

1. 首先请找到你所要显示的字符在 unicode 中的编码范围 <https://www.unicode.org/charts/>
例如:

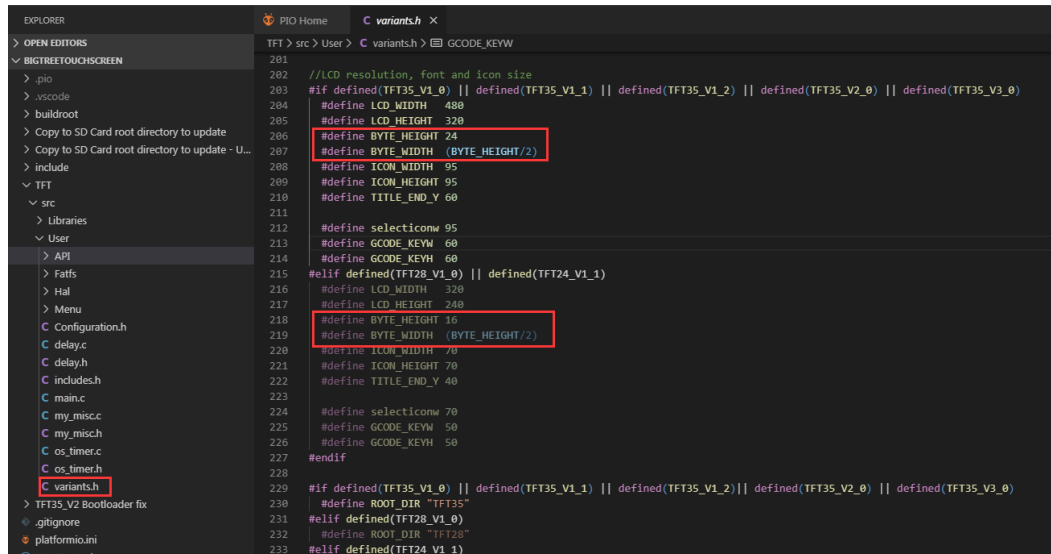
Basic Latin(ASCII) 是 0x00~0x7F <https://www.unicode.org/charts/PDF/U0000.pdf>

Cyrillic(Russia) 是 0x400~0x4FF <https://www.unicode.org/charts/PDF/U0400.pdf>

Armenian 是 0x530~0x58F <https://www.unicode.org/charts/PDF/U0530.pdf>

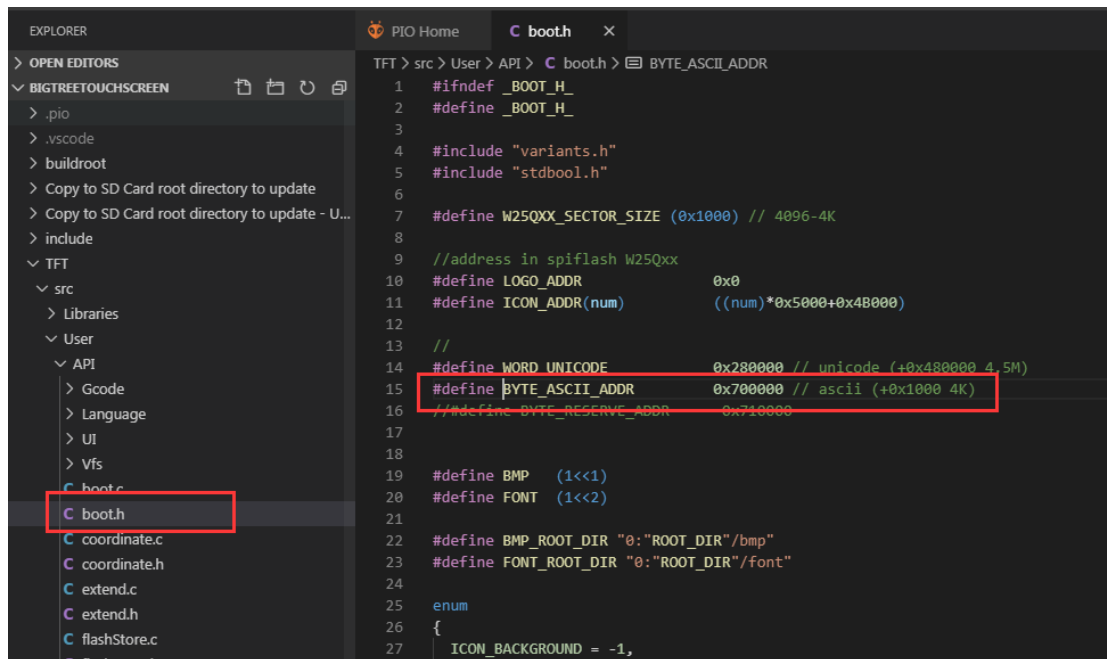
Latin extended (Czech/French/etc...)

2. 我们以 ASCII 为例, 你需要先生成一个 ASCII 所有字符的点阵字体, 点阵字体的扫描方式要求先从上到下, 再从左到右, 高位在前。字体的默认尺寸高*宽为 24*12(TFT35), 16*8(TFT24/TFT28), 可以在 variants.h 中修改字体的尺寸



```
201
202 //LCD resolution, font and icon size
203 #if defined(TFT35_V1_0) || defined(TFT35_V1_1) || defined(TFT35_V1_2) || defined(TFT35_V2_0) || defined(TFT35_V3_0)
204 #define LCD_WIDTH 480
205 #define LCD_HEIGHT 320
206 #define BYTE_HEIGHT 24
207 #define BYTE_WIDTH (BYTE_HEIGHT/2)
208 #define ICON_WIDTH 95
209 #define ICON_HEIGHT 95
210 #define TITLE_END_Y 60
211
212 #define selecticonw 95
213 #define GCODE_KEYW 60
214 #define GCODE_KEYH 60
215 #elif defined(TFT28_V1_0) || defined(TFT24_V1_1)
216 #define LCD_WIDTH 320
217 #define LCD_HEIGHT 240
218 #define BYTE_HEIGHT 16
219 #define BYTE_WIDTH (BYTE_HEIGHT/2)
220 #define ICON_WIDTH 78
221 #define ICON_HEIGHT 70
222 #define TITLE_END_Y 40
223
224 #define selecticonw 70
225 #define GCODE_KEYW 50
226 #define GCODE_KEYH 50
227 #endif
228
229 #if defined(TFT35_V1_0) || defined(TFT35_V1_1) || defined(TFT35_V1_2) || defined(TFT35_V2_0) || defined(TFT35_V3_0)
230 #define ROOT_DIR "TFT35"
231 #elif defined(TFT28_V1_0)
232 #define ROOT_DIR "TFT28"
233 #elif defined(TFT24_V1_1)
```

3. 在 boot.h 中设置点阵字体存放在 SPI Flash 中的起始地址(注意字体文件的总大小, 不要跟其他的字体地址有重叠, 且 Flash 的总容量为 8MByte, 结束地址为 0x800000)



```
1 #ifndef _BOOT_H_
2 #define _BOOT_H_
3
4 #include "variants.h"
5 #include "stdbool.h"
6
7 #define W25QXX_SECTOR_SIZE (0x1000) // 4096-4K
8
9 //address in spiflash W25Qxx
10 #define LOGO_ADDR 0x0
11 #define ICON_ADDR(num) ((num)*0x5000+0x4B000)
12
13 //
14 #define WORD_UNICODE 0x280000 // unicode (+0x480000 4.5M)
15 #define BYTE_ASCII_ADDR 0x700000 // ascii (+0x1000 4K)
16 // #define BYTE_RESERVE_ADDR 0x710000
17
18 #define BMP (1<<1)
19 #define FONT (1<<2)
20
21 #define BMP_ROOT_DIR "0:"ROOT_DIR"/bmp"
22 #define FONT_ROOT_DIR "0:"ROOT_DIR"/font"
23
24 enum
25 {
26
27 ICON_BACKGROUND = -1,
```

4. 在 boot.c 中添加将字体更新到 SPI Flash 中的功能

```
171  
172     f_close(&myfp);  
173     free(tempbuf);  
174 }  
175  
176  
177 void scanUpdates(void)  
178 {  
179     volatile u8 result = 0; //must volatile!  
180     if(mountSDCard())  
181     {  
182         result = scanUpdateFile();  
183         if (result & FONT)  
184         {  
185             updateFont(FONT_ROOT_DIR"/byte_ascii.fon", BYTE_ASCII_ADDR);  
186             updateFont(FONT_ROOT_DIR"/word_unicode.fon", WORD_UNICODE);  
187         }  
188         if (result & BMP) //bmp  
189         {  
190             updateIcon();  
191         }  
192         if (result) f_rename(ROOT_DIR, ROOT_DIR".CUR");  
193     }  
194 }  
195
```

5. 在 utf8_decode.c 文件的 static FONT_BITMAP font[] 数组中，添加待解析的字符编码，需要添加的信息如下

```
1 #ifndef UTF8_DECODE_H  
2 #define UTF8_DECODE_H  
3  
4 #include "stdint.h"  
5  
6 typedef struct {  
7     uint32_t startCodePoint; // start unicode code point for language 0x00  
8     uint32_t endCodePoint; // end unicode code point for language 0x7F  
9     uint8_t pixelHeight; // font display pixel height 24/16  
10    uint8_t pixelWidth; // font display pixel width 12/8  
11    uint32_t bitMapStartAddr; // dot matrix font library start address in w25qxx BYTE_ASCII_ADDR  
12    uint8_t bitMapHeight; // dot matrix font library pixel height 24/16  
13    uint8_t bitMapWidth; // dot matrix font library pixel width 12/8  
14    uint32_t bitMapStartCodePoint; // the first character code point in this font bitmap file 0x00  
15 }FONT_BITMAP;  
16  
17 typedef struct  
18 {  
19     // encode info  
20     uint8_t bytes; // Number of bytes occupied by one character  
21     uint32_t codePoint; // Actual encoding index of characters  
22     // font info  
23     uint8_t pixelHeight; // The pixel height of a character display  
24     uint8_t pixelWidth; // The pixel width of a character display  
25     uint32_t bitMapAddr; // the address of font bitmap in w25qxx  
26 }CHAR_INFO;  
27  
28 void getCharacterInfo(const uint8_t *ch, CHAR_INFO *pinfo);  
29 uint16_t GUI_StrPixelWidth(const uint8_t *const str);  
30  
31 #endif  
32  
33
```

```
1 #include "utf8_decode.h"  
2 #include "includes.h"  
3  
4  
5  
6 static FONT_BITMAP font[] = {  
7     // Visible ASCII code, from ' ' to '~'  
8     // start unicode code point for language  
9     0x00, // 0x00 means the first control character 'NULL'  
10    // end unicode code point for language  
11    0x7F, // 0x7F means the last control character 'DEL'  
12    // font display pixel height  
13    24,  
14    // font display pixel width  
15    12,  
16    // dot matrix font library start address in w25qxx  
17    BYTE_ASCII_ADDR,  
18    // dot matrix font library pixel height  
19    24,  
20    // dot matrix font library pixel width  
21    12,  
22    // the first character code point in this font bitmap file  
23    0x00, // the first character in BYTE_ASCII_ADDR is 0x00('NULL')  
24 },  
25 { // Czech(Latin 1 Supplement, Extended-A&B)  
26     0x80,  
27     0x24F,  
28     BYTE_HEIGHT,  
29     BYTE_WIDTH,  
30     WORD_UNICODE,  
31     BYTE_HEIGHT,  
32     BYTE_WIDTH * 2, // default "word_unicode.fon" dot matrix library font size is 24*24 / 16*16  
33     0x0, // the first character in WORD_UNICODE is 0x0000  
34 },  
35
```

6. 编译生成并更新新的固件，将字体文件的名称修改为固件 boot.c 中设置的名称“byte_ascii.fon”，放入 SD 卡的“TFT35(TFT28、TFT24)/font”文件夹中，将 SD 卡插入触摸屏的卡槽，复位更新字体文件，更新完成后在设置中切换到你所使用的语言即可享用你自定义的字体。

