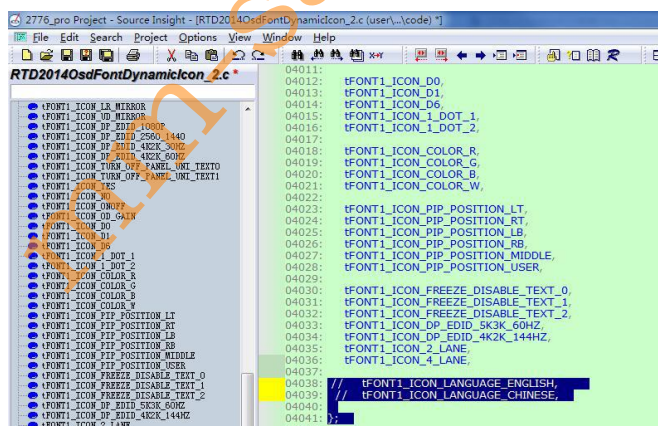


解决:

1) 下面限定了图标的范围

```
void OsdFontLoadNoticelcon(BYTE uclconPos, WORD uslcon)
{
    if((uslcon >= _ICON_NO_SUPPORT) && (uslcon <= _ICON_NO_SIGNAL_TEXT_1))
    {
        uslcon = uslcon - _ICON_ON;
        ScalerOsdHardwareVLC(tFONT1_MAIN_ICON_TABLE_2[uslcon],
        VLC_TABLE_SIZE(tFONT1_MAIN_ICON_TABLE_2[uslcon]),GET_CURRENT_BANK_NUMBER(),
        (_1BIT_NOTICE_ICON_START + (uclconPos * 24)), g_usFontTableStart, ucOsdRotateStatus);
    }
}
```

所以不能将新增的图标数组数据添加到末尾,



必须放在 code BYTE *tFONT1_MAIN_ICON_TABLE_2[] 范围之间:



if((uslcon >= _ICON_NO_SUPPORT) && (uslcon <= _ICON_NO_SIGNAL_TEXT_1))

```
RTD2014OsdFontDynamicIcon_2.c
03936: tFONT1_ICON_COLOR_TEMP_USER_B,
03937:
03938: tFONT1_ICON_COLOR_PCM_SRGB,
03939: tFONT1_ICON_COLOR_PCM_ADOBE_RGB,
03940: tFONT1_ICON_COLOR_PCM_USER,
03941: tFONT1_ICON_COLOR_PCM_NACTIVE,
03942: tFONT1_ICON_COLOR_PCM_SOFT_PROFT,
03943:
03944: //Color Effect User Adjust
03945: tFONT1_ICON_COLOR_EFF_USER_Y,
03946: tFONT1_ICON_COLOR_EFF_USER_C,
03947: tFONT1_ICON_COLOR_EFF_USER_M,
03948: tFONT1_ICON_COLOR_EFF_USER_HUE,
03949: tFONT1_ICON_COLOR_EFF_USER_SAT,
03950:
03951: //Osd Message
03952: tFONT1_ICON_NO_SUPPORT,
03953: tFONT1_ICON_NO_SUPPORT_TEXT_0,
03954: tFONT1_ICON_NO_SUPPORT_TEXT_1,
03955:
03956: tFONT1_ICON_NO_CABLE,
03957: tFONT1_ICON_NO_CABLE_TEXT_0,
03958: tFONT1_ICON_NO_CABLE_TEXT_1,
03959:
03960:
03961: tFONT1_ICON_LANGUAGE_ENGLISH,
03962: tFONT1_ICON_LANGUAGE_CHINESE,
03963:
03964:
03965: tFONT1_ICON_NO_SIGNAL,
03966: tFONT1_ICON_NO_SIGNAL_TEXT_0,
03967: tFONT1_ICON_NO_SIGNAL_TEXT_1,
03968:
03969:
```

另外图标的存放也要和数据数组位置一致:

```
_ICON_INFORMATION_HINT,
// _ICON_LANGUAGE_ENGLISH,
// _ICON_LANGUAGE_CHINESE,
_OPTION_ICON_END,
}EnumIconIndex;
```

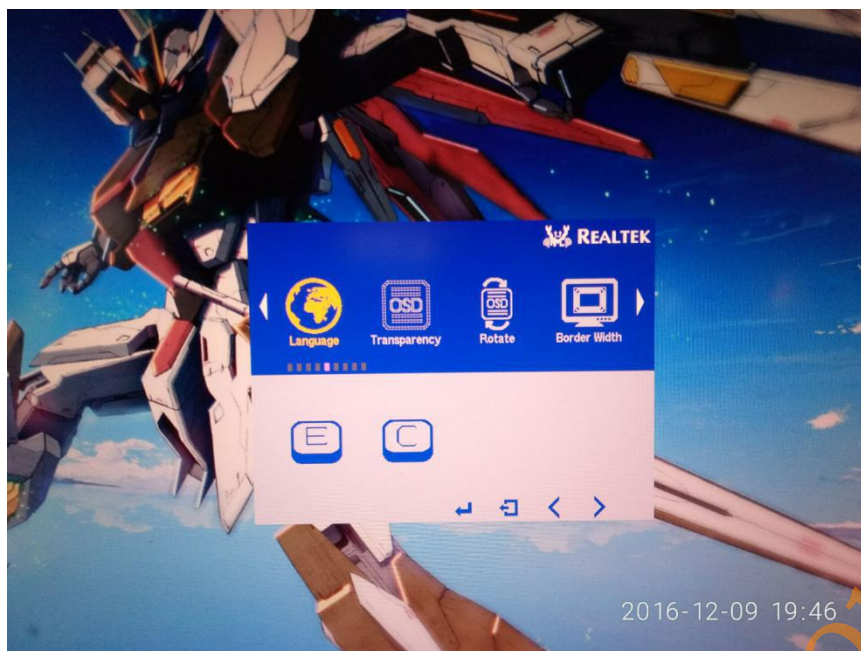
```
RTD2014Include.h
02730:
02731: _ICON_TEMP_USER_R,
02732: _ICON_TEMP_USER_G,
02733: _ICON_TEMP_USER_B,
02734:
02735: _ICON_COLOR_PCM_SRGB,
02736: _ICON_COLOR_PCM_ADOBE_RGB,
02737: _ICON_COLOR_PCM_USER,
02738: _ICON_COLOR_PCM_NACTIVE,
02739: _ICON_COLOR_PCM_SOFT_PROFT,
02740:
02741: _ICON_COLOR_EFF_USER_Y,
02742: _ICON_COLOR_EFF_USER_C,
02743: _ICON_COLOR_EFF_USER_M,
02744: _ICON_COLOR_EFF_USER_HUE,
02745: _ICON_COLOR_EFF_USER_SAT,
02746:
02747: _ICON_NO_SUPPORT,
02748: _ICON_NO_SUPPORT_TEXT_0,
02749: _ICON_NO_SUPPORT_TEXT_1,
02750:
02751: _ICON_NO_CABLE,
02752: _ICON_NO_CABLE_TEXT_0,
02753: _ICON_NO_CABLE_TEXT_1,
02754:
02755: _ICON_LANGUAGE_ENGLISH,
02756: _ICON_LANGUAGE_CHINESE,
02757:
02758: _ICON_NO_SIGNAL,
02759: _ICON_NO_SIGNAL_TEXT_0,
02760: _ICON_NO_SIGNAL_TEXT_1,
02761:
02762: _ICON_POWER_SAVING,
02763: _ICON_POWER_SAVING_TEXT_0,
02764: _ICON_POWER_SAVING_TEXT_1,
```

修改正确后效果



Dragon Source

龍源電子



最后添加中文字库成功效果:

