

例程 二十三 中英文在 TLT 液晶的显示

说到中英文的显示，实际上就是对中英文字符进行取模，你在液晶显示一个中英文字符的时候，就是按照那个字符的写法，相对应的点就在液晶上面的显示。下面先讲讲英文的显示方法，相对中文来说，英文显示就简单一点。因为英文就是那 26 个字母，还有一些标点符号而已。我们可以把字母和标点符号取模放在一个数组里面即可。而这些字模的数组是当你购买液晶的时候别人会给你的，所以就拿来用就是。

下面介绍几个重要的函数：

```

/*****
函数名：ASCII码子程序
输入参数：u8 ASCII 输入的ASCII码，如'A'
           BaseAddr 基址 即ASCII显示代码在FLASH中的启示位置
返回值：无
说明：输入一个ASCII码，取得它在字符字模数组中的16Byte显示代码
      并将其存放到一个16byte的ASCII显示缓冲CharBuffer[]中
*****/
void GetASCIICode(u8* pBuffer,u8 ASCII)
{
    u8 i;
    for(i=0;i<16;i++)
        *(pBuffer+i)=*(AsciiLib + (ASCII - 32)*16 + i);
    /* Offset = (*pAscii - 32)*16;*/
}

```

这个函数就是从字符字模的数组 AsciiLib[]的数组中取出相对应的字母的字模放在缓存 pBuffer 中。

```

/*****
函数名：Lcd写1个ASCII字符函数
入口参数：x,横向坐标，由左到右分别是0~29
           y,纵向坐标，由上到下分别为0~19
           CharColaor, 字符的颜色
           CharBackColor, 字符背景颜色
           ASCIICode, 相应字符的ASCII码
也就是说，320*240分辨率的显示屏，横向能显示30个ASCII字符，竖向能显示20行
返回值：无
说明：内部调用函数
*****/
void Lcd_WriteASCII(u8 x,u8 y,u16 CharColor,u16 CharBackColor,u8 ASCIICode)
{
    u8 RowCounter,BitCounter;
    u8 *ASCIIPointer;
    u8 ASCIIBuffer[16];

    #if (ID_AM==000) | (ID_AM==010) | (ID_AM==100) | (ID_AM==110)
        Lcd_SetBox(x*8,y*16,8,16);
    #else
        Lcd_SetBox(x*16,y*8,16,8);
    #endif

    Lcd_WR_Start();
    GetASCIICode(ASCIIBuffer,ASCIICode); //取这个字符的显示代码
    ASCIIPointer=ASCIIBuffer;
}

```

```

for (RowCounter=0; RowCounter<16; RowCounter++)
{
    for (BitCounter=0; BitCounter<8; BitCounter++)
    {
        if ((*ASCIIPointer&(0x80 >> BitCounter)) == 0x00)
        {
            //Set_Rs;
            DataToWrite(CharBackColor);
            Clr_nWr;
            Set_nWr;
        }
        else
        {
            //Set_Rs;
            DataToWrite(CharColor);
            Clr_nWr;
            Set_nWr;
        }
    }
    ASCIIPointer++;
}
// Set_Cs;
}

```

这个函数就是在液晶上任何一个地方显示一个英文或者标点符号，他的思想步骤就是这样的，先选定一块显示的地方，然后取该字符的字模，也相对应的点在 LCD 上面点亮。

```

/*****
函数名: Lcd写16*8字符串函数
功能: 向指定位置写入一个或多个字符，本函数带自动换行功能
入口参数: x, 横向坐标, 由左到右分别是0~29
           y, 纵向坐标, 由上到下分别为0~19
           CharColaor, 字符的颜色
           CharBackColor, 字符背景颜色
           *s 指向要写的字符串
返回值: 无
举例: Lcd_WriteString(1,1, GREEN, RED, "hello word")
*****/
void Lcd_WriteString(u8 x, u8 y, u16 CharColor, u16 CharBackColor, char *s)
{
    u8 databuff;
    Set_Rs;

    do
    {
        databuff=*s++;
        Lcd_WriteASCII(x, y, CharColor, CharBackColor, databuff);
    } while (*s);

    #if (ID_AM==000) | (ID_AM==010) | (ID_AM==100) | (ID_AM==110)
    if (x<29)
    {
        x++;
    }
}

```

```
        else if (y<19)
        {
            x=0;
            y++;
        }
        else
        {
            x=0;
            y=0;
        }

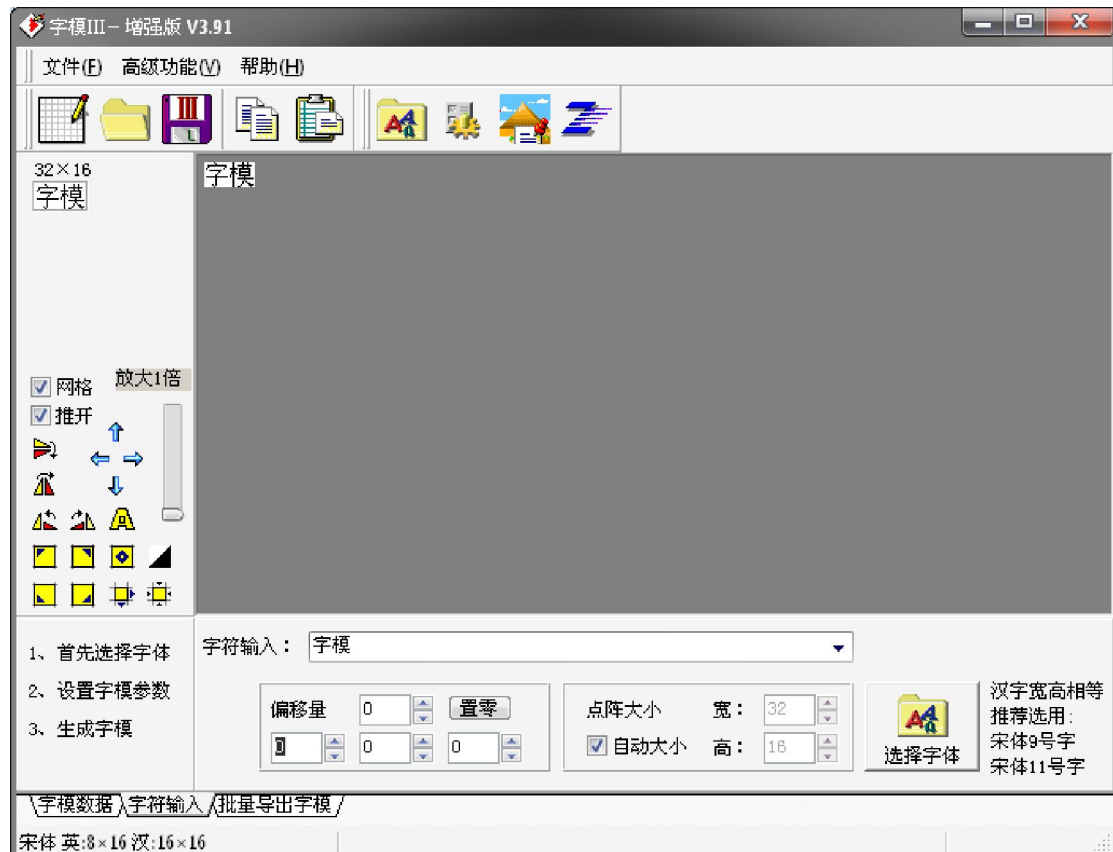
    #else
        if (y<39)
        {
            y++;
        }
        else if (x<14)
        {
            y=0;
            x++;
        }
        else
        {
            x=0;
            y=0;
        }
    #endif

}
while (*s!=0);

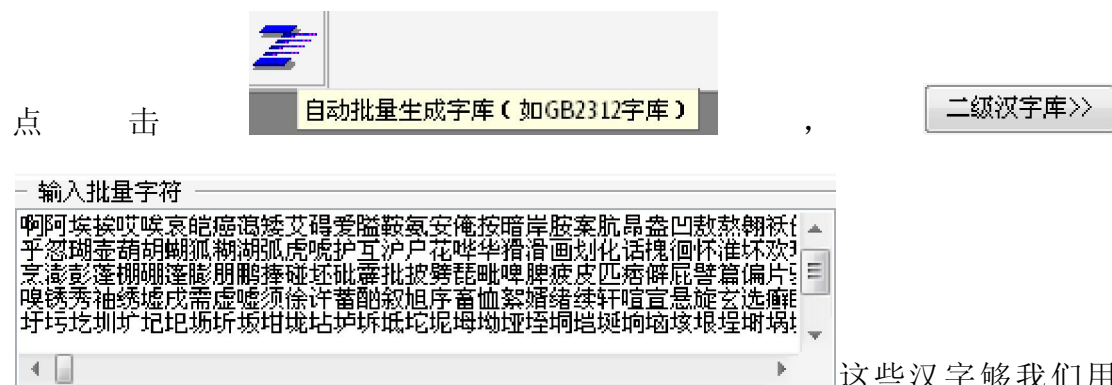
Set_Cs;

}
```

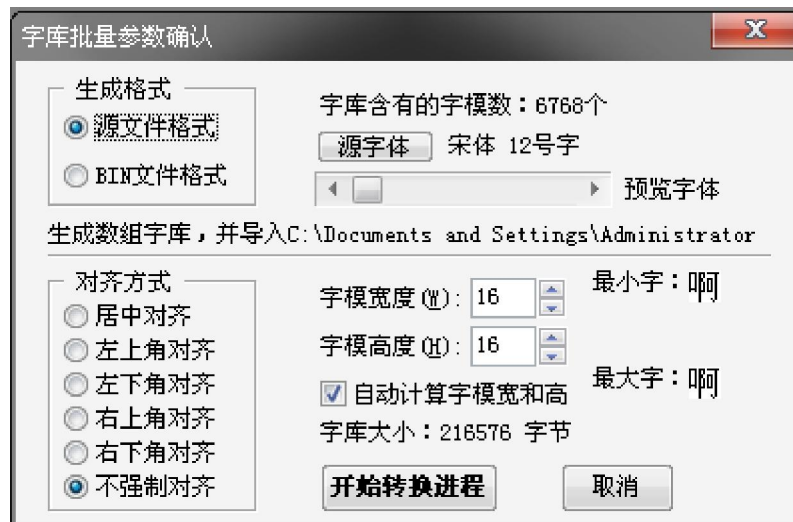
这个函数是显示一个字符串的函数，还可以自动换行。学会这 3 个函数就可以随意在 LCD 上显示英文了。这几个函数显示的英文大小是 16*8，在例程中还有 32*16 大小的英文显示函数，自己可以去看看，在这里就不多说了。英文显示还是简单些。下面讲讲中文的显示，因为我们的 LCD 是没带字库的，需要我们从 SD 卡里面读取字模。先讲讲汉字的字模的字模取。下面讲讲一个软件的使用，这个软件就是字模 3，我用的是增强版。
安装后，双击该软件



这个软件可以生成批量的汉字字库，如果不是一些怪癖的汉字的话，应该还是可以满足大家的需求的。



这些汉字够我们用了。现在我们要设置一下汉字的参数，因为我们显示出来的汉字为 16*16 的，所以我们要设置成以下：点击 **字库智能生成**，



按照这样设置好

的话，点击 **开始转换进程**，就会生成一个 xx.c 的文件

```
0x10,0x10,0x02,//宽的像素数,高的像素数, 宽的字节数, 高的字节数, 参数设置可逆
0x00,0x00,0xF7,0x7E,0x95,0x04,0x95,0x04,0x96,0x74,0x96,0x54,0x95,0x54,0x95,0x54,
0xFF,0xFF,0xF5,0x54,0x97,0x74,0x04,0x04,0x04,0x04,0x04,0x04,0x04,0x14,0x04,0x08,
};
/* 阿 */
```

这个就是阿的相应的字模。假如我们直接把这些几千个汉字的字模组成一个数组，那是编译不了了，因为我们的单片机的 RAM 没那么大，那么怎么办呢？我们刚才生成的 xx.c 文件，其实还可以生成 xx.bin 文件的。怎么设置呢？跟上面的一样，只是修改一个地方就行。



其他的跟刚才的一样,这里就不说了,BIN 文件生成是放在你安装的根目录下的。该名称是 Font.dat,你找到他,把它改为 xx.bin 文件就行。我就把它改为 HZLib.bin,然后复制到你的 SD 卡就行了。

上一讲就是讲怎样从 SD 卡里面读取系统文件了,大家务必先看看上讲的内容,要不就不知道某些语句的作用的了。下面介绍几个重要的 的函数

```

/*****
* Function Name   : GetGBKCode_from_sd
* Description     : 从SD卡字库中读取自模数据到指定的缓冲区
* Input          : pBuffer---数据保存地址
*                : c--汉字字符低字节码
* Output         : None
* Return         : 0(success)  -1(fail)
* Attention      : None
*****/
int GetGBKCode_from_sd(unsigned char* pBuffer,unsigned char * c)
{
    unsigned char High8bit,Low8bit;
    u32 pos;
    High8bit=*c; /* 取高8位数据 */
    Low8bit=*(c+1); /* 取低8位数据 */

    pos = (u32)((High8bit-0xb0)*94+Low8bit-0xa1)*2*16 ;
    f_mount(0, &myfs[0]);
    myres = f_open(&myfsrc, "0:/HZLib.bin", FA_OPEN_EXISTING | FA_READ);

    if ( myres == FR_OK )
    {
        f_lseek (&myfsrc, pos);
        myres = f_read( &myfsrc, pBuffer, 32, &mybr ); //16*16大小的汉字 其字模 占用16*2个字节

        f_close(&myfsrc);

        return 0;
    }

    else
        return -1;
}

```

这个函数是读取 SD 卡里面的 系统文件,读取汉字相对应的字模。跟其他的系统文件的读写是一样的,唯一一点不同的是有一个取地址计算公式

$pos = (u32)((High8bit-0xb0)*94+Low8bit-0xa1)*2*16$; 这个公司来源于那个软件。大家在安装字模 3 这个软件,打开安装的根目录就可以看到,我是改编过来的。能把字模取出来的话,其他的就跟英文显示的流程是一样的了。

/******

函数名：获取中文显示码子程序

输入参数：u16 ASCII 输入的中文，如“我”

返回值：无

说明：输入一个中文，取得它在字模的32Byte显示代码

并将其存放到一个32byte的显示缓冲pBuffer

*****/

```
void GetChineseCode(u8* pBuffer,unsigned char * ChineseCode)
{
    GetGBKCode_from_sd(pBuffer,ChineseCode);
}
```

/******

函数名：Lcd写1个中文函数

入口参数：x,横向坐标，由左到右分别是0~15

y,纵向坐标，由上到下分别为0~19

CharColaor,字符的颜色

CharBackColor,字符背景颜色

ASCIICode,相应中文的编码

也就是说，320240分辨率的显示屏，横向能显示15中文字符，竖向能显示20行

返回值：无

说明：内部调用函数

*****/

```
void Lcd_WriteChinese(u8 x,u8 y,u16 CharColor,u16 CharBackColor,unsigned char * Chinese)
{
    u8 ByteCounter,BitCounter;
    u8 *ChinesePointer;
    u8 ChineseBuffer[32];

    Lcd_SetBox(x*16,y*16,16,16);
    Lcd_WR_Start();
    GetChineseCode(ChineseBuffer,Chinese);
    ChinesePointer=ChineseBuffer;

    for(ByteCounter=0; ByteCounter<32; ByteCounter++)
    {
        for(BitCounter=0;BitCounter<8;BitCounter++)
        {
            if((*ChinesePointer & (0x80 >> BitCounter)) == 0x00)
            {
                DataToWrite(CharBackColor);
                Clr_nWr;
                Set_nWr;
            }
            else
            {
                DataToWrite(CharColor);
                Clr_nWr;
                Set_nWr;
            }
        }
        ChinesePointer++;
    }
}
```

```

/*****
函数名: Lcd写16*16中文函数
功能: 向指定位置写入一个或多个中文, 本函数带自动换行功能
入口参数: x, 横向坐标, 由左到右分别是0~14
           y, 纵向坐标, 由上到下分别为0~19
           CharColor, 字符的颜色
           CharBackColor, 字符背景颜色
           *Chinese 指向要写的字符串
返回值: 无
举例: Lcd_WriteChineseString(1,1, GREEN, RED, "你好")
*****/
void Lcd_WriteChineseString(u8 x, u8 y, u16 CharColor, u16 CharBackColor, unsigned char * Chinese)
{
    u8 *Databuff_Chinese;
    Databuff_Chinese=Chinese;
    Set_Rs;

    do
    {
        Lcd_WriteChinese(x,y, CharColor, CharBackColor, Databuff_Chinese);
        Databuff_Chinese+=2;
        #if (ID_AM==000) | (ID_AM==010) | (ID_AM==100) | (ID_AM==110)
        if (x<14)
        {
            x++;
        }

        }
        else if (y<19)
        {
            x=0;
            y++;
        }
        else
        {
            x=0;
            y=0;
        }
    }
    #else
    if (y<19)
    {
        y++;
    }
    else if (x<14)
    {
        y=0;
        x++;
    }
    else
    {
        x=0;
        y=0;
    }
    }
    #endif
    }while(*Databuff_Chinese!='\0');

    Set-Cs;
}

```

这些流程都是一样的, 其他的函数, 那在例程里面看看吧
下面看看主函数吧


```
int main(void)
{
    /* Infinite loop */

    /*设置内部时钟16M为主时钟*/

    CLK_HSIPrescalerConfig(CLK_PRESCALER_HSIDIV1);
    /*!<Set High speed internal clock */
    Lcd_Configuration();
    Lcd_Initialize();
    Lcd_Clear(RED);
    while(SD_CARD_USER_Init())
    {
        Lcd_Write32X16String(0,0, GREEN, RED, "SD Card Failed!");
    }
    Lcd_Write32X16String(0,0, GREEN, RED, "SD Card Checked OK!");
    Lcd_Clear(RED);
    Lcd_WriteString(0,0, GREEN, RED, "this is 16*8 size English test");
    Lcd_Write32X16String(0,1, GREEN, RED, "this is 32*16 size English test");
    Lcd_WriteChineseString(0,10, GREEN, RED, "中文显示测试成功");
    Lcd_Write32X32ChineseString(0,6, GREEN, RED, "中文显示测试成功");
    while (1)
    {
        /* 添加你的代码 */
    }
}
```

在主函数里面就是调用几个函数来测试，大家下载例程到风驰电子 STM8 开发板就会看到相对应的中英文的显示。
好了，中英文显示就讲解到这里了。

风驰电子祝您学习愉快~~~!!!!