

Secret Status: Top Secret ( )    Secret ( )    Restrict ( ☒ )    Public ( )

# RKNanoD MP3 SDK 字库格式说明 (Version 1.0)

**RKNano Team**

**Rockchip Electronics Co., Ltd**

**www.rock-chips.com**

<b>Status:</b>  [ ] development  [ <input checked="" type="checkbox"/> ] public  [ ] modify	<b>File Tag:</b>	Font Format User Manual	
	<b>Version:</b>	1.0	
	<b>Author:</b>	Ma Longchang	
	<b>Data:</b>	2016 年 6 月 6 日	
	<b>Verify:</b>		

### Version History

Version	Author	Date	Remarks
1.0	Ma Longchang	2016-06-6	Original

## 目录

1	简介.....	4
2	字库文件结构说明.....	5
2.1	Font12.bin.....	5
2.1.1	文件结构.....	5
2.1.2	字形结构.....	6
2.2	Font16.bin 文件结构.....	8
2.2.1	文件结构.....	8
2.2.2	字形结构.....	9
3	用户自定义字库.....	11

# 1 简介

RKNanoD MP SDK 目前支持  $12 \times 12$ ,  $16 \times 16$  点阵大小的 UNICODE 字符的显示。本文档将对字库文件中的组成结构，字符在字库文件中的数据组成结构做以详细说明，并结合 SDK 中的显示代码作为辅助说明，以使用户开发过程中的使用。

## 2 字库文件结构说明

对 EMMC flash 的 SDK 而已，字库文件目录放在如下路径中：..\SDK\_160\_128\Development\firmware\_generate\_eMMC。

字库文件分别命名为：

12×12 dot:Font12.bin。

16×16 dot:Font16.bin。

### 2.1 Font12.bin

#### 2.1.1 文件结构

在该文件中包含了所有 Unicode 字符的字形数据，包括可显示与不可显示的，即从 0x0000 ~0xFFFF，每个 Unicode 字符的字形数据占用 32 个 bytes，其中有效数据为 26 个 bytes，其他剩余 6 个字节为 0，为无效字节，起补位作用，方便在文件中定位。

使用 UltraEdit 或 WinHex 二进制工具打开该字库文件，可以查看该文件的组成结构：如下图所示：

Font12.bin x	Font16.bin x
0 1 2 3 4 5 6 7 8 9 a b c d e f	
00000000h: 00 00 00 00 7F 80 40 80 40 80 40 80 40 80 40 80	.... €@€@€@€@€@€@
00000010h: 40 80 7F 80 00 00 00 00 8A 00 00 00 00 00 00 00	@€ €....?.....
00000020h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000030h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....
00000040h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000050h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....
00000060h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000070h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....
00000080h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
00000090h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....
000000a0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000b0h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....
000000c0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000d0h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....
000000e0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	.....
000000f0h: 00 00 00 00 00 00 00 00 80 00 00 00 00 00 00 00	..... €.....

Font12.bin 文件结构

其中 26 个 bytes 有效数据中前 24 个字节为 Unicode 字符的字形数据，第 25,26 个字节为字符的类型，最重要的是第 25 个字节，该类型包含了字符的字形种类及宽度信息，第 26 个

字节默认为 0。

具体描述如下表：

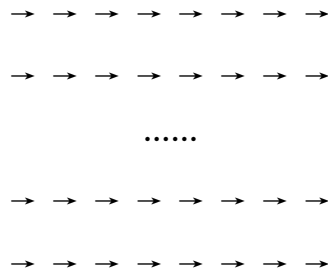
iType（第 25 个 bytes）	备注（单位：pixels）
0x10	仅当语言为俄语时，字符宽度为 8
0x20	字符宽度为 12
0x80	字符 Unicode 大于 0x7F 时，字符宽度为 12
0x80 – 0xFF	字符宽度为（iType – 0x80）

iType 类型含义

代码详见 LcdChar.c 中 LCD\_NFDispUnicodeChar 接口。

## 2.1.2 字形结构

每个 Unicode 字符的字形数据是按照从左→右，从上↓下的扫描顺序获取出来保存的，每行 12 个 bit（每个 pixel 占一个 bit），用 2 个字节表示，有 12 行，共 24 个字节的数据。



以大写的字符“W”举例说明：

“W”字符在 font12.bin 中的位置： $0x57 \times 0x20 (0x57 \ll 5) = 0xAE0$

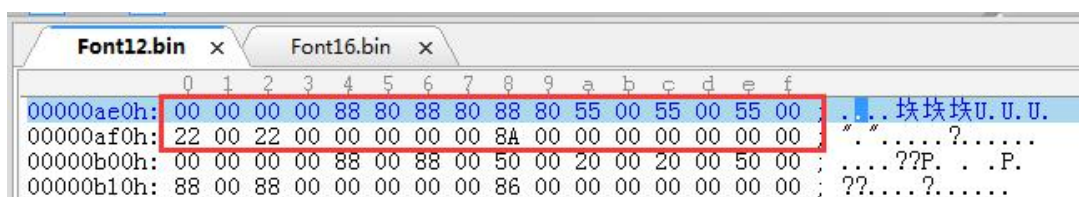
其字形数据为：从左到右，从上到下。

十六进制表示	二进制表示
00 00:	0 0 0 0 0 0 0 0 0 0 0 0
00 00:	0 0 0 0 0 0 0 0 0 0 0 0
88 80:	1 0 0 0 1 0 0 0 1 0 0 0
88 80:	1 0 0 0 1 0 0 0 1 0 0 0
88 80:	1 0 0 0 1 0 0 0 1 0 0 0

```

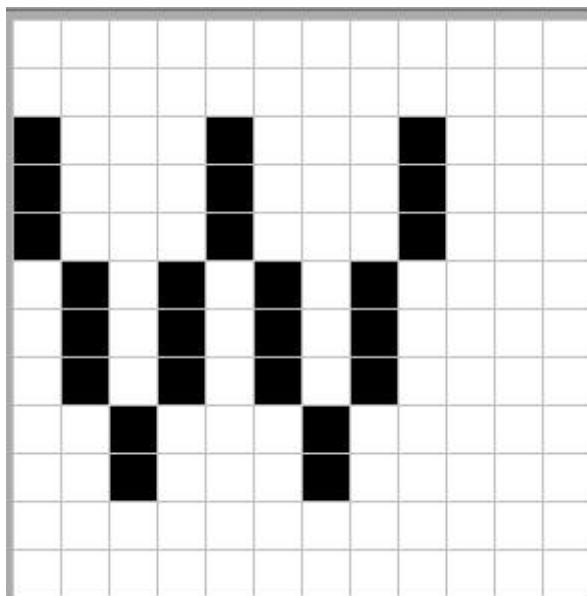
55 00:      0 1 0 1 0 1 0 1 0 0 0 0
55 00:      0 1 0 1 0 1 0 1 0 0 0 0
55 00:      0 1 0 1 0 1 0 1 0 0 0 0
22 00:      0 0 1 0 0 0 1 0 0 0 0 0
22 00:      0 0 1 0 0 0 1 0 0 0 0 0
00 00:      0 0 0 0 0 0 0 0 0 0 0 0
00 00:      0 0 0 0 0 0 0 0 0 0 0 0

```



“W” 在 Font12.bin 中的位置与内容

“W” 的显示效果图:



“W” 字符显示，宽度为 10 个 pixel (0x8A–0x80)

## 2.2 Font16.bin 文件结构

### 2.2.1 文件结构

在该文件中包含了所有 Unicode 字符的字形数据，包括可显示与不可显示的，即从 0x0000~0xFFFF，每个 Unicode 字符的字形数据占用 33 个 bytes，其中有效数据为 32 个 bytes，其他剩余 1 个字节为 iType，包含字符类型及宽度信息。

使用 UltraEdit 或 WinHex 二进制工具打开该字库文件，可以查看该文件的组成结构：如下图所示：



Font16.bin 文件结构

如上图所示前 32 个字节为 Unicode 字符的字形数据，最后 1 个字节为字符的类型，该类型包含了字符的字形种类及宽度信息。

具体描述如下表：

iType（第 33 个字节）	备注（单位：pixels）
0x10	仅当语言为俄语时，字符宽度为 8
0x20	字符宽度为 12
0x80	字符 Unicode 大于 0x7F 时，字符宽度为 12
0x80 – 0xFF	字符宽度为（iType – 0x80）

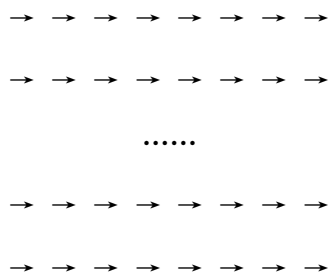
iType 类型含义

代码详见 LcdChar.c 中 LCD\_NFDispUnicodeChar 接口。



## 2.2.2 字形结构

同 2.1.2 节中描述的一样，每个 Unicode 字符的字形数据是按照从左→右，从上↓下的扫描顺序获取出来保存的，每行 16 个 bit（每个 pixel 占一个 bit），用 2 个字节表示，有 16 行，共 32 个字节的数据。



以大写的字符“W”举例说明：“W”字符在 font16.bin 中的位置： $0x57 \times 33 = 0xB37$

其字形数据为：从左到右，从上到下。

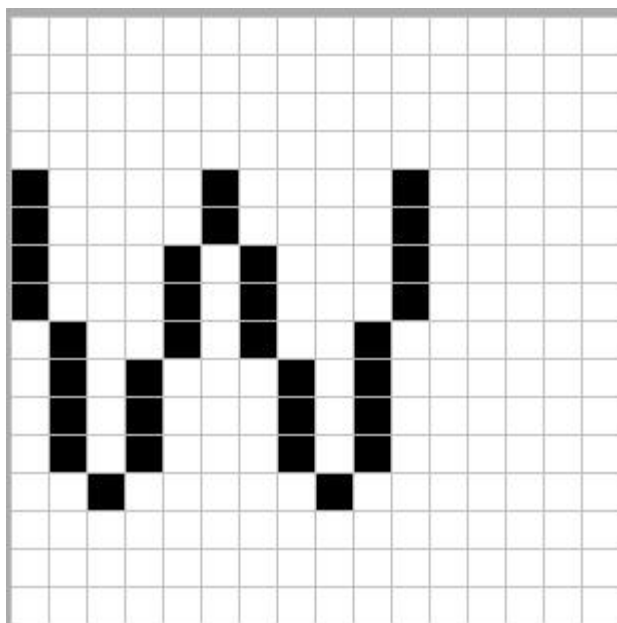
十六进制表示	二进制表示
00 00:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00 00:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00 00:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00 00:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
84 20:	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
84 20:	1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0
8A 20:	1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
8A 20:	1 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0
4A 40:	0 1 0 0 1 0 1 0 0 1 0 0 0 0 0 0
51 40:	0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0
51 40:	0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0
51 40:	0 1 0 1 0 0 0 1 0 1 0 0 0 0 0 0
20 80:	0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0
00 00:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
00 00:	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

00 00: 0000000000000000

Font16.bin x																
	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
00000b10h:	00	00	00	00	00	88	00	00	00	00	00	00	00	00	82	00
00000b20h:	82	00	82	00	44	00	44	00	28	00	28	00	10	00	10	00
00000b30h:	00	00	00	00	00	00	88	00	00	00	00	00	00	00	84	00
00000b40h:	20	84	20	8A	20	8A	20	4A	40	51	40	51	40	51	40	20
00000b50h:	80	00	00	00	00	00	00	8C	00	00	00	00	00	00	00	00

“W” 在文件中的位置与内容

“W” 的显示效果图:



“W” 字符显示，宽度为 12 个 pixel (0x8C – 0x80)

### 3 用户自定义字库

用户如需使用自定义的字库，需按照以上章节描述的组织适当的字库文件和所有 Unicode 字符字形数据，同时需要命名为 Font12.bin 或 Font16.bin 的文件名，否则无法顺便打包进固件文件。