

1. First, find the encoding range of the character you want to display in unicode

<https://www.unicode.org/charts/>

For example

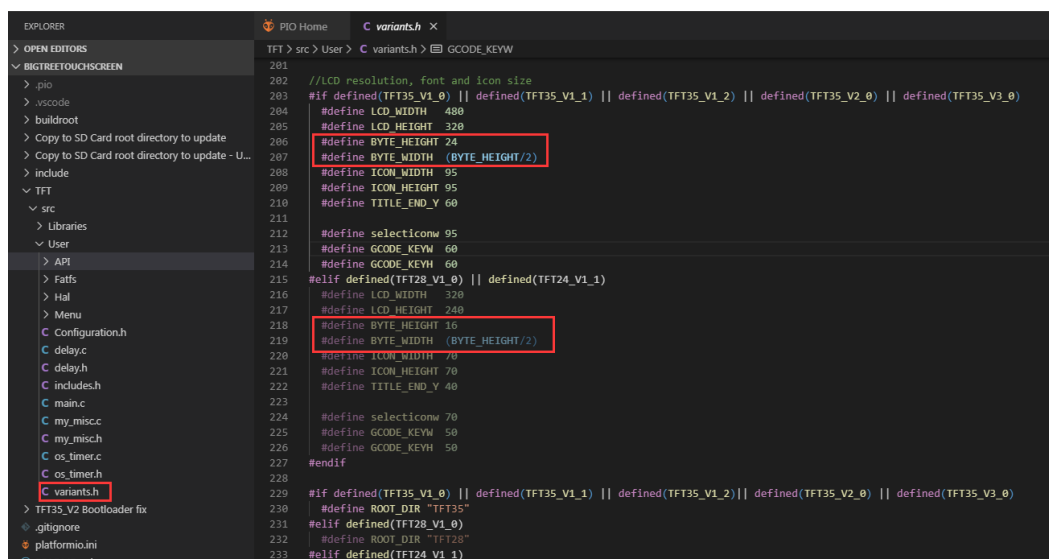
Basic Latin(ASCII) is 0x00~0x7F <https://www.unicode.org/charts/PDF/U0000.pdf>

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

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

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

2. Let's take ASCII as an example. You need to create a dot matrix font with all characters in ASCII. The scanning method of dot matrix fonts requires first from top to bottom, then from left to right, and the high position is first. The default size of the font is high\*width 24\*12 (TFT35), 16\*8 (TFT24/TFT28), and the font size can be modified in variants.h



```
281 //LCD resolution, font and icon size
282
283 #if defined(TFT35_V1_0) || defined(TFT35_V1_1) || defined(TFT35_V1_2) || defined(TFT35_V2_0) || defined(TFT35_V3_0)
284 #define LCD_WIDTH 480
285 #define LCD_HEIGHT 320
286 #define BYTE_HEIGHT 24
287 #define BYTE_WIDTH (BYTE_HEIGHT/2)
288 #define ICON_WIDTH 95
289 #define ICON_HEIGHT 95
290 #define TITLE_END_Y 60
291
292 #define selecticow 95
293 #define GCODE_KEYW 60
294 #define GCODE_KEYH 60
295 #elif defined(TFT28_V1_0) || defined(TFT24_V1_1)
296 #define LCD_WIDTH 320
297 #define LCD_HEIGHT 240
298 #define BYTE_HEIGHT 16
299 #define BYTE_WIDTH (BYTE_HEIGHT/2)
300 #define ICON_WIDTH 70
301 #define ICON_HEIGHT 70
302 #define TITLE_END_Y 40
303
304 #define selecticow 70
305 #define GCODE_KEYW 50
306 #define GCODE_KEYH 50
307 #endif
308
309 #if defined(TFT35_V1_0) || defined(TFT35_V1_1) || defined(TFT35_V1_2) || defined(TFT35_V2_0) || defined(TFT35_V3_0)
310 #define ROOT_DIR "TFT35"
311 #elif defined(TFT28_V1_0)
312 #define ROOT_DIR "TFT28"
313 #elif defined(TFT24_V1_1)
314 #define ROOT_DIR "TFT24_V1_1"
```

3. Set the dot matrix font in boot.h to store the starting address in SPI Flash (note the total size of the font file, do not overlap with other font addresses, and the total capacity of Flash is 8MByte, the ending address is 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
19 #define BMP (1<<1)
20 #define FONT (1<<2)
21
22 #define BMP_ROOT_DIR "0:"ROOT_DIR"/bmp"
23 #define FONT_ROOT_DIR "0:"ROOT_DIR"/font"
24
25 enum
26 {
27     ICON_BACKGROUND = -1,

```

4. Add the ability to update fonts to SPI Flash in boot.c

```

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. In the static FONT\_BITMAP font[] array of the utf8\_decode.c file, add the character encoding to be parsed. The information to be added is as follows

```

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 #endif
31
32
33

```

```

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

```

6. Compile, generate and update new firmware, change the name of the font file to the name "byte\_ascii.fon" set in the firmware boot.c, put it in the "TFT35 (TFT28, TFT24)/font" folder of the SD card, and then put the SD Insert the card into the card slot of the touch screen, reset the font file, then switch to the language you want in the settings to use your customized font.

