

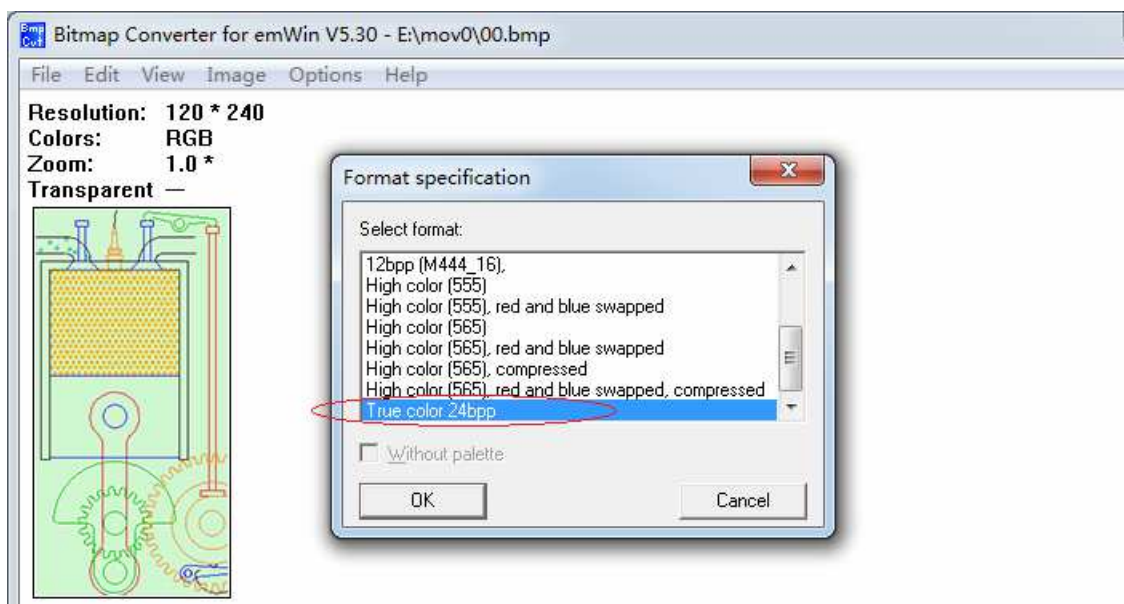
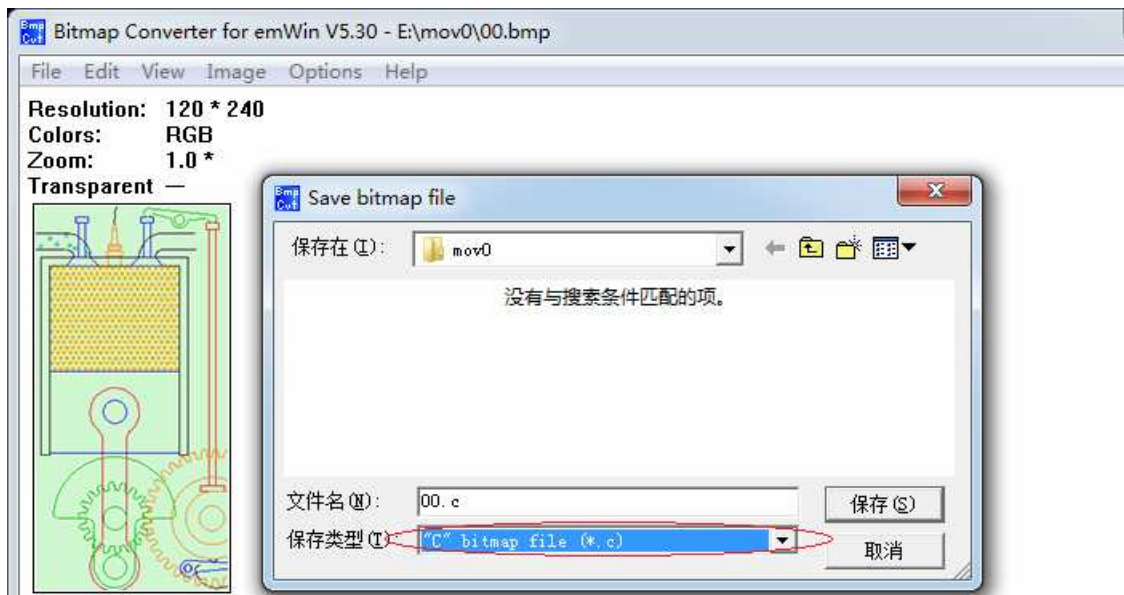
emwin 2-day quick tutorial 003_软件定时器(Timer)和图片动画

emWin 的 TIMER 是一个软件定时器,它以 OS_TimeMS 变量的值作为定时基准,而 OS_TimeMS 则通过一个硬件定时器每 1ms 中断加 1,在 emWin 内部程序不断查询 OS_TimeMS 的变化实现软件定时。

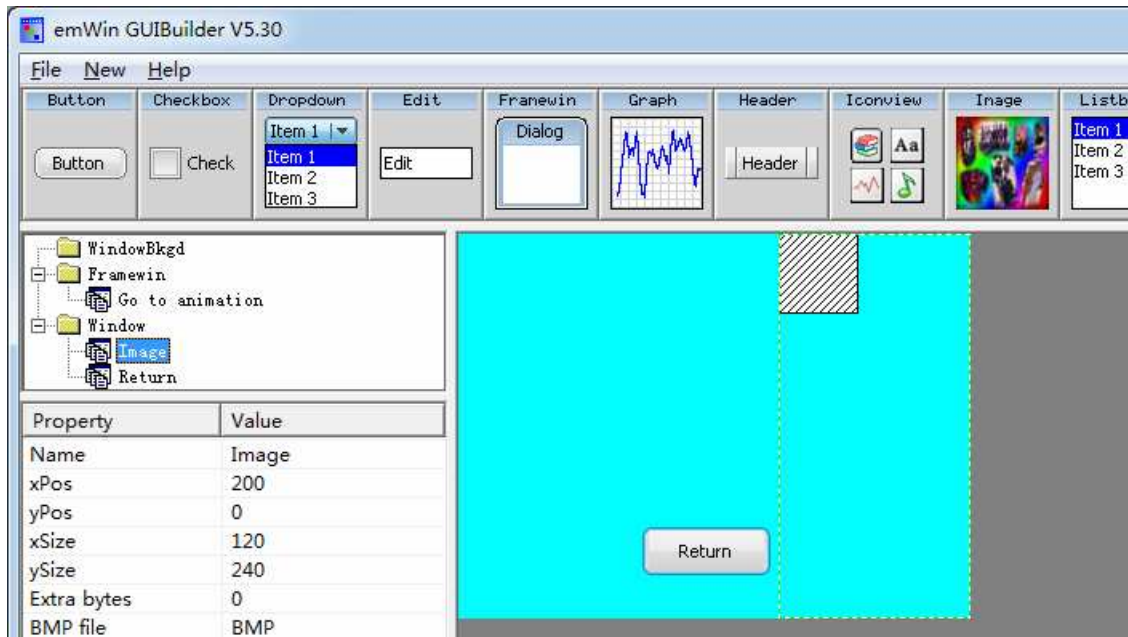
在很多应用场合,我们需要用到定时器,比如每隔一定时间查询某一事件有没有被触发,每隔一定时间改变数字显示以实现动态计数,每隔一定时间显示一幅图片实现图片动画等等...

TIMER 的使用是比较简单的(请阅读例程源码),不过有几个地方需要注意:(1)因为是通过程序查询 OS_TimeMS 的变化实现定时,因此使用 TIMER 实现定时的精度是比较差的(但 OS_TimeMS 的值是精确的),如果需要精确定时请使用硬件定时器产生中断实现。(2)用 TIMER 定时刷新数字或图片显示时,时间间隔不宜过短,比如需要实现一个毫秒级的计数器显示,你当然不能每 1ms 刷新一次数字,比较合适的是每 30ms 以上刷新一次数字显示即可(即计数器每 30ms 变化一次)。(3)TIMER 不会因为隐藏其父窗体而停止计数(当然有处理的方法,请阅读例程源码)。

利用 BmpCvt.exe 工具(在 Tool 目录下)将动画 BMP 图片转成 C 文件,然后将此 C 文件加入 emWin 工程,并每隔 100ms(WM_TIMER 消息有效时)用 IMAGE_SetBitmap() 函数将不同图片设置给 IMAGE 控件即可(emWin 内部会自己处理更新,无需我们操心):



在例程中, 其实是提供了一个 emWin 实现图片动画的模板, 用户需要做动画时只需要在此例程的基础上实现即可. 还有就是由于 emWin 的模拟器是单显存显示, 因此画面难免会有“撕裂”现象, 在双显存的 GLCD 上不会有这种现象:



小技巧: 放置 xxxx 控件之后不要用鼠标移动, 用上下左右键移动更容易对齐(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.

