

# 《iBoard 电子学堂》群课 第八讲

## 点阵字体驱动技术

---

关于 《iBoard 电子学堂》 .....



《iBoard 电子学堂》是一个综合型的电子研发开发平台，适合在校学生、一线工程师及电子爱好者等。

**交流方式：**

官方博客：[XiaomaGee.cnblogs.com](http://XiaomaGee.cnblogs.com)

官方论坛：[www.oshcn.com](http://www.oshcn.com)

官方淘宝店铺：[i-Board.taobao.com](http://i-Board.taobao.com)

**QQ群：**

《iBoard 电子学堂 群【A】》：204255896（500人，已满）

《iBoard 电子学堂 群【B】》：165201798（500人，已满）

《iBoard 电子学堂 群【C】》：215053598（200人高级群）

《iBoard 电子学堂 群【D】》：215054675（200人高级群）

《iBoard 电子学堂 群【E】》：215055211（200人高级群）

《iBoard 电子学堂 群【F】》：78538605（200人高级群）

《iBoard 电子学堂 群【G】》：158560047（500人高级群）

XiaomaGee(15959622) 20:30:01  
关于字体驱动，我首先简单介绍一下。

XiaomaGee(15959622) 20:31:02  
电子产品的开发，客户、老板要求越来越高，原来我们做东西，加几个数码管设置参数就了不得了，现在不一样了。

XiaomaGee(15959622) 20:31:22  
现在追求的都是图文并茂的傻瓜式操作

XiaomaGee(15959622) 20:31:48  
其中，文字作为人机界面交互主要的信息载体，里面有很多学问。

XiaomaGee(15959622) 20:32:53  
从技术角度来说，字体分为矢量字体和点阵字体，矢量字体是通过存储的矢量信息，经过运算后产生图形，然后显示在计算机上。

XiaomaGee(15959622) 20:33:22  
所以，矢量字体对系统存储空间、计算速度等要求较高。

XiaomaGee(15959622) 20:34:06  
但是呢，矢量字体的好处是显而易见的，就是不管放大到多少倍，他都不会失真（都是平滑的）

XiaomaGee(15959622) 20:34:43  
我们计算机上，大部分字体都属于矢量字体。如果是 windows，就在 系统根目录 windows 下 font 目录里。

XiaomaGee(15959622) 20:35:22  
我打开一个，让大家看一下。

王紫豪-XiaomaGee(15959622) 20:35:42



王紫豪-XiaomaGee(15959622) 20:36:10

大家全屏看图片，这样效果较好。图中就是 windows 系统下的 tahoma 矢量字体。

王紫豪-XiaomaGee(15959622) 20:37:09

从大小我们可以看出，仅仅的一个英文字体，占了 600 多 k 的存储空间。我们用单片机去驱动，是没有这个条件的。

王紫豪-XiaomaGee(15959622) 20:37:36

相对于矢量字体，另外一个字体类型是：点阵字体

王紫豪-XiaomaGee(15959622) 20:38:02

点阵字体，顾名思义，就是以“点阵”的形式，去存储我们得字体信息。

王紫豪-XiaomaGee(15959622) 20:38:40

所以，点阵字库一般较小，而且便于生成最终显示的“图形”。

王紫豪-XiaomaGee(15959622) 20:39:21

我们这节课，就讲一下基于点阵字体的英文、汉字驱动方法。

王紫豪-XiaomaGee(15959622) 20:40:12

为了便于后面的讲解，我首先说下一个基本概念，字符的编码方法。

王紫豪-XiaomaGee(15959622) 20:40:54

讲到字符编码，我们最容易接触到的，也就是 ASCII 这个名词了。

王紫豪-XiaomaGee(15959622) 20:41:43

ASCII 是 (American Standard Code for Information Interchange, 美国信息互换标准代码)，这个句子的缩写。

王紫豪-XiaomaGee(15959622) 20:42:19

这也是当前计算机行业的一个标准。

王紫豪-XiaomaGee(15959622) 20:42:51

我在群聊打的每一个英文字符，都是通过这个编码方式传送到各位电脑里的。

王紫豪-XiaomaGee(15959622) 20:43:52

ASCII 码表，我贴一下，这个其实很有用。

王紫豪-XiaomaGee(15959622) 20:44:00

Dec	Hx	Oct	Char	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr	Dec	Hx	Oct	Html	Chr
0	0	000	NUL (null)	32	20	040	&#32; Space		64	40	100	&#64; @		96	60	140	&#96; `	
1	1	001	SOH (start of heading)	33	21	041	&#33; !		65	41	101	&#65; A		97	61	141	&#97; a	
2	2	002	STX (start of text)	34	22	042	&#34; "		66	42	102	&#66; B		98	62	142	&#98; b	
3	3	003	ETX (end of text)	35	23	043	&#35; #		67	43	103	&#67; C		99	63	143	&#99; c	
4	4	004	EOT (end of transmission)	36	24	044	&#36; \$		68	44	104	&#68; D		100	64	144	&#100; d	
5	5	005	ENQ (enquiry)	37	25	045	&#37; %		69	45	105	&#69; E		101	65	145	&#101; e	
6	6	006	ACK (acknowledge)	38	26	046	&#38; &		70	46	106	&#70; F		102	66	146	&#102; f	
7	7	007	BEL (bell)	39	27	047	&#39; '		71	47	107	&#71; G		103	67	147	&#103; g	
8	8	010	BS (backspace)	40	28	050	&#40; (		72	48	110	&#72; H		104	68	150	&#104; h	
9	9	011	TAB (horizontal tab)	41	29	051	&#41; )		73	49	111	&#73; I		105	69	151	&#105; i	
10	A	012	LF (NL line feed, new line)	42	2A	052	&#42; *		74	4A	112	&#74; J		106	6A	152	&#106; j	
11	B	013	VT (vertical tab)	43	2B	053	&#43; +		75	4B	113	&#75; K		107	6B	153	&#107; k	
12	C	014	FF (NP form feed, new page)	44	2C	054	&#44; ,		76	4C	114	&#76; L		108	6C	154	&#108; l	
13	D	015	CR (carriage return)	45	2D	055	&#45; -		77	4D	115	&#77; M		109	6D	155	&#109; m	
14	E	016	SO (shift out)	46	2E	056	&#46; .		78	4E	116	&#78; N		110	6E	156	&#110; n	
15	F	017	SI (shift in)	47	2F	057	&#47; /		79	4F	117	&#79; O		111	6F	157	&#111; o	
16	10	020	DLE (data link escape)	48	30	060	&#48; 0		80	50	120	&#80; P		112	70	160	&#112; p	
17	11	021	DC1 (device control 1)	49	31	061	&#49; 1		81	51	121	&#81; Q		113	71	161	&#113; q	
18	12	022	DC2 (device control 2)	50	32	062	&#50; 2		82	52	122	&#82; R		114	72	162	&#114; r	
19	13	023	DC3 (device control 3)	51	33	063	&#51; 3		83	53	123	&#83; S		115	73	163	&#115; s	
20	14	024	DC4 (device control 4)	52	34	064	&#52; 4		84	54	124	&#84; T		116	74	164	&#116; t	
21	15	025	NAK (negative acknowledge)	53	35	065	&#53; 5		85	55	125	&#85; U		117	75	165	&#117; u	
22	16	026	SYN (synchronous idle)	54	36	066	&#54; 6		86	56	126	&#86; V		118	76	166	&#118; v	
23	17	027	ETB (end of trans. block)	55	37	067	&#55; 7		87	57	127	&#87; W		119	77	167	&#119; w	
24	18	030	CAN (cancel)	56	38	070	&#56; 8		88	58	130	&#88; X		120	78	170	&#120; x	
25	19	031	EM (end of medium)	57	39	071	&#57; 9		89	59	131	&#89; Y		121	79	171	&#121; y	
26	1A	032	SUB (substitute)	58	3A	072	&#58; :		90	5A	132	&#90; Z		122	7A	172	&#122; z	
27	1B	033	ESC (escape)	59	3B	073	&#59; ;		91	5B	133	&#91; [		123	7B	173	&#123; {	
28	1C	034	FS (file separator)	60	3C	074	&#60; <		92	5C	134	&#92; \		124	7C	174	&#124;	
29	1D	035	GS (group separator)	61	3D	075	&#61; =		93	5D	135	&#93; ]		125	7D	175	&#125; }	
30	1E	036	RS (record separator)	62	3E	076	&#62; >		94	5E	136	&#94; ^		126	7E	176	&#126; ~	
31	1F	037	US (unit separator)	63	3F	077	&#63; ?		95	5F	137	&#95; _		127	7F	177	&#127; DEL	

Source: [www.LookupTables.com](http://www.LookupTables.com)

王紫豪-XiaomaGee(15959622) 20:44:38

这里我们就可以简单地理解成英文字体的编码方式了。就是采用一个字节（8位），按照上表的方式，存储在计算机里。

王紫豪-XiaomaGee(15959622) 20:45:39

那么对于汉字来说呢？汉字可是好几千呢？一个字节表示不下啊，一个字节，才能表示 256 个( $2^8 = 256$ )

王紫豪-XiaomaGee(15959622) 20:46:16

汉字呢，其实是用两个字节表示的。

王紫豪-XiaomaGee(15959622) 20:47:17

当然，为了迎合全世界所有的语言，出现了 Unicode 这个编码方法，这个不在咱们的讨论范围内。

王紫豪-XiaomaGee(15959622) 20:47:46

我们首先说说英文字体怎么驱动的。

王紫豪-XiaomaGee(15959622) 20:49:14

刚才我们说了 ASCII 码的定义，其实 ascii 码里面很多都是用做控制的（例如换行符 \n，回车符\r），这些都是不可显示的。可显示的不是太多，就包含了数字、大小写字母、一些符号等等。就是键盘上那些东西

王紫豪-XiaomaGee(15959622) 20:50:21

[http://files.cnblogs.com/xiaomagee/iBoard\\_demo.7z](http://files.cnblogs.com/xiaomagee/iBoard_demo.7z)

王紫豪-XiaomaGee(15959622) 20:50:46

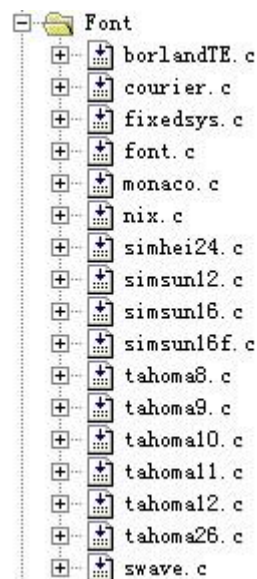
大家下载一下 iboard infinity 软件包，我配合这个给大家说一下。



王紫豪-XiaomaGee(15959622) 20:52:01

下载好后，用 keil MDK 打开工程。

王紫豪-XiaomaGee(15959622) 20:52:17



王紫豪-XiaomaGee(15959622) 20:52:34

font 文件夹内的，就是所有字库的驱动。

王紫豪-XiaomaGee(15959622) 20:53:38

讲解之前，还有个概念需要说一下。

王紫豪-XiaomaGee(15959622) 20:54:12

细心的网友应该能发现，单个英文字符有的宽度一样，有的宽度不一样。

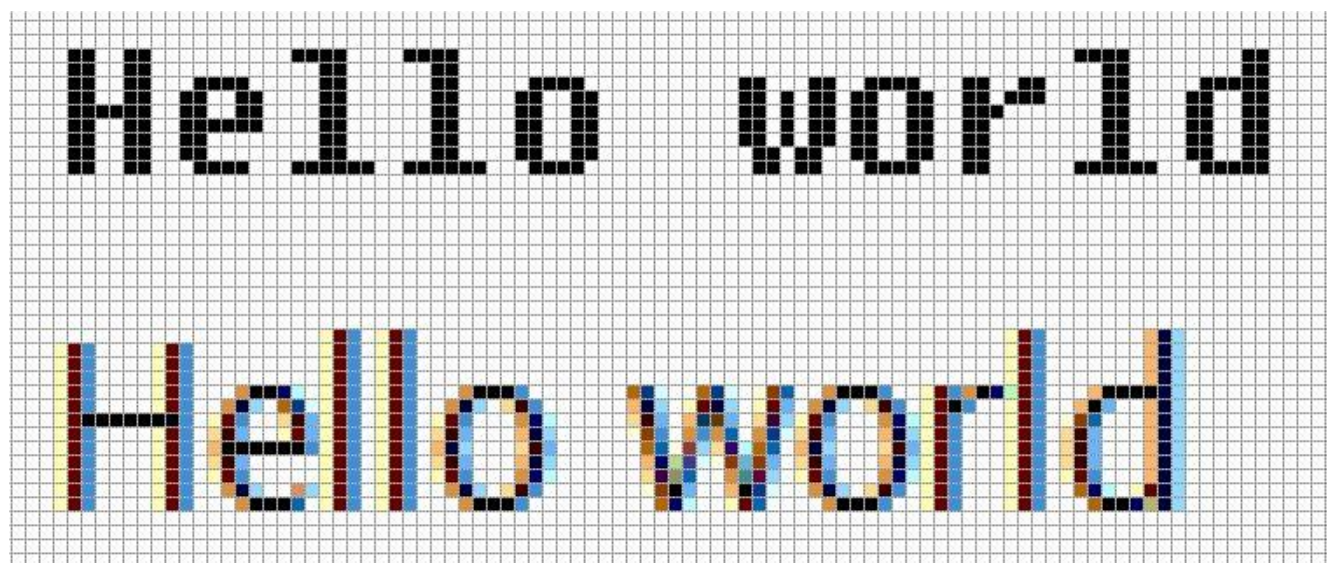
王紫豪-XiaomaGee(15959622) 20:54:26

我用画笔，分别写几个字符，给大家看一下。

XiaomaGee(15959622) 20:54:28

我用画笔，分别写几个字符，给大家看一下。

XiaomaGee(15959622) 20:56:20



XiaomaGee(15959622) 20:56:48

上面一行 hello word, 是 fixedsys 字体

XiaomaGee(15959622) 20:57:01

下面一行, 是 tahoma 字体

XiaomaGee(15959622) 20:57:39

大家可以看到, fixedsys 每个字符的宽度是一样的。而下面一行, 字符的宽度不一样 (w 的宽度, 明显大于 l)

XiaomaGee(15959622) 20:57:55

为了方便看, 我把它放大了。

XiaomaGee(15959622) 20:58:52

我们把宽度相同的字体, 叫做等宽字体 (英文翻译为 monospaced font)。

XiaomaGee(15959622) 20:59:23

宽度不一样的, 叫做非等宽字体, 或者比例字体。

XiaomaGee(15959622) 20:59:47

proportional font

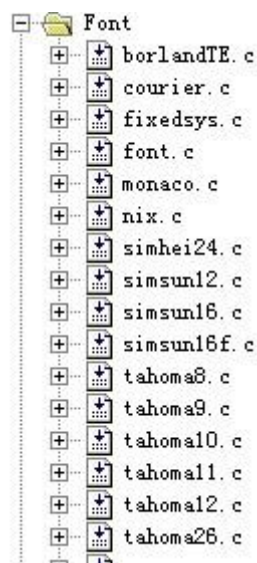
XiaomaGee(15959622) 21:00:04

为什么强调着一个概念呢, 因为这个东西, 关系着我们怎么去显示他。

XiaomaGee(15959622) 21:00:36

还回到刚才的程序上。

XiaomaGee(15959622) 21:01:00



XiaomaGee(15959622) 21:01:36

这些字体中, boardlandTE, courier, fixedsys, monaco 都是等宽字体, 而 tahoma 是比例字体。

XiaomaGee(15959622) 21:03:02

可以说, 根据不同的场合, 使用不同的字体, 才能做出完美的人机界面。

XiaomaGee(15959622) 21:03:15



XiaomaGee(15959622) 21:03:49

这是刚才那些字体的一个演示

XiaomaGee(15959622) 21:04:33

每一件事情都是有它的意义的。

the quick brown fox jumps over the lazy dog 这句话，也是有他的含义的。

XiaomaGee(15959622) 21:05:14

. 有心的网友可以数一下，这一句话包含了所有的 26 个英文字符。

XiaomaGee(15959622) 21:05:42

下面我们就说一下，等宽和比例字体的驱动方法。

XiaomaGee(15959622) 21:06:15

对于底层来说，所有的字体显示，最终都转化成描点操作。这句话我不多做解释

XiaomaGee(15959622) 21:06:48

我们以 fixedsys 为例，大家打开 fixedsys.c 这个源代码。

XiaomaGee(15959622) 21:07:22

```

1  /*-----*/
2  FILE NAME    : fixedsys.c
3  DESCRIPTION  : fixedsys
4  VERSION     : 0.0.0 (C)XiaomaGee
5  AUTHOR      : XiaomaGee
6  CREATE DATE : 2010-6-9
7  MODIFY DATE :
8  LOGS        :-\
9  -----*/
10
11 //----- Include files -----//
12 #include "..\include\font.h"
13 #include <string.h>
14
15 //----- Function Prototype -----//
16 static int get(FONT_MATRIX_T *);
17
18 //----- Variable -----//
19 extern const unsigned char rc_fixedsys[];
20
21 SINGLE_BYTE_FONT_T fixedsys = {
22     .name = "Fixedsys",
23     .id = FONT_UNREGISTERED,
24     .height = 16,
25     .width = 8,
26     .dat = (void*)rc_fixedsys,
27     .get_matrix = get,
28     .pre = NULL,
29     .next = NULL
30 };
31 //----- Function -----//
32 /*-----*/
33 NAME      : --
34 ENTRY TYPE: void
35 RETURN TYPE: void
36 PURPOSE   : ---
37 VERSION   : 0.0.0
38 DESCRIPTION:
39 CREATE DATE: ---
40 LOGS      :
41 -----*/
42 static int
43 get(FONT_MATRIX_T *f)
44 {
45     f->direction = DIRECTION_U;
46     f->height = fixedsys.height;
47     f->width = fixedsys.width;
48     f->dat = (char*)fixedsys.dat + (f->character[0] - ' ') * 16;
49
50     return 0;
51 }
52

```

XiaomaGee(15959622) 21:07:57

这些是这个字体的核心代码。当然，还有一个字库数组，没有显示出来。

XiaomaGee(15959622) 21:08:21



我把他放到 rc\_fixedsys.c 里面了。

XiaomaGee(15959622) 21:09:43

!..#&'()\*+,-./0123456789:;<=>@ABCDEFGHIJKLMNPQRSTUVWXYZ[\]^\_`abcdefghijklmnopqrstuvwxyz{|}~

XiaomaGee(15959622) 21:10:00

通过按照 ascii 码存储的一个序列，我们把它当成一个图片，需要显示什么，就给他个偏移量，就查到了这个字符的点阵信息了。

XiaomaGee(15959622) 21:10:27

当然，为啥让他是镜像的，只是我们取点阵的一个手段而已。

XiaomaGee(15959622) 21:11:25

英文字体占用空间都不大，所以我们就直接把他放到 arm 内部的 flash 里面了。  
fixedsys 才占用了 1.x k 的 flash

XiaomaGee(15959622) 21:12:25

总结一下，等宽字体，我们只需要把全部字体按照 ascii 顺序排列，然后显示的时候，按照偏移量去查找点阵就可以了。

XiaomaGee(15959622) 21:13:37

下面我说说比例字体，某些方面，比例字体显示效果要好很多。windows 系统里，很多时候都使用了比例字体。

XiaomaGee(15959622) 21:14:11

我们随便打开一个，例如：tahoma9.c 这里面，

XiaomaGee(15959622) 21:14:51

大家看到的首先是一个注释性的东西。

XiaomaGee(15959622) 21:15:12

```
-----  
!'"#$%&'()*+,-./0123456789:;<=>?@ABCDEFGHIJKLMNopQRSTUVWXYZ[\]^_`abcdefghijklmnopqrstuvwxyz  
-----
```

XiaomaGee(15959622) 21:15:32

这个，其实就是 ascii 码可显示字符排列顺序。

XiaomaGee(15959622) 21:15:54

有的网友就问了，为啥这样排列啊，这个咱管不着，米国人搞的。

XiaomaGee(15959622) 21:16:31

```
const short int tahoma9_offset[] = {  
    // ! " # $ % & ' ( )  
    0, 6, 10, 15, 24, 31, 43, 51, 54, 59,  
    // * + , - . / 0 1 2 3  
    64, 71, 79, 82, 87, 90, 96, 103, 110, 117,  
    // 4 5 6 7 8 9 : ; <  
    124, 131, 138, 145, 152, 159, 166, 169, 173, 183,  
    // = > ? @ A B C D E F  
    197, 201, 207, 218, 226, 233, 240, 248, 255, 261,  
    // G H I J K L M N O P  
    269, 277, 281, 286, 293, 299, 308, 316, 325, 332,  
    // Q R S T U V W X Y Z  
    341, 348, 355, 363, 371, 379, 391, 398, 406, 413,  
    // [ \ ] ^ _ ` a b c d  
    418, 423, 428, 437, 445, 450, 457, 464, 470, 477,  
    // e f g h i j k l m n  
    484, 489, 496, 503, 505, 509, 515, 517, 527, 534,  
    // o p q r s t u v w x  
    541, 548, 555, 559, 564, 569, 576, 582, 592, 598,  
    // y z { | } ~  
    604, 609, 615, 620, 626, 635  
};
```

XiaomaGee(15959622) 21:17:45

区别于等宽字体（等宽字体的偏移量是线性的，也就是说宽度都是一样的），而

比例字体，每个字符的宽度未知，所以我们就需要一个存储偏移量的数据表；就是上表。

XiaomaGee(15959622) 21:19:40

X-GUI 里面的字库驱动，我采用了同一封装的方法，真正显示字体的时候，屏蔽了底层操作。只是根据字体信息（长度、宽度等，以及取点阵信息），来完成字符的显示。

XiaomaGee(15959622) 21:20:07

休息一会，21:30 回来接着讲汉字驱动。



XiaomaGee(15959622) 21:31:48

刚才有网友，对这个仿真数码管的字体，比较感兴趣，我下面就说一下他的原理哦

XiaomaGee(15959622) 21:32:12

这个是《iBoard 电子学堂》电压表的界面。

XiaomaGee(15959622) 21:32:40

实际显示效果极佳（特别是较远去看的时候），比截图要好看。

XiaomaGee(15959622) 21:33:16

他的实现，在我们的 font 目录下 nix.c 里面。

XiaomaGee(15959622) 21:34:51

可以说，做到了真正意义上的“仿真”数码管。

XiaomaGee(15959622) 21:35:13

代码较长，我截取一部分。

XiaomaGee(15959622) 21:35:59

这句话，就是取出来数字 1 的字模

XiaomaGee(15959622) 21:36:03

```
case '1':  
    for (i = 0; i < 1332; i++) {  
        font_buffer[i] = (rc_nix96[1][i] | rc_nix96[2][i] | (flag ? rc_nix96[7][i] : 0))  
    }  
    break;
```

XiaomaGee(15959622) 21:36:56

当然，如果下功夫，可以用这个做成一个简单的“矢量”原理的仿真数码管，也是很好的。

XiaomaGee(15959622) 21:37:34

下面我说一下 X-GUI 对字库驱动的归一化处理。

XiaomaGee(15959622) 21:38:22

刚才说那么多英文字体驱动，大家看到的代码都差不多。其实，每个字体对上一层的信息，只需要提供一个这样的结构体既可：

XiaomaGee(15959622) 21:38:40

```
SINGLE_BYTE_FONT_T nix48 = {
    .name = "Nix48",
    .id = FONT_UNREGISTERED,
    .height = 56,
    .width = 48,
    .dat = (void*)rc_nix48,
    .get_matrix = get48,
    .pre = NULL,
    .next = NULL
};
```

XiaomaGee(15959622) 21:39:06

这个结构体的原型，再 font.h 里面定义。

XiaomaGee(15959622) 21:39:32

```
42 typedef struct _SINGLE_BYTE_FONT{
43     char * name;
44     unsigned long int id;
45     int height;
46     int width; //width!=0 is monospaced font, width==0 is proportional font
47     char first_char;
48
49     int (*get_matrix)(FONT_MATRIX_T *);
50     void * dat;
51     struct _SINGLE_BYTE_FONT * pre;
52     struct _SINGLE_BYTE_FONT * next;
53 }SINGLE_BYTE_FONT_T;
```

XiaomaGee(15959622) 21:39:53

它包含了一些基本信息，通过这个统一的接口，我们就很方便的去添加自己的字体。

王紫豪-XiaomaGee(15959622) 21:43:49

其实，不管是彩屏的液晶，还是单色的液晶，合理的运用资源，都可以做出来叫好看的界面。

王紫豪-XiaomaGee(15959622) 21:44:18





王紫豪-XiaomaGee(15959622) 21:44:46

这个是我原来采用单色液晶做的界面截图，实际效果也很好。

王紫豪-XiaomaGee(15959622) 21:45:19

下面说说汉字驱动。

王紫豪-XiaomaGee(15959622) 21:45:45

汉字驱动，比英文来说麻烦很多，主要是他的字库太大了。

王紫豪-XiaomaGee(15959622) 21:46:29

刚才讲的英文字体，都可以存 stm32 等单片机的 flash 里，汉字库可不敢这样存，一个汉字库，至少也得 200 多 k 字节。

王紫豪-XiaomaGee(15959622) 21:47:00

但是呢，有些场合我们只需要使用固定的那几十个字，或者几百个字；这时候怎么办呢？

王紫豪-XiaomaGee(15959622) 21:47:47

很多年前，我做过一个取摸软件，就是为了这样的应用，在汉字用量不大的情况下，采用小工具自动取出来字库的模。

王紫豪-XiaomaGee(15959622) 21:48:03

这里我跟大家共享一下。

王紫豪-XiaomaGee(15959622) 21:51:02



王紫豪-XiaomaGee(15959622) 21:51:49

包含了四个文件：

1. font\_gen.exe 取摸主程序
2. gen.bat，是一个批处理
3. hzk12, 12 点阵字库
4. source.txt，保存我们需要取摸的汉字

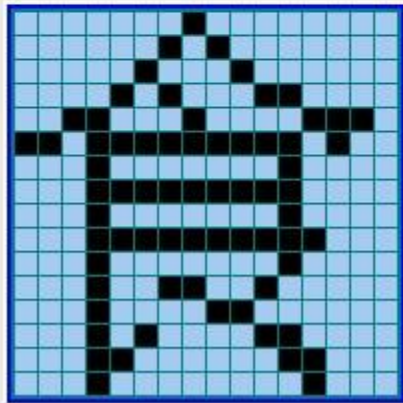
## 液晶汉字模提取

```
// ----- 汉字字模的数据结构定义 ----- //
typedef struct typFNT_GB16          // 汉字字模数据结构
{
    signed char Index[2];            // 汉字内码索引
    char Msk[32];                    // 点阵码数据
};

////////////////////////////////////
// 汉字字模表
// 汉字库：宋体16.dot, 横向取模左高位, 数据排列：从左到右从上到下
////////////////////////////////////
struct typFNT_GB16 code GB_16[] =    // 数据表
{
    "明", 0x00, 0x04, 0x04, 0xFE, 0x7E, 0x84, 0x44, 0x84,
           0x44, 0x84, 0x44, 0xFC, 0x7C, 0x84, 0x44, 0x84,
           0x44, 0x84, 0x44, 0xFC, 0x7C, 0x84, 0x44, 0x84,
           0x01, 0x04, 0x01, 0x04, 0x02, 0x14, 0x04, 0x08,

    "天", 0x00, 0x08, 0x7F, 0xFC, 0x01, 0x00, 0x01, 0x00,
           0x01, 0x00, 0x01, 0x04, 0xFF, 0xFE, 0x01, 0x00,
           0x02, 0x80, 0x02, 0x80, 0x04, 0x40, 0x04, 0x40,
           0x08, 0x20, 0x10, 0x10, 0x20, 0x0E, 0xC0, 0x04,

    "有", 0x02, 0x00, 0x02, 0x04, 0xFF, 0xFE, 0x04, 0x00,
           0x04, 0x10, 0x0F, 0xF8, 0x08, 0x10, 0x18, 0x10,
```



明天有日食

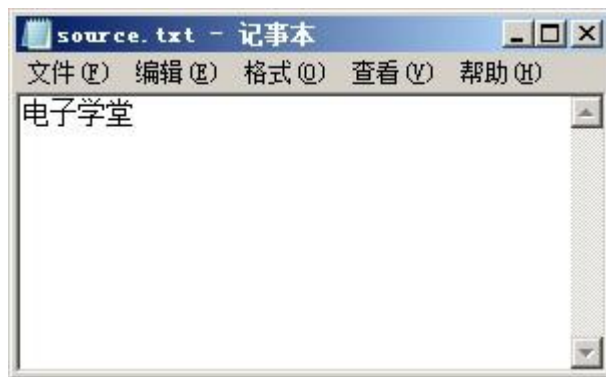
宋体16 H=16 W=16

食

王紫豪-XiaomaGee(15959622) 21:52:35

我们只需要把取摸的汉字写在 source.txt 里，例如，我们要取 ” 电子学堂 “，这四个字的模

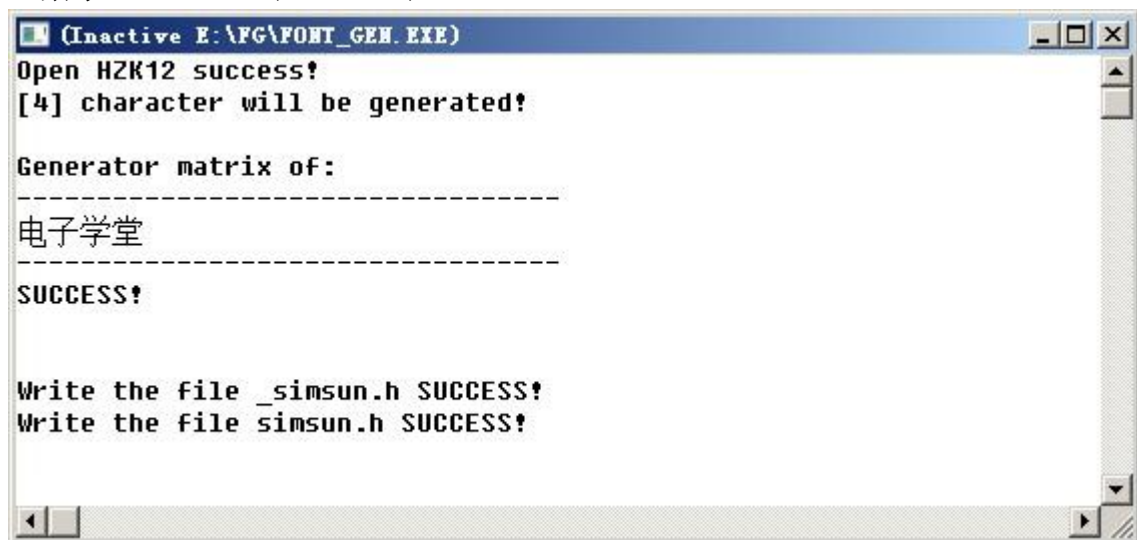
王紫豪-XiaomaGee(15959622) 21:52:50



王紫豪-XiaomaGee(15959622) 21:53:13

双击运行

王紫豪-XiaomaGee(15959622) 21:53:33



王紫豪-XiaomaGee(15959622) 21:53:45

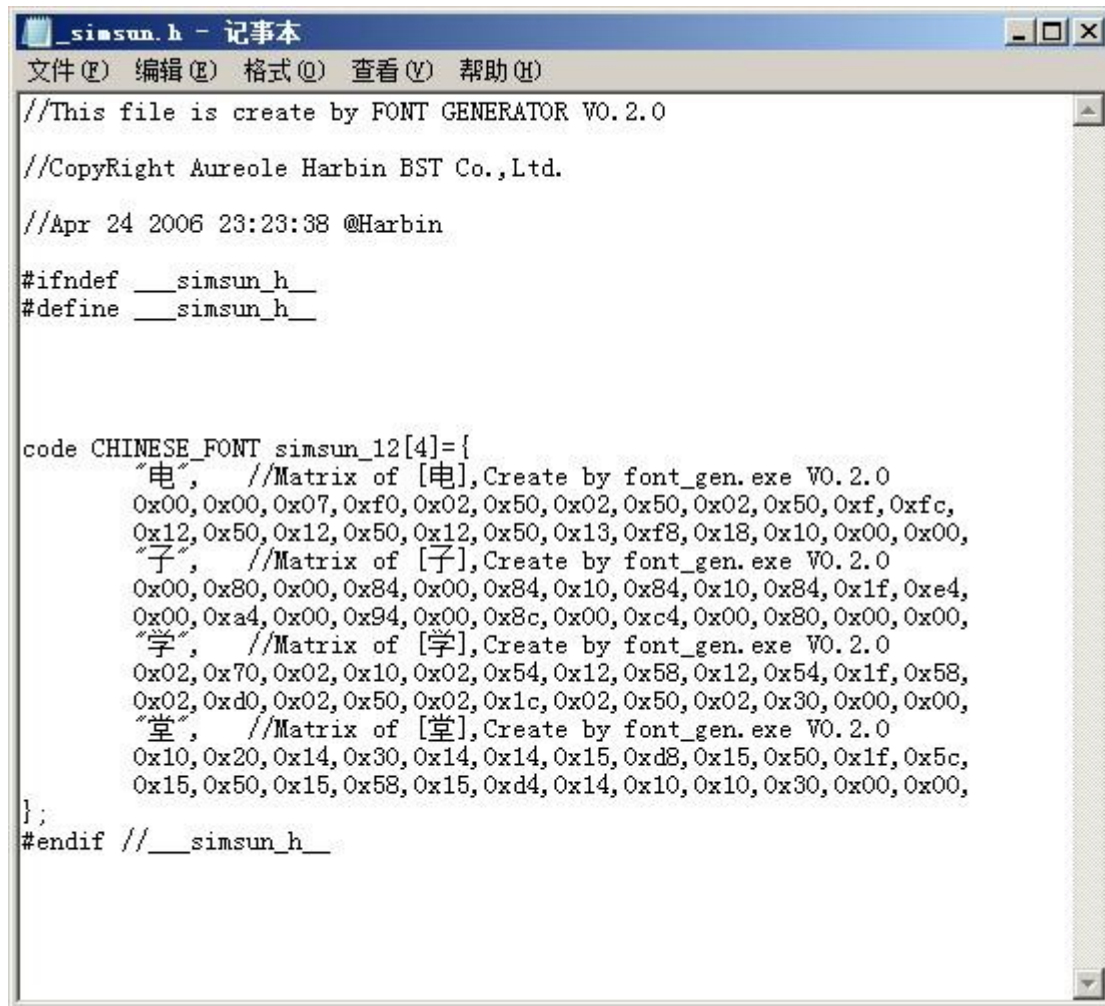
这是弹出界面。

王紫豪-XiaomaGee(15959622) 21:54:12

最后会在这个目录内，产生两个文件



王紫豪-XiaomaGee(15959622) 21:54:36



```
_simsun.h - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)

//This file is create by FONT GENERATOR V0.2.0

//CopyRight Aureole Harbin BST Co.,Ltd.

//Apr 24 2006 23:23:38 @Harbin

#ifndef __simsun_h__
#define __simsun_h__

code CHINESE_FONT simsun_12[4]={
    "电", //Matrix of [电],Create by font_gen.exe V0.2.0
    0x00,0x00,0x07,0xf0,0x02,0x50,0x02,0x50,0x02,0x50,0xf,0xfc,
    0x12,0x50,0x12,0x50,0x12,0x50,0x13,0xf8,0x18,0x10,0x00,0x00,
    "子", //Matrix of [子],Create by font_gen.exe V0.2.0
    0x00,0x80,0x00,0x84,0x00,0x84,0x10,0x84,0x10,0x84,0x1f,0xe4,
    0x00,0xa4,0x00,0x94,0x00,0x8c,0x00,0xc4,0x00,0x80,0x00,0x00,
    "学", //Matrix of [学],Create by font_gen.exe V0.2.0
    0x02,0x70,0x02,0x10,0x02,0x54,0x12,0x58,0x12,0x54,0x1f,0x58,
    0x02,0xd0,0x02,0x50,0x02,0x1c,0x02,0x50,0x02,0x30,0x00,0x00,
    "堂", //Matrix of [堂],Create by font_gen.exe V0.2.0
    0x10,0x20,0x14,0x30,0x14,0x14,0x15,0xd8,0x15,0x50,0x1f,0x5c,
    0x15,0x50,0x15,0x58,0x15,0xd4,0x14,0x10,0x10,0x30,0x00,0x00,
};
#endif //__simsun_h__
```

王紫豪-XiaomaGee(15959622) 21:54:46

这个就是程序自动生成的。

Allen(15330302) 21:55:05

不错

王紫豪-XiaomaGee(15959622) 21:55:19

可以看出，他的优势在于，可以批量取摸（而不用单个汉字去弄）。只需要把我们要显示的，放到一个文本文件里既可，它自动就取出来了。

王紫豪-XiaomaGee(15959622) 21:55:34

而且重复的汉字，他可以剔除掉，节约空间

王紫豪-XiaomaGee(15959622) 21:56:28

其实我这个也不人性化，只是我多年自己用的一个小工具

王紫豪-XiaomaGee(15959622) 21:56:30

我自己写的。

王紫豪-XiaomaGee(15959622) 21:56:37

课后我会共享

王紫豪-XiaomaGee(15959622) 21:57:11

刚才说的情况，是不需要显示大量汉字的时候，的一个节约资源的解决方案。

王紫豪-XiaomaGee(15959622) 21:57:40



如果需要完整的汉字呢，我们只能把字库，存储于外部的 flash，或者 sd 卡，或者 u 盘中。

王紫豪-XiaomaGee(15959622) 21:58:06

《iboard 电子学堂》专门采用了 2M 的 spi flash 存储字库。

王紫豪-XiaomaGee(15959622) 21:58:55

回到代码中，大家可以看看，simsun12、16 点阵，还有 16 点阵繁体，以及 24 点阵黑体，都存储于外部的 spi flash 里。

王紫豪-XiaomaGee(15959622) 21:59:14

做产品没人这样做

王紫豪-XiaomaGee(15959622) 21:59:18

片内 flash 很宝贵

王紫豪-XiaomaGee(15959622) 21:59:57

以 16 点阵字库来说，他的核心代码非常少，就一个函数。

王紫豪-XiaomaGee(15959622) 22:00:10

```
static int
get(FONT_MATRIX_T *f)
{
    unsigned long int offset;

    offset = f->character[0];
    offset -= 0xa1;
    offset *= 94;
    offset += f->character[1];
    offset -= 0xa1;
    offset *= 32;

    f->direction = DIRECTION_H;
    f->height = simsun16.height;
    f->width = simsun16.width;

    flash.read_32(offset + SIMSUN16_ADDRESS_BASE, font_buffer);
    f->dat = font_buffer;

    return 0;
}
```

王紫豪-XiaomaGee(15959622) 22:00:22

我给大家解释下代码的意思。

王紫豪-XiaomaGee(15959622) 22:00:39

```
offset = f->character[0];
offset -= 0xa1;
offset *= 94;
offset += f->character[1];
offset -= 0xa1;
offset *= 32;
```

王紫豪-XiaomaGee(15959622) 22:01:15

这些语句，是根据汉字内码，计算出这个汉字在字库中的偏移量，从而定位到点阵信息

王紫豪-XiaomaGee(15959622) 22:01:33

关于这个公式，网上多得是，他是汉字编码的规定。

王紫豪-XiaomaGee(15959622) 22:01:57

```
flash.read_32(offset + SIMSUN16_ADDRESS_BASE, font_buffer);  
f->dat = font_buffer;
```

王紫豪-XiaomaGee(15959622) 22:02:14

这两句，是读出来点阵字库，交给上层处理。

王紫豪-XiaomaGee(15959622) 22:02:43

这是存储于 spi flash 里的，在其他空间也一样。

王紫豪-XiaomaGee(15959622) 22:03:10

```
static int  
get(FONT_MATRIX_T *f)  
{  
    unsigned long int offset;  
    static int first = 0;  
    unsigned int counter=0;  
    static FRESULT res;  
  
    if(first == 0){  
        res=f_open(&fil,"0:/system/hzk16",FA_READ);  
        first = 1;  
    }  
  
    offset = f->character[0];  
    offset -= 0xa1;  
    offset *= 94;  
    offset += f->character[1];  
    offset -= 0xa1;  
    offset *= 32;  
  
    res =f_lseek(&fil,offset);  
  
    res = f_read(&fil,font_buffer,32,&counter);  
  
    f->direction = DIRECTION_H;  
    f->height = simsun16.height;  
    f->width = simsun16.width;  
  
    //flash.read_32(offset + SIMSUN16_ADDRESS_BASE, font_buffer);  
    f->dat = font_buffer;  
  
    return 0;  
}
```

王紫豪-XiaomaGee(15959622) 22:04:05

这个函数，也是 16 点阵字库的驱动，他的字库存储于 SD 卡上，用 fatfs 作为文件系统读取的。

做软件就要做到一点，各层关联少，很容易就移植了。

王紫豪-XiaomaGee(15959622) 22:07:37

首先: font.c 实现了中英文(包括等宽和比例)字体的注册管理;

第二: font.c 实现了中英文字体的点阵驱动;

第三: font.c 实现了字间距、行间距以及反白效果的属性。

第四: 实现了可变参数列表功能

王紫豪-XiaomaGee(15959622) 22:08:20

对外的显示函数只有一个, 就是 font.printf

王紫豪-XiaomaGee(15959622) 22:08:34

我去代码里找一段话, 解释一下他的用处。

王紫豪-XiaomaGee(15959622) 22:09:53

```
s.x = 10;  
s.y = 235;  
s.color = COLOR_WHITE;  
s.background_color = COLOR_RED;
```

```
s.space.line = 2;  
s.space.word = 0;  
s.inverse = NULL;
```

```
font.printf(&s, "[iBoard infinity] firmware packages, Rev <%s>\nCopyright XiaomaGe
```

王紫豪-XiaomaGee(15959622) 22:10:51

这里的 s. 什么什么, 是我们自定义的

STRING\_T 的一个结构体。

王紫豪-XiaomaGee(15959622) 22:12:32

STRING\_T 每个成员的含义如下:

x 坐标 x

y 坐标 y

color 前景色

background\_color 背景色

inverse 包含反白信息的一个链表

space.line 行间距

space.word 字间距

王紫豪-XiaomaGee(15959622) 22:13:00

可以看出, 通过这个结构体信息, 我们就能把一个字符串, 很好滴表达到液晶显示器上了。

王紫豪-XiaomaGee(15959622) 22:13:49

当然, 这个函数支持中英文混合的。怎么去区别中英文混在一起显示, 大家可以参考代码。

王紫豪-XiaomaGee(15959622) 22:14:58

我们看看刚才那些代码实现的效果

王紫豪-XiaomaGee(15959622) 22:15:19



王紫豪-XiaomaGee(15959622) 22:15:35

以这个为例。

王紫豪-XiaomaGee(15959622) 22:15:56

屏幕下面、右边的虚拟按键，采用了 16 点阵的宋体字。

王紫豪-XiaomaGee(15959622) 22:16:19

中间对话框表头采用了 16 点阵宋体字和 fixedsys 英文字体

王紫豪-XiaomaGee(15959622) 22:16:42

对话框内容，采用了 12 点阵宋体字和 tahoma 9 号 比例字体。

王紫豪-XiaomaGee(15959622) 22:17:38

我们要表达不同属性的时候，怎么去体现他的差异化，也是我们使用那么多字体的唯一目的所在。

王紫豪-XiaomaGee(15959622) 22:18:49



王紫豪-XiaomaGee(15959622) 22:19:15

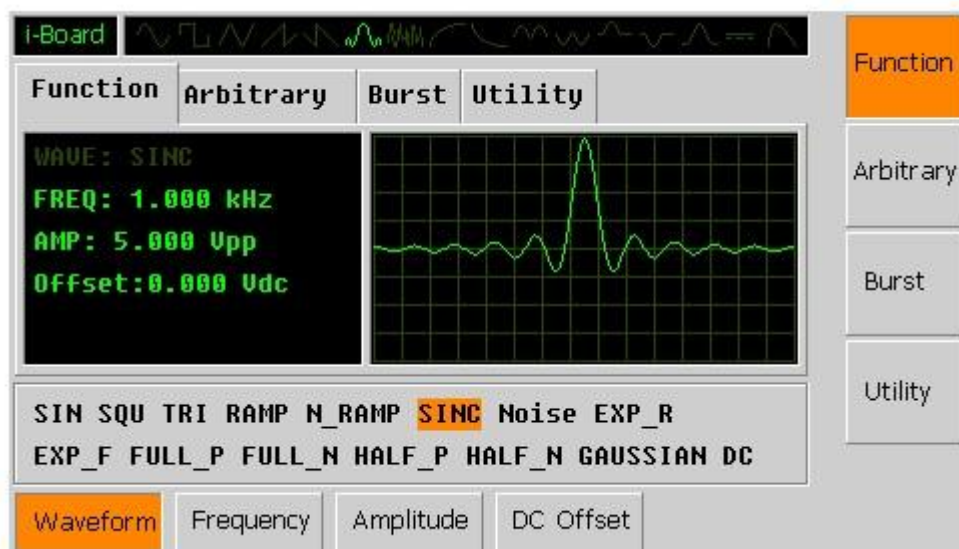
这个设置框，里面的大个子数字、英文字母，是采用了 tahoma 26 点阵字体。

王紫豪-XiaomaGee(15959622) 22:20:28

我给大家再找几个英文界面，看看他们的差异化。



王紫豪-XiaomaGee(15959622) 22:23:05



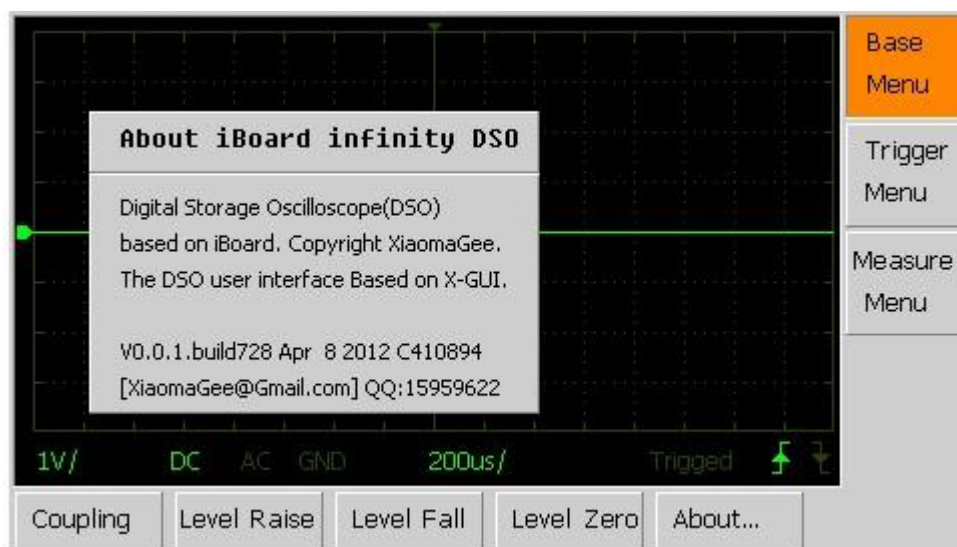
王紫豪-XiaomaGee(15959622) 22:23:22

通过不同的英文字体，表达出不同属性。

王紫豪-XiaomaGee(15959622) 22:23:40

这些都是电子工程师必备的面子活工作。

王紫豪-XiaomaGee(15959622) 22:24:41



王紫豪-XiaomaGee(15959622) 22:26:12

图就贴到这里，总结一下，大家如果要学习 X-GUI 里的字体驱动，只需要学习 font 文件夹的一些源文件。

王紫豪-XiaomaGee(15959622) 22:26:37

font.c 是驱动的中间层，负责写液晶描点，以及实现一些逻辑的。

王紫豪-XiaomaGee(15959622) 22:27:12

而以每个字体命名的 c 文件，是他们各自的取点阵动作。也就是字体驱动

王紫豪-XiaomaGee(15959622) 22:27:29

讲完了！