第16章 Unicode

Unicode 编码标准是一个 16 位字符的编码规范。所有全世界的有效字符都在一个 16 位的字符集中（适用于全世界范围内使用）。Unicode 标准由 Unicode 协会定义。

μC/GUI 能够以 Unicode 编码显示单个字符或字符串，尽管最普遍是使用混合字符串，即在一个 ASCII 字符串当中有许多 Unicode 序列。

16.1 显示 Unicode 字符

Unicode 字符

µC/GUI 使用的字符输出函数（GUI_DispChar）总是处理 16 位无符号数（U16），具有显示一个由 Unicode 定义的字符的基本的能力。它仅仅需要一种包括有你想显示字符的字体。

显示 Unicode 字符串

µC/GUI 中用于字符串输出的函数通常是 GUI_DispString。既然 GUI_DispString 使用 8 位字符，你不能直接将一个 Unicode 字符串传递给它，因为 Unicode 字符串使用 16 位字符。有两个选项可利用：

使用 GUI_DispString_UC，它接受 16 位 Unicode 字符串，或者通过嵌入式 Unicode 将一个 8 位字符串转换为 2 字节的字符。这样就可以支持任何编译器支持的标准 C 字符串函数的使用。

显示混合 Unicode 和 ASCII 的代码

这是 Unicode 显示的最值得推荐的方法。你不必使用专门的函数做这项工作。只需要两个宏定义一个字符串中 Unicode 序列的起始点和结束点。

宏	说明
GUI_UC_START	标志一个 Unicode 序列的开始。
GUI_UC_END	标志一个 Unicode 序列的结束。

16.2 Unicode 和双字节转换

为什么使用双字节结构？

首先，使用 Unicode 双字节变量不是必需的。不过，这样做可以减少文字消息的内存消耗，处理字符串更容易。如果你的编译器不能直接处理 16 位字符串的话，这样做的效果确实显著。

如何构造Unicode 字符串

通常的想法是：当能够使用 Unicode 时，能够与已有的软件保持 100%的兼容性。

这意思是说，那些仍旧是字节（8 位）阵列的字符串和那些“标准”的 ASCII 或西欧字符串可以写成规则的字符串。

Unicode 部分起点由 GUI_UC_START 说明，可以在字符串中的任何位置。

GUI_UC_END 定义 Unicode 部分的终点。按惯例，“0”字符结束字符串；如果最后的字符是 Unicode 编码，GUI_UC_END 可以被忽略。

范例

```
GUI_SetBkColor(GUI_RED);
GUI_SetColor(GUI_WHITE);
GUI_Clear();
GUI_SetFont(&GUI_Font16_1HK);
GUI_DispStringHCenterAt (   “English mixed ... “
                            GUI_UC_START     /* 切换到 UNICODE（双字节） */
                            “\x30\x40”       /* 日文 */
                            “\x30\x45”       /* 日文 */
                            “\x30\x52”       /* 日文 */
                            “\x30\x51”       /* 日文 */
                            GUI_UC_END       /* 返回单字节字符 */
                            “ ..with Japanese”,160,50);
```

输出结果：

双字节结构如何工作？

所有字符，不存在 which do not have a code where 无论是 16 位字符代码的高位字节还是低位字节为 0 的情况的，either the high- or low-byte of the 16-bit character code is 0 will 将保持它们的 16 位字符编码。高字节为 0 的字符移到 0xe000～0xe0ff 区域。对于低字节代码为 0 的字符，高字节当作低字节使用，新的高字节设为 0xe1。结果的代码保存为 2 字节，高字节在前。

16.3 范例

下面的范例定义一种包括 6 个字符：“A”，“B”，“C”及 Unicode 字符 0x3060，0x3061 和 0x3062 的小字体。然后向显示屏写一个混合字符串。源代码文件是 Sample\Misc\Unicode.c。

```
/*----------------------------------------------------------------------
文件:       Unicode.c
目的:       展示 μC/GUI 的 Unicode 性能的例子
----------------------------------------------------------------------*/
#include “GUI.H”

/*********************************************************************
*                          定义包括 Unicode 字符的字体                  *
*********************************************************************/

/* unicode 区开始 <基本拉丁文> */

static const unsigned char acFontUC13_0041[13] = {  /* 代码 0041 */
_________,
_________,
___X_____,
___X_____,
__X_X____,
__X_X____,
__X_X____,
_XXXXX___,
_X___X___,
_X___X___,
XXX_XXX__,
_________,
_________};

static const unsigned char acFontUC13_0042[13] = {  /* 代码 0042 */
_________,
_________,
XXXXX____,
_X___X___,
_X___X___,
_X___X___,
_XXXX____,
_X___X___,
```

```
_X___X__,
_X___X__,
XXXXX___,
________,
________};

static const unsigned char acFontUC13_0043[13] = {  /* 代码 0043 */
________,
________,
__XX_X__,
_X__XX__,
X____X__,
X________,
X________,
X________,
X________,
_X___X__,
__XXX___,
________,
________};

/* unicode 区开始〈平假名〉*/

static const unsigned char acFontUC13_3060[26] = {  /* 代码 3060 */
___XX___,X_X_____,
____X___,_X_X____,
X___XXX_,________,
_XXXX___,________,
___X__XX,XXX_____,
___X____,X_______,
___X___X,________,
__X_____,________,
__X__X__,________,
_X___X__,________,
_X____XX,XXX_____,
________,________,
________,________};

static const unsigned char acFontUC13_3061[26] = {  /* 代码 3061 */
___XX___,________,
____X___,________,
X___XXXX,XX______,
_XXXX___,________,
___X____,________,
```

```
___X_XXX,X_______,
__XXX___,_X______,
__X_____,__X_____,
_________,__X_____,
_________,_X______,
___XXXXX,X_______,
_________,_________,
_________,_________};

static const unsigned char acFontUC13_3062[26] = {  /* 代码 3062 */
___XX___,X_X_____,
_____X___,_X_X____,
X___XXXX,X_______,
_XXXX___,_________,
___X____,_________,
___X_XXX,X_______,
__XXX___,_X______,
__X_____,__X_____,
_________,__X_____,
_________,_X______,
___XXXXX,X_______,
_________,_________,
_________,_________};

static const GUI_CHARINFO GUI_FontUC13_CharInfo[6] =
{
    {7, 7, 1, (void *)&acFontUC13_0041 },       /* 代码 0041 */
    {7, 7, 1, (void *)&acFontUC13_0042 },       /* 代码 0042 */
    {7, 7, 1, (void *)&acFontUC13_0043 },       /* 代码 0043 */
    {14, 14, 2, (void *)&acFontUC13_3060},      /* 代码 3060 */
    {14, 14, 2, (void *)&acFontUC13_3061},      /* 代码 3061 */
    {14, 14, 2, (void *)&acFontUC13_3062}       /* 代码 3062 */
};

 static const GUI_FONT_PROP GUI_FontUC13_Prop2 =

{
    0x3060 ,                                    /* 第一个字符 */
    0x3062 ,                                    /* 最后一个字符 */
    &GUI_FontUC13_CharInfo[3],                  /* 第一个字符的地址 */
    (void*)0                                    /* 下一个 GUI_FONT_PROP 的指针 */
};
```

```
static const GUI_FONT_PROP GUI_FontUC13_Prop1 =

{
    0x0041,                                 /* 第一个字符 */
    0x0043,                                 /* 最后一个字符 */
    &GUI_FontUC13_CharInfo[0],              /* 第一个字符的地址 */
    (void *)&GUI_FontUC13_Prop2             /* 下一个 GUI_FONT_PROP 的指针*/
};

static const GUI_FONT GUI_FontUC13 =

{
    GUI_FONTTYPE_PROP,                      /* 字体类型 */
    13,                                     /* 字体的调试*/
    13,                                     /* 字体 Y 轴的间距 */
    1,                                      /* X 轴的放大倍数 */
    1,                                      /* Y 轴的放大倍数 */
    (void *)&GUI_FontUC13_Prop1
};

/*********************************************************************
*                  定义包括 ASCII 和 UNICODE 字符的字符串                  *
*********************************************************************/

static const char sUC_ASCII [] =
{
    "ABC"GUI_UC_START"\x30\x60\x30\x61\x30\x62"GUI_UC_END"\x0"
};

/*********************************************************************
*                          展示 UNICODE 字符的输出                       *
*********************************************************************/

static void DemoUNICODE(void)
{
    GUI_SetFont(&GUI_Font13HB_1);
    GUI_DispStringHCenterAt("μC/GUI-sample: UNICODE characters", 160, 0);
                                            /* 设置 ShiftJIS 字体 */
    GUI_SetFont(&GUI_FontUC13);
```

```
/* 显示字符串 * /
    GUI_DispStringHCenterAt(sUC_ASCII, 160, 40);
}

/*********************************************************************
*                                主函数                              *
*********************************************************************/

void main (void)
{
    GUI_Init () ;
    DemoUNICODE();
    while(1)
        GUI_Delay(100);
}
```

上面范例执行结果的屏幕截图