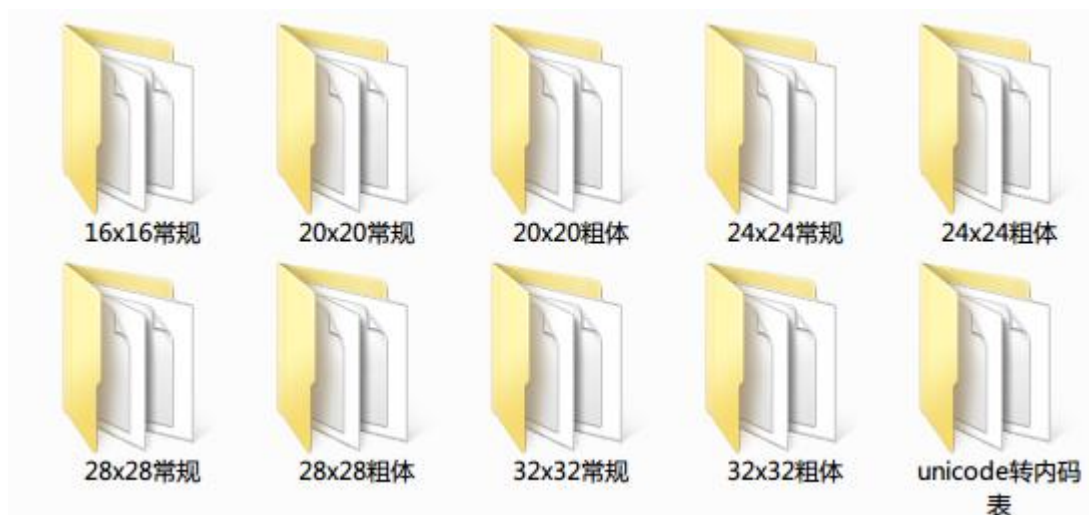


# AC54xx 字库使用说明

## 1. 添加字库

请先确定字号以及字形，再根据所选的多国语言选择相应的字库文件以及 TAB 文件拷贝到资源目录 (res 文件夹)。(注：若只需要显示内码，\*.TAB 文件可不拷贝)



以下为各国语言所对应的字库文件以及 Unicode 转内码表

Chinese\_Simplified(简体中文)：

F\_ASCII.PIX F\_GB2312.PIX F\_GB2312.TAB (GB2312 字库)

F\_ASCII.PIX F\_GBK.PIX F\_GBK.TAB (GBK 字库)

Chinese\_Traditional(繁体中文)：

F\_ASCII.PIX F\_BIG5.PIX F\_BIG5.TAB

Japanese(日语)：

F\_SJISASC.PIX F\_SJIS.PIX F\_SJIS.TAB

Korean(韩语)：

F\_ASCII.PIX F\_KSC.PIX F\_KSC.TAB

English(英语)：

F\_CP1252.PIX F\_CP1252.TAB

French(法语)：

F\_CP1252.PIX F\_CP1252.TAB

German(德语)：

F\_CP1252.PIX F\_CP1252.TAB

Italian(意大利语)：

F\_CP1252.PIX F\_CP1252.TAB

Dutch(荷兰语)：

F\_CP1252.PIX F\_CP1252.TAB

Portuguese(葡萄牙语)：

F\_CP1252.PIX F\_CP1252.TAB

Spanish(西班牙语):  
F\_CP1252.PIX F\_CP1252.TAB  
Swedish(瑞典语):  
F\_CP1252.PIX F\_CP1252.TAB  
Czech(捷克语):  
F\_CP1250.PIX F\_CP1250.TAB  
Danish(丹麦语):  
F\_CP1252.PIX F\_CP1252.TAB  
Polish(波兰语):  
F\_CP1250.PIX F\_CP1250.TAB  
Russian(俄罗斯语):  
F\_CP1251.PIX F\_CP1251.TAB  
Turkey(土耳其语):  
F\_CP1254.PIX F\_CP1254.TAB  
Hebrew(希伯来语):  
F\_CP1255.PIX F\_CP1255.TAB  
Thai(泰语):  
F\_CP874.PIX F\_CP874.TAB  
Hungarian(匈牙利语):  
F\_CP1250.PIX F\_CP1250.TAB  
Romanian(罗马尼亚语):  
F\_CP1250.PIX F\_CP1250.TAB  
Arabic(阿拉伯语):  
F\_CP1256.PIX F\_CP1256.TAB

## 2.创建文本控件

编码格式填“text”



### 3.初始化字库

修改 fontinit.c 文件，配置需要的语言

注:简体中文包含 GB2312 字库以及 GBK 字库,GB2312 是 GBK 的精简版,适用于 spiflash 空间比较小的应用.

```
/* 语言字库配置 */
#define LANGUAGE BIT(Chinese_Simplified)|\
                BIT(Japanese)|\
                BIT(English)
```

选择 GB2312 字库时的初始化

```
#if LANGUAGE&BIT(Chinese_Simplified)
REGISTER_LANGUAGE_BEGIN(Chinese_Simplified) = {
    .language_id = Chinese_Simplified,
    .flags = FONT_SHOW_PIXEL | FONT_SHOW_MULTI_LINE,
    .pixel.file.name = (char *)"mnt/spiflash/res/F_GB2312.PIX",
    .ascpixel.file.name = (char *)"mnt/spiflash/res/F_ASCII.PIX",
    .tabfile.name = (char *)"mnt/spiflash/res/F_GB2312.TAB",
    .isgb2312 = true,
    .bigendian = false,
    .putchar = platform_putchar,

    /* .language_id = Chinese_Simplified, */
    /* .flags = FONT_SHOW_PIXEL, */
    /* .pixel.file.name = (char *)"mnt/spiflash/res/F_GBK.PIX", */
    /* .ascpixel.file.name = (char *)"mnt/spiflash/res/F_ASCII.PIX", */
    /* .tabfile.name = (char *)"mnt/spiflash/res/F_GBK.TAB", */
    /* .isgb2312 = false, */
    /* .putchar = platform_putchar, */
};
#endif
```

选择 GBK 字库时的初始化

```
#if LANGUAGE&BIT(Chinese_Simplified)
REGISTER_LANGUAGE_BEGIN(Chinese_Simplified) = {
    /* .language_id = Chinese_Simplified, */
    /* .flags = FONT_SHOW_PIXEL | FONT_SHOW_MULTI_LINE, */
    /* .pixel.file.name = (char *)"mnt/spiflash/res/F_GB2312.PIX", */
    /* .ascpixel.file.name = (char *)"mnt/spiflash/res/F_ASCII.PIX", */
    /* .tabfile.name = (char *)"mnt/spiflash/res/F_GB2312.TAB", */
    /* .isgb2312 = true, */
    /* .bigendian = false, */
    /* .putchar = platform_putchar, */

    .language_id = Chinese_Simplified,
    .flags = FONT_SHOW_PIXEL,
    .pixel.file.name = (char *)"mnt/spiflash/res/F_GBK.PIX",
    .ascpixel.file.name = (char *)"mnt/spiflash/res/F_ASCII.PIX",
    .tabfile.name = (char *)"mnt/spiflash/res/F_GBK.TAB",
    .isgb2312 = false,
    .putchar = platform_putchar,
};
#endif
```

## 4.添加文本控件回调函数

```
static int text_txt_test_onchange(void *ctr, enum
element_change_event e, void *arg)
{
    struct draw_context *dc = (struct draw_context *)arg;
    struct ui_text *text = (struct ui_text *)ctr;
    const char *ascii = "0123abcd\n";
    /*
    注:此处字符串 ascii 不能直接输入中文, 中文的编码由本文件的编码格式决定, 如需
    显示中文, 请使用数组的形式.
    */

    switch (e) {
    case ON_CHANGE_INIT:
        /*此处添加控件默认显示文字, 若无显示默认文字的需求可不加*/

        ui_text_set_text_attrs(text, ascii, strlen(ascii), FONT_ENCODE
_UTF8, 0, FONT_DEFAULT);

        break;
    case ON_CHANGE_SHOW:
        break;
    default:
        return false;
    }

    return false;
}

REGISTER_UI_EVENT_HANDLER(TEXT_TXT_TEST)
.onchange = text_txt_test_onchange,
.onkey = NULL,
.ontouch = NULL,
};
```

## 5.字库接口

```
/*
函数功能：设置文本的属性,用于文本控件的初始化
输入参数：
Text    : ui_text 结构体
str      : 编码
Strlen  : 编码长度
Encode  : 编码类型
Endian  : 大小端 (只针对 FONT_ENCODE_UNICODE)
flags   : 标志位
        FONT_DEFAULT           //默认显示（单行显示）
        FONT_SHOW_PIXEL       //显示字符
        FONT_SHOW_MULTI_LINE  //多行显示
返回值：无
*/
void ui_text_set_text_attrs(struct ui_text *text, const char *str, int
strlen, u8 encode, u8 endian, u32 flags);

/*
函数功能：内码显示函数
输入参数：
id       : 文本控件 ID 号
str      : 内码
Strlen   : 内码长度
flags    : 标志位
        FONT_DEFAULT           //默认显示（单行显示）
        FONT_SHOW_PIXEL       //显示字符
        FONT_SHOW_MULTI_LINE  //多行显示
返回值：
        0: 失败
        其他: 已显示的编码长度
*/
int ui_text_set_text_by_id(int id, const char *str, int strlen, u32
flags);

/*
函数功能：unicode 编码显示函数
输入参数：
id       : 文本控件 ID 号
str      : unicode 编码
strlen   : unicode 编码长度
endian   : 大小端
```

```

        FONT_ENDIAN_BIG          //大端
        FONT_ENDIAN_SMALL        //小端
flags :标志位
        FONT_DEFAULT             //默认显示（单行显示）
        FONT_SHOW_PIXEL          //显示字符
        FONT_SHOW_MULTI_LINE     //多行显示
返回值：
    0:失败
    其他:已显示的编码长度
*/
int ui_text_set_textw_by_id(int id, const char *str, int strlen, int
endian, u32 flags);
/*
函数功能: utf8 编码显示函数
输入参数:
id      :文本控件 ID 号
str     :utf8 编码
strlen  :utf8 编码长度
flags   :标志位
        FONT_DEFAULT             //默认显示（单行显示）
        FONT_SHOW_PIXEL          //显示字符
        FONT_SHOW_MULTI_LINE     //多行显示
返回值：
    0:失败
    其他:已显示的编码长度
*/

int ui_text_set_textu_by_id(int id, const char *str, int strlen, u32
flags);

```

示例:

```

/*内码,直接显示字符串*/
ui_text_set_text_by_id(TEXT_TXT_TEST, "abc", -1, FONT_DEFAULT);
/*内码,单行显示*/
ui_text_set_text_by_id(TEXT_TXT_TEST, ansicode, sizeof(ansicode),
FONT_DEFAULT);
/*utf16 大端编码, 多行显示*/
ui_text_set_textw_by_id(TEXT_TXT_TEST, utf16_be, sizeof(utf16_be),
FONT_ENDIAN_BIG, FONT_DEFAULT | FONT_SHOW_MULTI_LINE);
/*utf16 小端编码,单行显示*/
ui_text_set_textw_by_id(TEXT_TXT_TEST, utf16_le, sizeof(utf16_le),
FONT_ENDIAN_SMALL, FONT_DEFAULT);
/*utf8 编码,多行显示*/

```

```
ui_text_set_textu_by_id(TEXT_TXT_TEST,text_utf8,sizeof(text_utf8),  
FONT_DEFAULT|FONT_SHOW_MULTI_LINE);
```