

## emwin 2-day quick tutorial 014\_实体键盘消息的使用方法

因为 emWin 是通过消息和用户交互的,所以要在 emWin 中使用实体键盘,需要先熟悉 emWin 的消息和聚焦,如果不熟悉可以先看“教程 004\_控件聚焦和用户消息的使用方法”。

emWin 在执行动作引擎函数 GUI\_Exec() 过程中,会在某个地方调用获取 PID 输入设备(比如键盘、鼠标、触摸屏等等)的动作,在这个动作函数里面利用我们自己写的底层驱动程序获取到 PID 输入设备的动作、键值、坐标,比如 GLCD 例程中 glcd.c 文件下的 GLCD\_read\_exec() 函数,然后我们再将这些信息通过 emWin 的接口函数推送给 emWin 即可,emWin 内部会自己处理,接口函数有两个:触摸屏是 GUI\_TOUCH\_StoreState(), 键盘是 GUI\_StoreKeyMsg()。

对于键盘来说,emWin 除了内部处理键值之外(比如 emWin 接收到键值 0x30,会在 Edit 控件显示出“0”),还会发送一个 WM\_KEY 消息给我们,告诉我们有实体按键按下,在窗口回调函数的 WM\_KEY 消息中我们可以获取到这个键值并执行我们需要的操作。但是如果有多个窗口对应多个回调函数,那我们怎么知道 WM\_KEY 消息是发给哪个窗口呢?难不成全部窗口都接收到这个消息?emWin 不会这么傻,这样我们处理起来该多麻烦!这里就是 emWin 聚焦的重要性了。

WM\_KEY 消息只会发给当前聚焦的那个窗口,注意不是可见的那个窗口,可见窗口未必是当前激活聚焦的。利用 WM\_SetFocus() 函数可以聚焦到指定窗口,参数可以是控件的句柄,也可以是窗口的句柄,如果是窗口的句柄,那么 emWin 会自动寻找并聚焦到第一个控件上。那么就可以在这个窗口的回调函数的 WM\_KEY 消息中获取键值做我们想要做的事情了。

也许你还有一个疑问,键值是怎么回事?我该如何定义?这是完全由你自己把控的,比如你的键盘有 20 个按键,由你的单片机程序扫描出来后,哪个按键规定什么键值岂不是由你自己顺便改的?当然,规定键值的时候也需要考虑和 emWin 配合,除了标准 ASCII 字符以外(比如 0x30 对应“0”),在 emWin 的 GUI.h 中已经定义了一些特殊功能对应的键值,当然我们也可以更改 GUI.h 中的定义,但有必要吗?为什么不迁就它就行了?

```

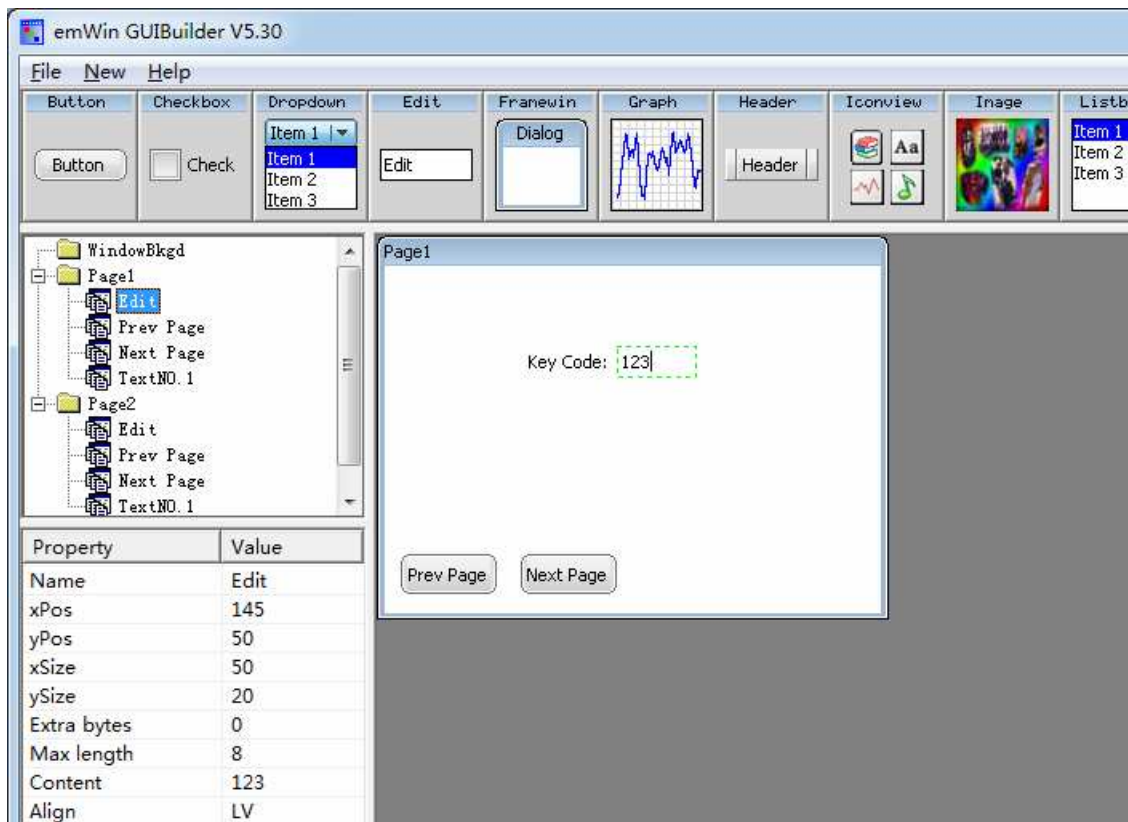
1464  */
1465  #define GUI_KEY_BACKSPACE      8      /* ASCII:
1466  #define GUI_KEY_TAB            9      /* ASCII:
1467  #define GUI_KEY_BACKTAB       10
1468  #define GUI_KEY_ENTER         13      /* ASCII:
1469  #define GUI_KEY_LEFT          16
1470  #define GUI_KEY_UP            17
1471  #define GUI_KEY_RIGHT         18
1472  #define GUI_KEY_DOWN         19
1473  #define GUI_KEY_HOME         23
1474  #define GUI_KEY_END          24
1475  #define GUI_KEY_SHIFT        25
1476  #define GUI_KEY_CONTROL      26
1477  #define GUI_KEY_ESCAPE       27      /* ASCII:
1478  #define GUI_KEY_INSERT       29
1479  #define GUI_KEY_DELETE       30
1480  #define GUI_KEY_SPACE        32
1481  #define GUI_KEY_PGUP         33
1482  #define GUI_KEY_PGDOWN       34
1483

```

另外值得注意的是,键盘扫描程序如果是用中断或另外一个独立任务实现的,那么就不能在中断或其他任务中直接调用 GUI\_TOUCH\_StoreState() 或 GUI\_StoreKeyMsg(), 这样做会出问题的!我们可以开一个缓冲区先缓存很多个键盘事件和键值,emWin 任务再查询读取这个缓冲区,但切忌这个缓冲区需要定义为 volatile 类型,因为是共享变量,包括其他共同访问的变量也需要这样做。

当你需要做按键连接时,如果因为 emWin 处理不过来而导致缓冲区满怎么办?这个问题是很容易解决的,按键按下和弹起动作是一定要写进缓冲区的,但是不是每个连接动作也都是必要的呢?在按键连接时,可以判断下缓冲区是否还有足够的空间,否则将这个连接丢弃就行了。值得注意的是,你需要预留出多点空间,以保证后面可能会出现按下和弹起动作不会因为缓冲区满而丢失。

使用 GUIBuilder V5.30 创建如下页面并保存为 c 文件, 然后添加到 CodeBlocks 工程, 其中 WindowBkgd 是作为桌面背景用的; 用 Framewin 做了两个 Page1 和 Page2 是为了说明聚焦; Edit 控件用来显示出键值, 因为 Edit 可以设置工作在文本或数值模式, 所以使用起来比较方便:



小技巧: 放置 xxxx 控件之后不要用鼠标移动, 用上下左右建移动更容易对齐(步进是 5).

备注: GUIBuilder 生成的 c 文件, 用户代码最好加在 "USER START" 和 "USER END" 之间, 其他地方除了数字以外, 不要做任何修改, 否则 GUIBuilder 将无法再次打开此 c 文件; 另外, GUIBuilder 再次打开编辑并保存时, "USER START" 和 "USER END" 之间的内容将不会被更改; 还有 c 文件不要在 GUIBuilder 打开状态直接去修改 c 代码, 否则点 GUIBuilder 保存之后修改的内容将会丢失.

