

PHY62XX

多语言字库介绍

Version 1.0

Phyplus Microelectronics Limited

2019/01/01

版本控制信息

版本/状态	作者	参与者	起止日期	备注
V1.0	Eagle		01/01/2019	文档初稿

目录

1	简介.....	1
2	PHY62XX Font API.....	2
2.1	font_ui_font_load.....	2
2.2	utf8_to_unicode	2
2.3	ui_font_unicode.....	3
3	应用例程.....	4
4	定制要求.....	6

1 简介

PHY62XX Font library 提供多语言，可配置的字库方案。

字库方案包括两个部分：

- PHY62XX Font library + API
 - 字库驱动库和 API 函数，提供字库的注册，UTF-8 解析和字库 bitmap 接口
 - 字库 bitmap 接口支持可变宽度文字。
- PHY62XX 多语言字库
 - 根据客户需要生成多语言字库，在 Unicode 基本多文种平面（ BMP 0x0000~0xffff ）可以根据需要配置任意语种组合，并且可以根据客户需要配置文字分辨率，扫描方式。

2 PHY62XX Font API

2.1 font_ui_font_load

加载字库，将已烧录在内部 flash 的字库加载，加载之后会获得一个字库句柄，可以通过该句柄调用函数 ui_font_unicode() 获取 Unicode 字符的 bitmap。

- 函数原型

```
void* ui_font_load(uint32_t flash_addr);
```

- 参数

类型	参数名	说明
uint32_t	flash_addr	字库数据存储的 flash 地址

- 返回值

返回 void* 类型字库句柄，如果数值为 NULL 表示加载失败。

2.2 utf8_to_unicode

将 UTF-8 字符串转换为 Unicode 字符。

- 函数原型

```
int utf8_to_unicode(const char* utf8, uint16_t *unicode);
```

- 参数

类型	参数名	说明
const char*	utf8	输入参数，输入的 UTF-8 字符串，以 '\0' 结束。
uint16_t *	unicode	输出参数，转换 UTF-8 得到的一个 Unicode 字符。

- 返回值

返回 int 类型。

0: 表示字符串解析完成，输出参数为无效数据。

1~n: 表示转换获得一个 Unicode 字符，对应的 UTF-8 字符串解析消耗 n 个字节。

2.3 ui_font_unicode

将 UTF-8 字符串转换为 Unicode 字符。

- 函数原型

```
int ui_font_unicode(void* font, uint16_t unicode, uint8_t *bitmap);
```

- 参数

类型	参数名	说明
void*	font	字库句柄。
uint16_t	unicode	Unicode 字符，需要转换为 bitmap
uint8_t *	bitmap	<p>Unicode 字符的 bitmap.</p> <p>其中字节 0~3 为分辨率信息，字节 4 及其以后数据为 bitmap 数据。</p> <p>字节 0~3 具体内容：</p> <p>Byte0: bitmap 数据宽度</p> <p>Byte1: bitmap 数据高度</p> <p>Byte2: bitmap 实际有效数据宽度</p> <p>Byte4: reserved。</p>

- 返回值

返回 int 类型。

0: 操作成功。

其他数值: 错误码。

3 应用例程

- 字库加载

通常我们在显示初始化的代码段加载字库驱动：

```
void disp_init(void)
{
    ui_disp_t* pdisp = &s_disp;
    disp_rect_t rect;

    lcd_init();
    memset(pdisp, 0, sizeof(ui_disp_t));
    pdisp->status = UI_DISP_OFF;
    pdisp->font = ui_font_load(0x11100000);
    DISP_RECT_FULL(rect);
    disp_clrscn(rect);
    disp_reflash_all();
}
```

- UTF-8 解析以及 Unicode bitmap 获取

```
disp_rect_t disp_string(disp_point_t point, const char *utf8)
{
    int ret;
    const char* utf8_str = utf8;
    uint16_t unic_ch;
    uint8_t bitmap[FONT_BITMAP_SIZE];
    uint16_t offset;
    ui_disp_t* pdisp = &s_disp;

    disp_point_t point_new = point;
    disp_rect_t rect;
    disp_rect_t rect1;
    DISP_RECT_EMPTY(rect);
    DISP_RECT_EMPTY(rect1);
    while(1)
    {
        offset = utf8_to_unicode(utf8_str, &unic_ch);
```

```
        if(offset == 0)
            break;
        utf8_str += offset;
        ret = ui_font_unicode(pdsp->font, unic_ch,bitmap);
        if(ret != APP_SUCCESS)
            continue;
        rect1 = disp_img_x(point_new, bitmap);
        point_new.x += rect1.w;
    }
    rect.x = point.x;
    rect.y = point.y;
    rect.w = point_new.x + 1 - point.x;
    rect.h = rect1.h + rect1.y - rect.y;
    return rect;
}
```


4 定制要求

客户如需定制字库，需要提供以下信息。

参数	说明
字体长宽高	
扫描方式	Bitmap 扫描方式，自左向右，或者自右向左。
中文常用字	中文字库是否使用 3000 个常用中文（可选）
段落列表	需要的语言种类在基本多语种平面的段落列表。