

## Make a new larger font for Waveshare SPI e-Paper



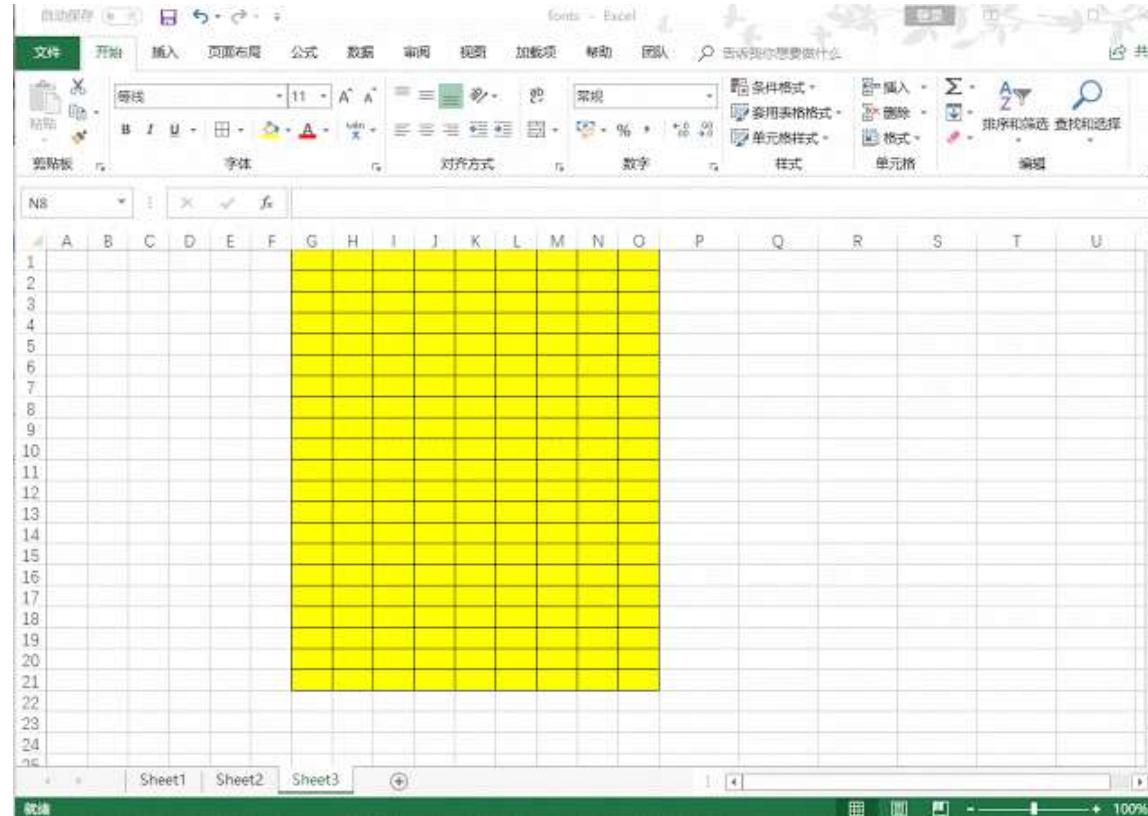
八月 23, 2018

There are five fonts (font8, font12, font16, font20 and font24) provided for e-Paper. Some users just want to make their customized fonts or bigger font, but have no idea about how to do it.

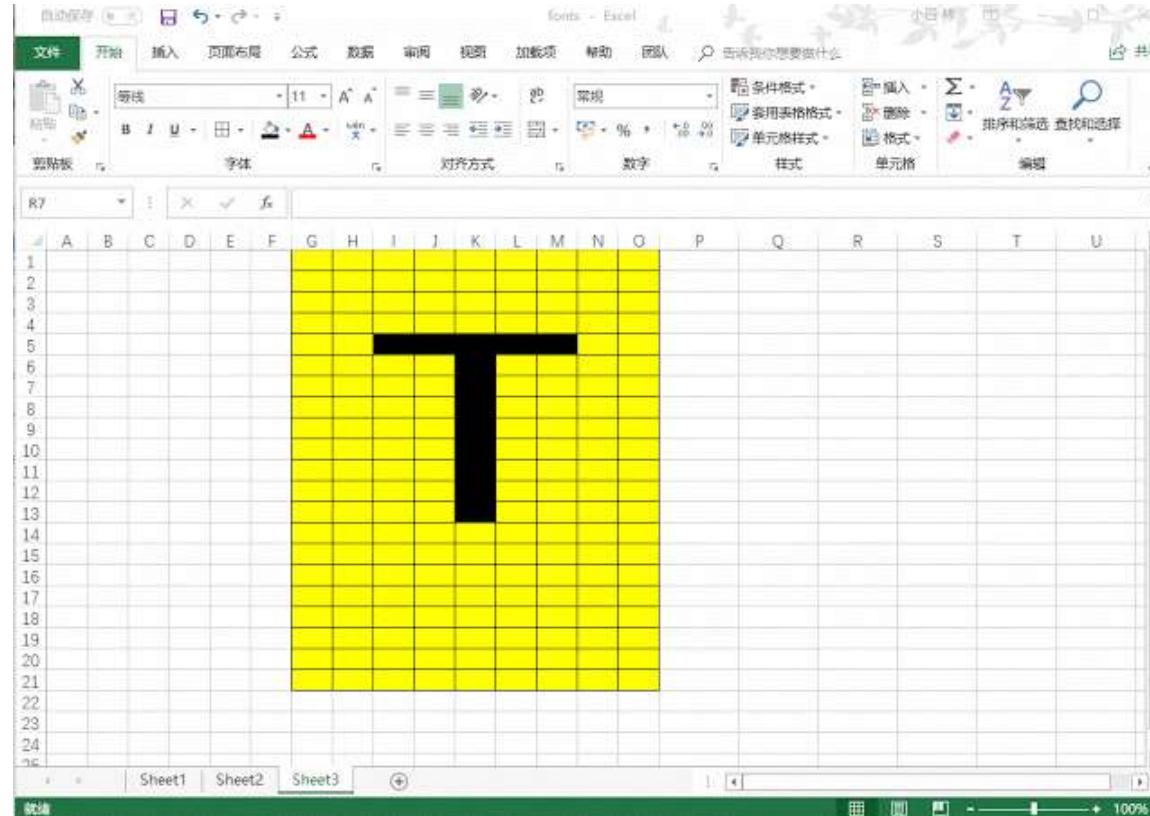
This Blog, I will tell you how the provided fonts works and how to make a new bigger font.

### How do the fonts provided works?

To draw a character on dots matrix screen, it works like to draw many dots on screen and make it look like a character. Just like this: we draw every "pixel" to white (for visibility I set it Yellow) or black, and finally we get the character "T".



As you can see, we set the paint area to 9(wide) x 20 (wide). White is 0 and black is 1, then we can get a set of binary value for every raw, convert them to HEX format, we can get the value as the fonts file. Note that one HEX value stand for 8bits, if the last bits are less than 8, we could extend it. We say it is scanning from left to right and top to bottom:



Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

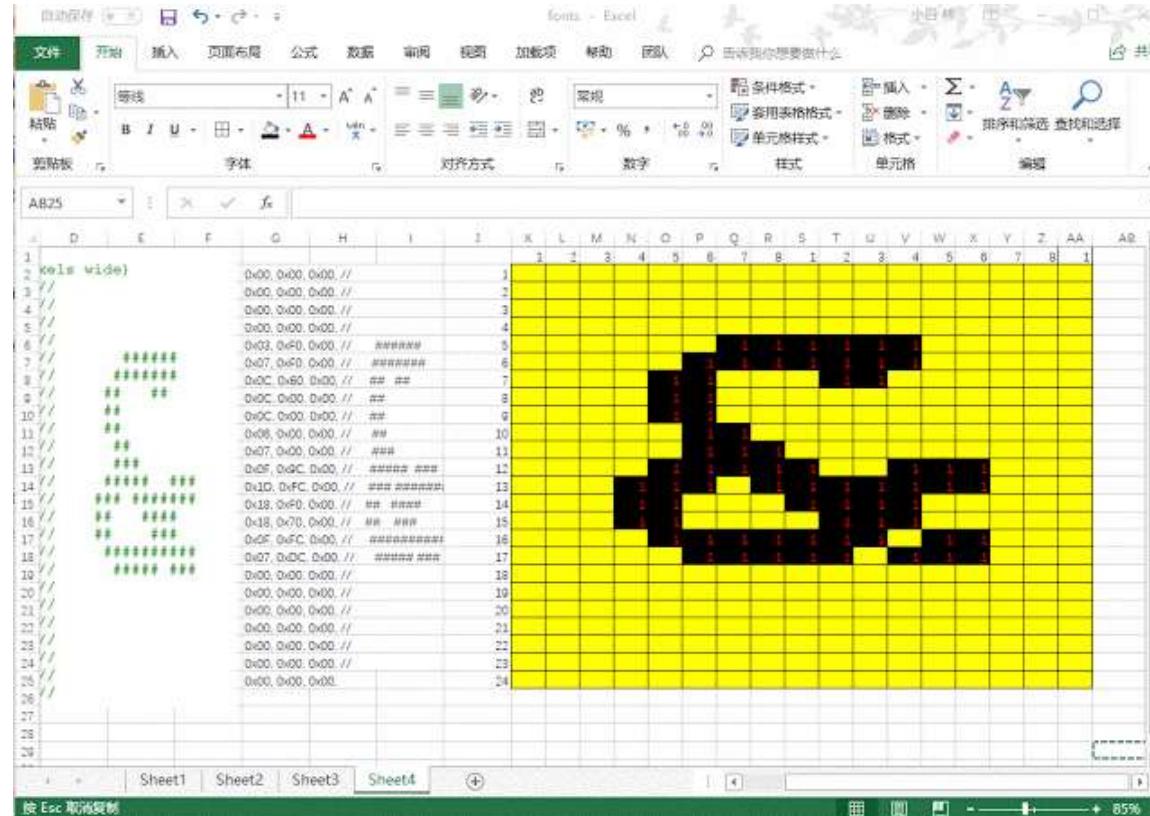
[MÁS INFORMACIÓN](#) [ACEPTAR](#)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
2						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
3						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
4						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
5						0	0	1	1	1	1	1	1	0	0011 1110	0	0x3E	0x00		
6						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
7						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
8						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
9						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
10						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
11						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
12						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
13						0	0	0	0	0	0	0	0	0	0000 1000	0	0x08	0x00		
14						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
15						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
16						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
17						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
18						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
19						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
20						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
21						0	0	0	0	0	0	0	0	0	0000 0000	0	0x00	0x00		
22																				
23																				
24																				

We take the font24 as example. The fonts we used are created by MCD Application Team. Its scanning way is from left to right and from top to bottom as well.

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)



## Make new font

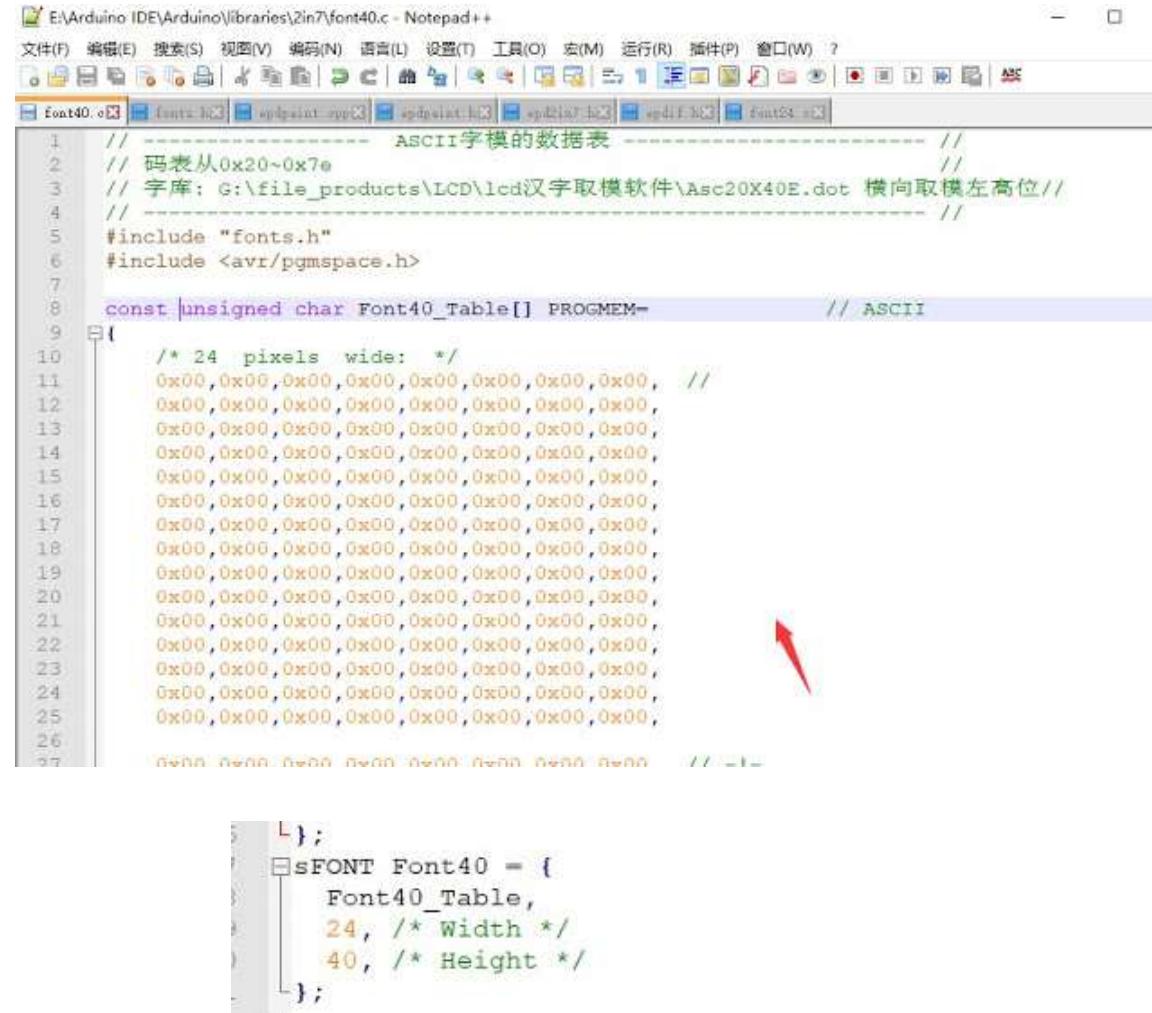
In this way, we can make a larger font manually, or use tool. Like this one, as the software only supports Chinese, I don't provide it. The character I made is 24 wide and 40 height.



The next step we need to add the font40.c to libraries. We name it Font40\_Table[ ] and add the Font40 structure

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)



```
1 // ASCII字模的数据表 //  
2 // 码表从0x20~0x7e //  
3 // 字库: G:\file_products\LCD\lcd汉字取模软件\Asc20X40E.dot 横向取模左高位//  
4 //  
5 #include "fonts.h"  
6 #include <avr/pgmspace.h>  
7  
8 const unsigned char Font40_Table[] PROGMEM = // ASCII  
9 {  
10     /* 24 pixels wide: */  
11     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, //  
12     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
13     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
14     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
15     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
16     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
17     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
18     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
19     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
20     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
21     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
22     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
23     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
24     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
25     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
26     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,  
27     0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00 } // ...  
};  
sFONT Font40 = {  
    Font40_Table,  
    24, /* Width */  
    40, /* Height */  
};
```

Add it to heard file and don't forget to update the max height/width font value

The screenshot shows a code editor with multiple tabs at the top: Font40.c, fonts.h, spdpaint.cpp, spdpaint.h, spdpaint.hx, spdi2.hx, and Font24.c. The fonts.h tab is active, displaying C-style code. A red box highlights the defines for MAX\_HEIGHT\_FONT (40/24) and MAX\_WIDTH\_FONT (24/17). Another red box highlights the sFONT struct definition. A red arrow points from the line 'extern sFONT Font40;' to the sFONT struct definition.

```
38  /* Define to prevent recursive inclusion */
39  ifndef __FONTS_H
40  define __FONTS_H
41
42  /* Max size of bitmap will based on a font24 (17x24) */
43  define MAX_HEIGHT_FONT      40/24
44  define MAX_WIDTH_FONT      24/17
45  define OFFSET_BITMAP        54
46
47  ifdef __cplusplus
48  extern "C" {
49  endif
50
51  /* Includes -----
52  #include <stdint.h>
53
54  typedef struct _tFont
55  {
56      const uint8_t *table;
57      uint16_t Width;
58      uint16_t Height;
59
60  } sFONT;
61  extern sFONT Font40; ←
62  extern sFONT Font24;
63  extern sFONT Font20;
64  extern sFONT Font16;
65  extern sFONT Font12;
66  extern sFONT Font8;
67
68  ifdef __cplusplus
69  }
```

Then we could use it on the code.

```
epd2in7-demo | imagedata.cpp | imagedata.h
```

```
46 epd.ClearFrame();
47
48 /**
49 * Due to RAM not enough in Arduino UNO, a frame buffer is not allowed.
50 * In this case, a smaller image buffer is allocated and you have to
51 * update a partial display several times.
52 * 1 byte = 8 pixels, therefore you have to set 8*H pixels at a time.
53 */
54 unsigned char image[1024];
55 Paint paint(image, 48, 168); //width should be the multiple of 8
56 paint.SetRotate(1);
57
58 // paint.Clear(UNCOLORED);
59 // paint.DrawString(0, 0, "e-Paper Demo", &Font16, COLORED);
60 // epd.TransmitPartialData(paint.GetImage(), 32, 0, paint.GetWidth(), paint.GetHeight());
61
62 paint.Clear(COLORED);
63 paint.DrawString(8, 2, "FONT40", &Font40, UNCOLORED);
64 epd.TransmitPartialData(paint.GetImage(), 80, 40, paint.GetWidth(), paint.GetHeight());
65
66
67 /* This displays the data from the SRAM in e-Paper module */
68 epd.DisplayFrame();
```

## Effect

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)



### Draw char function

Some users want to make fonts for special character or other language. The most information things which they

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)

0	NUT	32	(space)	64	@	96	'
1	SOH	33	!	65	A	97	a
2	STX	34	"	66	B	98	b
3	ETX	35	#	67	C	99	c
4	EOT	36	\$	68	D	100	d
5	ENQ	37	%	69	E	101	e
6	ACK	38	&	70	F	102	f
7	BEL	39	,	71	G	103	g
8	BS	40	(	72	H	104	h
9	HT	41	)	73	I	105	i
10	LF	42	*	74	J	106	j
11	VT	43	+	75	K	107	k
12	FF	44	,	76	L	108	l
13	CR	45	-	77	M	109	m
14	SO	46	.	78	N	110	n
15	SI	47	/	79	O	111	o
16	DLE	48	0	80	P	112	p
17	DC1	49	1	81	Q	113	q
18	DC2	50	2	82	R	114	r
19	DC3	51	3	83	S	115	s
20	DC4	52	4	84	T	116	t
21	NAK	53	5	85	U	117	u
22	SYN	54	6	86	V	118	v
23	TB	55	7	87	W	119	w
24	CAN	56	8	88	X	120	x
25	EM	57	9	89	Y	121	y
26	SUB	58	:	90	Z	122	z
27	ESC	59	;	91	[	123	{
28	FS	60	<	92	/	124	
29	GS	61	=	93	]	125	}
30	RS	62	>	94	^	126	`
31	US	63	?	95		127	DFI

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)

We can see, we get the offset value of the characters by minus **character '' (space)\* the numbers of HEX data per character.**

```
/**  
 * @brief: this draws a character on the frame buffer but not refresh  
 */  
void Paint::DrawCharAt(int x, int y, char ascii_char, sFONT* font, int colored) {  
    int i, j;  
    unsigned int char_offset = (ascii_char - ' ') * font->Height * (font->Width / 8 + (font->Width * 8 ? 1 : 0));  
    const unsigned char* ptr = &font->table[char_offset];  
  
    for (j = 0; j < font->Height; j++) {  
        for (i = 0; i < font->Width; i++) {  
            if (pgm_read_byte(ptr) & (0x80 >> (i % 8))) {  
                DrawPixel(x + i, y + j, colored);  
            }  
            if (i % 8 == 7) {  
                ptr++;  
            }  
            if (font->Width * 8 != 0) {  
                ptr++;  
            }  
        }  
    }  
}
```

For example, if we want to draw a char A, we know that its ASCII value is 65 according to the above table. And in the fonts the first character is '' (space) which ASCII value is 32. We get the offset 33 (65-32). And we know that, the font40 wide 24 and height 40, we can get that  $24 \times 40 = 960$  bits for every character, and because we save it in HEX format, so we can get that, for every character, it has  $960/8 = 120$  HEX values. With this we can get the address of first data of character "A" is  $33 \times 120 = 3960$ . So Font40\_Table[3960] is the first HEX data of character "A". That is how the function working with the fonts.

If you want to make a new fonts for other characters which are not inclued in the ASCII table, you need to change this function to get the right values.



THRISHOOL GUDELLI 2018年11月20日上午2:46

hello i have created single letter as you did but i am not getting the letter .send me the solution to  
thrishulgudelli@gmail.com

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)



输入您的评论...

此博客中的热门博文

## Raspberry Pi drive 1.44inch LCD HAT with fbtft

[三月 30, 2018](#)

The hardware:



[Raspberry Pi 3B 1.44inch LCD HAT](#) The software:

...

[阅读全文](#)

## Re-Compile Raspbian Kernel for Touch Rotation of Waveshare HDMI LCD

[三月 02, 2018](#)



Several months ago, if users of Waveshare HDMI LCD ask about how could I rotate the LCD not only the display but as well as the touch. We always said that you need to re-compile the kernel but we have no instruction about it. And two months ago we provided another method that modi

...

[阅读全文](#)

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

[MÁS INFORMACIÓN](#) [ACEPTAR](#)



网页浏览总次数

11,434

搜索此博客

搜索此博客



WAVESHARE ELECTRONICS

G+ Seguir 2

访问个人资料

归档



举报滥用情况

Este sitio utiliza cookies de Google para prestar sus servicios y para analizar su tráfico. Tu dirección IP y user-agent se comparten con Google, junto con las métricas de rendimiento y de seguridad, para garantizar la calidad del servicio, generar estadísticas de uso y detectar y solucionar abusos.

MÁS INFORMACIÓN ACEPTAR