

查看: 1581 | 回复: 30

Jmhh247



646 | 473 | 18

积分 | 莫元 | 飞币

中级会员

发消息

[代码分享]一个简单好用的轮询延时器模块  [复制链接]

 发表于 2017-5-23 10:08:59 | 只看该作者

1楼 电梯直达

本帖最后由 Jmhh247 于 2017-5-23 10:08 编辑

关键字: 延时器; 非阻塞自由延时

在一些按键或其它形式的命令响应中, 要求严格的话通常会需要队列缓存命令, 以防丢失。

但是我通常还会遇到一些要求非常不严格的场景, 这些场景甚至要限制响应的次数或频率(这里指不能响应太快)。比如(只是个例子), 我的产品是一个根据接收到的串口命令来工作的。串口命令发送的速度是不确定的, 有时候1秒会发送10帧命令过来, 有的时候1秒确能发送100帧, 但是这个产品每秒只打算处理20帧命令。而且由于命令有多个, 对于有些命令码可能每秒响应40帧或30帧等, 这种场景用多个延时器就很合适了。

延时器模块的接口函数一共3个, 如下:

```
01.  /* Exported functions ----- */
02.  bool zl_poll_delay_set(uint8_t chUserId, uint16_t hwTime);
03.  bool zl_poll_delay_timeout(uint8_t chUserId);
04.  void zl_poll_delay_tick(void);
复制代码
```

看下接口函数的实现, 很简单的

```
01.  /* Z L _ P O L L _ D E L A Y _ S E T  */
02.  /*-----*/
03.  功能描述 : 设定延时时间, 同时启动延时。
04.  输入参数 : uint8_t chUserId  延时器ID
05.             uint16_t hwTime  延时时间
06.  输出参数 : 无
07.  返回值 :
08.
09.  修改历史
10.  1.日期 : 2017年2月6日
11.  作者 : zys
12.  修改内容 : 新生成函数,OK。
13.
14.  -----*/
15.  bool zl_poll_delay_set(uint8_t chUserId, uint16_t hwTime)
16.  {
17.      if (chUserId >= ZL_POLL_DELAY_NUM) {
18.          return false;
19.      }
20.
21.      if (0 == hwTime) {
22.          return false;
23.      }
24.
25.      s_tPollBuf[chUserId].Time = hwTime;
26.
27.      return true;
```

```
28.
29.  }
30.
31.
32.  /*  Z L _ P O L L _ D E L A Y _ T I M E O U T  */
33.  /*-----*/
34.  功能描述   :  查询给定ID的延时器是否超时。
35.  输入参数   :  uint8_t chUserId
36.  输出参数   :  无
37.  返 回 值   :  true 超时; false 未超时。
38.
39.  修改历史
40.      1.日    期    :  2017年5月22日
41.      作    者    :  zys
42.      修改内容    :  新生成函数。
43.
44.  -----*/
45.  bool zl_poll_delay_timeout(uint8_t chUserId)
46.  {
47.      if (chUserId >= ZL_POLL_DELAY_NUM) {
48.          return false;
49.      }
50.
51.      if ( !s_tPollBuf[chUserId].Time) {
52.          return true;
53.      }
54.
55.      return false;
56.  }
57.
58.
59.  /*  Z L _ P O L L _ D E L A Y _ T I C K  */
60.  /*-----*/
61.  功能描述   :  延时模块的驱动任务，在心跳中断里运行。
62.  输入参数   :  void
63.  输出参数   :  无
64.  返 回 值   :
65.
66.  修改历史
67.      1.日    期    :  2017年5月22日
68.      作    者    :  zys
69.      修改内容    :  新生成函数。
70.
71.  -----*/
72.  void zl_poll_delay_tick(void)
73.  {
74.      uint8_t i;
75.
76.
77.      for ( i = 0 ; i < ZL_POLL_DELAY_NUM; i++ ) {
78.          if (s_tPollBuf[i].Time) {
79.              s_tPollBuf[i].Time--;
80.          }
81.      }
82.  }
83.
```

[复制代码](#)

模块的使用

使用很简单，首先把模块源文件添加到工程中，然后把

```
01.  void zl_poll_delay_tick(void)
```

[复制代码](#)

放到心跳中断里，我一般用1ms的心跳中断。

现在就能使用

```
01.  bool zl_poll_delay_set(uint8_t chUserId, uint16_t hwTime)
    复制代码
```

来设置延时器了，入参分别是延时器的ID和延时时间，延时时间单位和心跳中断有关，看自己需求了。。。

然后用

```
01.  bool zl_poll_delay_timeout(uint8_t chUserId)
    复制代码
```

来查询延时器是否超时，入参是要查询的延时器ID号。

延时器的可用最大数量通过宏定义配置

```
01.  /* 定义可用的延时数量 */
02.  #define ZL_POLL_DELAY_NUM    5
    复制代码
```

模拟一个应用场景演示代码

```
01.
02.
03.  /* 定义延时器信息，便于阅读修改 */
04.  #define DELAY_ID0          0
05.  #define DELAY_TIME0        50
06.
07.  #define DELAY_ID1          1
08.  #define DELAY_TIME1        900
09.
10.
11.
12.  /* 模拟一个应用场景 */
13.  void xxx_cmd_process(uint8_t chCmd)
14.  {
15.      switch (chCmd) {
16.          case CMD_LEFT:
17.
18.              /* 该命令执行间隔不能小于50ms */
19.              if (zl_poll_delay_timeout(DELAY_ID0)) {
20.                  zl_poll_delay_set(DELAY_ID0, DELAY_TIME0);
21.
22.                  cmd_do_left();
23.              }
24.              break;
25.
26.          case CMD_RIGHT:
27.
28.              /* 该命令执行间隔不能小于900ms */
29.              if (zl_poll_delay_timeout(DELAY_ID1)) {
30.                  zl_poll_delay_set(DELAY_ID1, DELAY_TIME1);
31.
32.                  cmd_do_left();
33.              }
34.              break;
35.
36.          case CMD_UP:
```

```
37.  
38.      /* 该命令无限制，来就执行 */  
39.      cmd_do_up();  
40.      break;  
41.  
42.      case CMD_DOWN:  
43.          break;  
44.  
45.      default:  
46.          break;  
47.  }  
48. }  
49.  
50.  
51. /* 在1ms心跳（时基）中断里运行 */  
52. void st_systick_1ms_irq(void)  
53. {  
54.     zl_poll_delay_tick ()  
55. }
```

[复制代码](#)

延时器的用处还有很多，可以灵活。。。

 [zlib poll delay模块源码.rar](#) (2.02 KB, 下载次数: 60)

#在这里快速回复#


 收藏 57

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

NM2012

 发表于 2017-5-23 12:25:45 来自手机 | 只看该作者

2楼

支持












1281 | 181 | 933  
积分 | 莫元 | 飞币

高级会员

[发消息](#)

回复

举报

<div data-bbox="100 94 151 116"><b>bitter</b></div> <div data-bbox="108 152 288 331"></div> <div data-bbox="108 360 288 414"><div>236</div><div>积分</div><div>197</div><div>莫元</div><div>78</div><div>飞币</div></div> <div data-bbox="108 439 180 461">中级会员</div> <div data-bbox="135 477 188 499"><a href="#">发消息</a></div>	<div data-bbox="349 94 722 116"> 发表于 2017-5-23 22:43:02   只看该作者</div> <div data-bbox="1449 94 1477 116">3楼</div> <div data-bbox="349 150 711 172"><p>定时时间到后自动调用回调函数，是否更方便</p></div> <div data-bbox="387 528 424 551"><a href="#">回复</a></div> <div data-bbox="1444 528 1481 551"><a href="#">举报</a></div>
<div data-bbox="100 591 183 613"><b>Jmhh247</b></div> <div data-bbox="108 649 288 884"></div> <div data-bbox="108 922 288 976"><div>646</div><div>积分</div><div>473</div><div>莫元</div><div>18</div><div>飞币</div></div> <div data-bbox="108 1001 180 1023">中级会员</div> <div data-bbox="135 1039 188 1061"><a href="#">发消息</a></div>	<div data-bbox="349 591 868 613"> 楼主   发表于 2017-5-23 23:15:17 来自手机   只看该作者</div> <div data-bbox="1449 591 1477 613">4楼</div> <div data-bbox="445 674 807 728"><p>bitter 发表于 2017-5-23 22:43 定时时间到后自动调用回调函数，是否更方便</p></div> <div data-bbox="349 786 960 808"><p>是方便，但是那样就变成定时器了。定时器是周期运行，用回调函数最合适。</p></div> <div data-bbox="349 840 852 862"><p>这里是作延时器用的，延时的处理是随机的，不适合回调函数。</p></div> <div data-bbox="349 893 584 916"><p>不知道我这样说合适不。。。</p></div> <div data-bbox="349 947 924 969"><p>我还有篇帖子是写的就是软件定时器，用回调函数的，你可以看看区别。</p></div> <div data-bbox="387 1090 424 1113"><a href="#">回复</a></div> <div data-bbox="1444 1090 1481 1113"><a href="#">举报</a></div>
<div data-bbox="100 1158 240 1180"><b>huangguimina4</b></div> <div data-bbox="108 1216 288 1391"></div> <div data-bbox="108 1420 288 1473"><div>607</div><div>积分</div><div>181</div><div>莫元</div><div>3</div><div>飞币</div></div> <div data-bbox="108 1496 180 1518">中级会员</div> <div data-bbox="135 1534 188 1556"><a href="#">发消息</a></div>	<div data-bbox="349 1158 722 1180"> 发表于 2017-5-23 23:49:47   只看该作者</div> <div data-bbox="1449 1158 1477 1180">5楼</div> <div data-bbox="349 1209 458 1232"><p>值得学习学习</p></div> <div data-bbox="387 1585 424 1608"><a href="#">回复</a></div> <div data-bbox="1444 1585 1481 1608"><a href="#">举报</a></div>
<div data-bbox="100 1650 185 1673"><b>bad_fpga</b></div> <div data-bbox="108 1709 288 1839"></div> <div data-bbox="108 1868 288 1921"><div>2218</div><div>积分</div><div>931</div><div>莫元</div><div>159</div><div>飞币</div></div> <div data-bbox="108 1944 180 1966">金牌会员</div> <div data-bbox="135 1982 188 2004"><a href="#">发消息</a></div>	<div data-bbox="349 1650 722 1673"> 发表于 2017-5-23 23:52:06   只看该作者</div> <div data-bbox="1449 1650 1477 1673">6楼</div> <div data-bbox="349 1704 458 1727"><p>先收藏再看看</p></div> <div data-bbox="387 2033 424 2056"><a href="#">回复</a></div> <div data-bbox="1444 2033 1481 2056"><a href="#">举报</a></div>
<div data-bbox="100 2098 169 2121"><b>dianzili</b></div>	<div data-bbox="349 2098 722 2121"> 发表于 2017-5-23 23:54:54   只看该作者</div> <div data-bbox="1449 2098 1477 2121">7楼</div>



50 | 26 | 9  
积分 | 莫元 | 飞币

注册会员  
发消息

943941946



689 | 373 | 90  
积分 | 莫元 | 飞币

中级会员  
发消息

Jmhh247




646 | 473 | 18  
积分 | 莫元 | 飞币

中级会员  
发消息

这种写法没用过，有机会在项目中试试

回复

举报

 发表于 2017-5-24 08:29:48 | 只看该作者

8楼


本帖最后由 943941946 于 2017-5-24 08:31 编辑

感觉不怎么好用，谢谢lz分享

恰恰用心时，恰恰无心用，用心恰恰时，恰恰心无用！

回复

举报

 楼主 | 发表于 2017-5-24 09:36:34 | 只看该作者

9楼

本帖最后由 Jmhh247 于 2017-5-24 09:38 编辑

943941946 发表于 2017-5-24 08:29

感觉不怎么好用，谢谢lz分享

哈哈，好不好用是对比才知道的，我原来的延时器是这么做的

```
01.  /* 使用标志位方式的延时器例子 */
02.
03.
04.  /* 标志，延时时间到为1，不到为0，同作为启动开关，等于0时启动延时 */
05.  uint32_t g_wDelayFlag1 = 1;
06.  uint32_t g_wDelayFlag2 = 1;
07.
08.
09.
10.  /*
11.   下面是用到延时器的服务，通过查询标志，需要说明的是，使用延时器的服务
12.   不是周期运行的，是随机发生的，否则就可以用软件定时器了。
13.  */
14.  void xxx1_service(void)
15.  {
16.      if (g_wDelayFlag1)
17.      {
18.          do_something1();
19.
20.          g_wDelayFlag1 = 0;          /* 启动延时器1 */
21.      }
22.
23.  }
24.
25.
```

```
26.
27. void xxx2_service(void)
28. {
29.     if (g_wDelayFlag2)
30.     {
31.         do_something2();
32.
33.         g_wDelayFlag2 = 0;          /* 启动延时器2 */
34.     }
35. }
36.
37.
38.
39. /* 在1ms心跳（时基）中断里运行 */
40. void st_systick_1ms_irq(void)
41. {
42.     static uint16_t s_hwCnt1 = 0;
43.     static uint16_t s_hwCnt2 = 0;
44.
45.
46.     /* 第一个延时 */
47.     if (!g_wDelayFlag1) {
48.         if (++s_hwCnt1 >= 1000) {
49.             g_wDelayFlag1 = 1;
50.             s_hwCnt1 = 0;
51.         }
52.     }
53.
54.     /* 第二个延时 */
55.     if (!g_wDelayFlag2) {
56.         if (++s_hwCnt2 >= 600) {
57.             g_wDelayFlag2 = 2;
58.             s_hwCnt2 = 0;
59.         }
60.     }
61.
62. }
63.
```

[复制代码](#)

可以看出在心跳中断里，有两段重复代码，事实上，按这种方式，需要几个延时器就必须重复几个这段代码。

而采用我帖子说的模块化后，就避免了这种重复劳动，好不好用就很明显了吧。。。

不知道大家是怎么做的。。。

修改原因：

把

/\*

下面是用到延时器的服务，通过查询标志，需要说明的是，延时器使用的服务不是周期运行的，是随机发生的，否则就可以用软件定时器了。

\*/

修改为

/\*

下面是用到延时器的服务，通过查询标志，需要说明的是，使用延时器的服务不是周期运行的，是随机发生的，否则就可以用软件定时器了。


\*/

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

[回复](#)




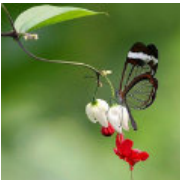

[举报](#)

meirenai

 发表于 2017-5-24 09:48:55 | 只看该作者

10楼

思路很清晰，谢谢楼主分享。

<div></div> <div><div>1128</div><div>积分</div></div> <div><div>264</div><div>莫元</div></div> <div><div>102</div><div>飞币</div></div> <div>高级会员</div> <div>发消息</div>	<div>回复</div> <div>举报</div>
<div>kinsno</div>	<div>发表于 2017-5-24 10:56:25   只看该作者</div> <div>11楼</div>
<div></div> <div><div>5208</div><div>积分</div></div> <div><div>1995</div><div>莫元</div></div> <div><div>3438</div><div>飞币</div></div> <div>金牌会员</div> <div>发消息</div>	<div>回复</div> <div>举报</div>
<div>TBG3</div>	<div>发表于 2017-5-24 11:31:40   只看该作者</div> <div>12楼</div>
<div></div> <div><div>1148</div><div>积分</div></div> <div><div>332</div><div>莫元</div></div> <div><div>0</div><div>飞币</div></div> <div>高级会员</div> <div>发消息</div>	<div>回复</div> <div>举报</div>
<div>hy317</div>	<div>发表于 2017-5-24 13:48:23   只看该作者</div> <div>13楼</div>
<div></div> <div><div>818</div><div>积分</div></div> <div><div>802</div><div>莫元</div></div> <div><div>2201</div><div>飞币</div></div> <div>高级会员</div> <div>发消息</div>	<div>回复</div> <div>举报</div>
<div>airwolf09921</div>	<div>发表于 2017-5-25 17:05:19 来自手机   只看该作者</div> <div>14楼</div>
<div></div> <div><div>1152</div><div>积分</div></div> <div><div>996</div><div>莫元</div></div> <div><div>3105</div><div>飞币</div></div>	



积分 | 莫元 | 飞币

高级会员

发消息

miscell



337 | 304 | 30

积分 | 莫元 | 飞币

中级会员

发消息

Jmhh247



646 | 473 | 18

积分 | 莫元 | 飞币

中级会员

发消息

fm007



304 | 255 | 9

积分 | 莫元 | 飞币

中级会员

发消息

istars2005

回复

举报

15楼

发表于 2017-5-25 22:54:07 | 只看该作者

如果没启动，其计数值也为0，不是返回成功的吗

回复

举报

16楼

楼主 | 发表于 2017-5-26 08:54:18 | 只看该作者

miscell 发表于 2017-5-25 22:54

如果没启动，其计数值也为0，不是返回成功的吗

是的，就是这样设计的。因为上电后是第一次执行，所以有没有延时，结果都是一样的。

所以，在我写的用例中，都是先查询，超时后再设定延时的

```
01.  /* 该命令执行间隔不能小于900ms */
02.      if (z1_poll_delay_timeout(DELAY_ID1)) {
03.          z1_poll_delay_set(DELAY_ID1, DELAY_TIME1);
04.
05.          cmd_do_left();
06.      }
```

复制代码

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

17楼

发表于 2017-7-19 16:30:04 | 只看该作者

感谢楼主分享，我目前也使用过这种思路的，但是用的是离散的结构体标志

回复

举报

18楼

发表于 2017-7-19 21:35:43 | 只看该作者

本帖最后由 istars2005 于 2017-7-19 21:37 编辑

楼主的方式貌似有点繁琐



1301 积分 | 662 莫元 | 0 飞币

高级会员

发消息

```
01.
02.  vu32 *gTimOut[TM_OUT_CNT] = {0};    // 超时计数变量指针数组
03.  static u8  tm_Out_Cnt = 0;            // 实际超时变量个数
复制代码
```

```
01.
02.  /**
03.   * 函数功能：配置超时指针
04.   * 输入参数：*ptout 需要传入的超时变量指针
05.   * 输出参数：配置结果 0:成功  -1:索引溢出  1:超时指针重复(不影响正常结果)
06.   * 功能说明：修改原来实现方式的bug,不再使用idx变量
07.               当超过最大允许的超时变量个数时返回-1
08.   */
09.  s8 TimeOutSet(vu32 *ptout)
10.  {
11.      if(tm_Out_Cnt >= TM_OUT_CNT)
12.          return -1;
13.      for(int i=0;i<tm_Out_Cnt;i++) {
14.          if(ptout == gTimOut[i])
15.              return 1;
16.      }
17.      gTimOut[tm_Out_Cnt] = ptout;
18.      tm_Out_Cnt++;
19.      return 0;
20.  }
复制代码
```

在系统滴答定时器中调用如下代码

```
01.
02.     int i=0;
03.     while(i<tm_Out_Cnt) {                // 扫描超时变量,并且把不为0
04.         的变量进行自减一                  的变量进行自减一
05.         *gTimOut[i] ? (*gTimOut[i])-- : 0;
06.         i++;
复制代码
```

使用方法

```
01.  vu32 _gTimeOut1 = 0;                    // 全局变量
02.  TimeOutSet(&_gTimeOut1 );              // 超时变量注册,最好注册一次,重复注册也可以使用,但是稍浪费一
03.                                          点时间
04.  //正式使用
05.  if(_gTimeOut1 == 0) {
06.      _gTimeOut1 = 50;
07.
08.      // DoSomething();
09.  }
复制代码
```

回复

举报

jssd

发表于 2017-7-19 22:28:31 来自手机 | 只看该作者

19楼

感谢分享



2262  
积分

371  
莫元

177  
飞币

金牌会员

发消息

Jmhh247



646  
积分

473  
莫元

18  
飞币

中级会员

发消息

Jmhh247



646  
积分

473  
莫元

18  
飞币

中级会员

发消息

istars2005



1301  
积分

662  
莫元


0  
飞币

高级会员

发消息

回复

举报

 楼主 | 发表于 2017-7-20 09:05:48 | 只看该作者

20楼

istars2005 发表于 2017-7-19 21:35

楼主的方式貌似有点繁琐

哈哈，你的方式很不错哈，简单说，你的是将数据与算法分离了

对于这种起辅助作用的功能模块，我现在的思维是尽可能的多做封装，不去管理数据，而是管理 ID，让使用者更关注自己的代码实现。

现在很火的共享单车(辅助模块)，对用户来说，有很多车摆在那里，根本无需申请一辆车(声明数据)，直接扫码(管理ID)骑走。


繁琐与否，就看个人理解吧。。。

我另外一个帖子，也是基于这样的考虑  
点我 [\[代码分享\]一个软件定时器模块，简单好用](#)

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

 楼主 | 发表于 2017-7-20 09:07:08 | 只看该作者

21楼

fm007 发表于 2017-7-19 16:30


感谢楼主分享，我目前也使用过这种思路的，但是用的是离散的结构体标志

结构体标志也不错

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

 发表于 2017-7-20 09:20:25 | 只看该作者

22楼

Jmhh247 发表于 2017-7-20 09:05

哈哈，你的方式很不错哈，简单说，你的是将数据与算法分离了

对于这种起辅助作用的功能模块，我现在的思 ...

我也是考虑到封装问题，把这部分功能放到了STM32的SysTick处理的模块当中，只提供一个延时变量注册的接口。

这样使用时只需注册一下自己模块内部的静态变量后面的操作都是本地变量的赋值和判断。

自我感觉封装的还不错 😊

https://www.amobbs.com/thread-5674774-1-1.html

11/17

Jmhh247



646

473

18

积分

莫元

飞币

中级会员

发消息

istars2005



1301

662

0

积分

莫元

飞币

高级会员

发消息

jiangyimfs1



5

5

0

积分

莫元

飞币

新注册者

发消息

编辑原因: 标题错误

回复

举报

 楼主 | 发表于 2017-7-20 10:36:21 | 只看该作者

23楼

istars2005 发表于 2017-7-20 09:20

我也是考虑到封装问题,  
把这部分功能放到了STM32的SysTick处理的模块当中,  
只提供一个延时变量注册的接口 ...

你这么说我就看懂了, 很有参考价值哈

用户处理好超时变量一次注册的问题就行了。

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

 发表于 2017-7-20 10:47:55 | 只看该作者

24楼

Jmhh247 发表于 2017-7-20 10:36

你这么说我就看懂了, 很有参考价值哈

用户处理好超时变量一次注册的问题就行了。

是的

开始时候是只允许注册一次,重复注册会多占用空间,而且重复意义不大  
后来就改了一下  
重复注册的话不会多占用空间了  
只是需要浪费几个时钟周期去检查一下

为了不占太多空间我一般配置不超过16个超时变量  
更改一下宏定义即可

回复

举报

 发表于 昨天 08:23 | 只看该作者

25楼

楼主的非阻塞延时器很实用, 在工程中使用了一下, 出了点问题, 请教一下楼主是不是哪个步骤使用错误了, 贴上代码

```
01. void Task_Thread(void)
02. {
03.     switch(step)
04.     {
05.
06.         case 0: //两个LED全灭
07.
08.         //
09.             if (z1_poll_delay_timeout(0)) //DELAY_ID1 0 标志位
10.             {
11.
12.                 z1_poll_delay_set(0, 4000); // 延时4000ms, 4s
13.
14.                 LED2=1;
15.
16.                 step=1;
17.             }
18.             break;
19.
20.
21.
22.         case 1: // led1亮
```

```
23.
24.
25.             LED1=0;
26.             if (z1_poll_delay_timeout(1))           //DELAY_ID1  1   标志位
27.                 {
28.
29.             z1_poll_delay_set(1, 4000);           //    延时4000ms, 4s
30.
31.
32.
33.                 break;
34.
35.
36.             case 2:           // led2亮
37.
38.
39.             if (z1_poll_delay_timeout(2))           //DELAY_ID1  0   标志位
40.                 {
41.
42.             z1_poll_delay_set(2, 4000);           //    延时4000ms, 4s
43.
44.
45.
46.
47.                 break;
48.
49.
50.
51.
52.
53.                 }
54. }
```

[复制代码](#)

```
//定时器3中断服务程序 1ms
void TIM3_IRQHandler(void) //TIM3中断
{
    u8 Index;

    z1_poll_delay_tick();
}
```

[回复](#)

[举报](#)

Jmhh247

 楼主 | 发表于 昨天 09:08 | 只看该作者

26楼



646 | 473 | 18  
积分 | 莫元 | 飞币

中级会员

[发消息](#)

[jiangyimfs1](#) 发表于 2017-8-22 08:23  
楼主的非阻塞延时器很实用，在工程中使用了一下，出了点问题，请教一下楼主是不是哪个步骤使用错误了，贴上 ...

呃，大哥，你得告诉我出了啥问题吧，光贴代码不知道你想表达啥问题。。。另，你的代码缩进好难受

只能让我猜猜你所谓的问题吧

你想要实现的是不是：  
>状态0：两个LED全灭，延时4S  
>状态1：LED1亮，延时4S  
>状态2：LED2亮，延时4S，LED2亮起的同时切换到状态0

如果是这样，你的写法确实存在些问题，用一个延时器就可以实现了。。。

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

Jmhh247

楼主 | 发表于 昨天 09:29 | 只看该作者

27楼



646 473 18  
积分 莫元 飞币

中级会员  
发消息

jiangyimfs1 发表于 2017-8-22 08:23

楼主的非阻塞延时器很实用，在工程中使用了一下，出了点问题，请教一下楼主是不是哪个步骤使用错误了，贴上 ...

按我猜测的功能写的，不过代码未测试，只是展示延时器的使用。。。

```
01. #define LED_DELAY_TIME 4000
02.
03. /* L E D _ D E M O _ T A S K */
04. /*-----
05.      *功能：延时器测试闪灯，在超级循环中运行。
06.      *参数：无
07.      *返回：无
08.      *备注：未测试。
09. -----*/
10. void led_demo_task(void)
11. {
12.     static uint8_t s_chStep = 0;
13.
14.
15.     switch (s_chStep)
16.     {
17.         case 0:
18.             /* 全灭 */
19.             /* 第一次用直接就是超时状态，因为还未设置超时时间 */
20.             if (z1_poll_delay_timeout(0)) {
21.                 z1_poll_delay_set(0, LED_DELAY_TIME);
22.
23.                 LED_1 = 1;
24.                 LED_2 = 1;
25.
26.                 s_chStep = 1;
27.             }
28.
29.             break;
30.
31.         case 1:
32.             /* LED1 亮 */
33.             if (z1_poll_delay_timeout(0)) {
34.                 z1_poll_delay_set(0, LED_DELAY_TIME);
35.
36.                 LED_1 = 0;
37.
38.                 s_chStep = 2;
39.             }
40.
41.             break;
42.
43.         case 2:
44.             /* LED2 亮 */
45.             if (z1_poll_delay_timeout(0)) {
46.                 z1_poll_delay_set(0, LED_DELAY_TIME);
47.
48.                 LED_2 = 0;
49.
50.                 s_chStep = 0;
51.             }
52.             break;
53.
54.         default:
```

```
55.             break;
56.         }
57.     }
58.
```

复制代码

在每个case里设置的延时，是为下个case服务的。

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报


 楼主 | 发表于 昨天 09:32 | 只看该作者

28楼

我去，我这个代码的缩进也不对，居然变长了。。。

回复

举报

 发表于 昨天 10:05 | 只看该作者

29楼

@楼主： 假如有大量的定时要处理，中断处理会消耗不少时间，会降低系统实时性。  
建议设计成一个查询函数，可以被调用，系统时钟可以查询定时器得到，这样可以不用中断函数了。

回复

举报

 楼主 | 发表于 昨天 14:01 | 只看该作者

30楼

gtnr 发表于 2017-8-22 10:05

@楼主： 假如有大量的定时要处理，中断处理会消耗不少时间，会降低系统实时性。  
建议设计成一个查询函数， ...

你说的对，量大消耗的时间也多

就是设计查询函数，我不太明白，是基于心跳计数查询吗

18楼的方法我觉得也不错。

用来分享的才是快乐。<http://www.amobbs.com/static/image/smiley/coolmonkey/07.gif>

回复

举报

 发表于 昨天 15:12 | 只看该作者

31楼

Jmhh247



646 | 473 | 18  
积分 | 莫元 | 飞币

中级会员

发消息

gtnr



979 | 546 | 159  
积分 | 莫元 | 飞币

高级会员

发消息

Jmhh247



646 | 473 | 18  
积分 | 莫元 | 飞币

中级会员

发消息

jiangyimfs1

本帖最后由 jiangyimfs1 于 2017-8-22 16:02 编辑

Jmhh247 发表于 2017-8-22 09:08

呃，大哥，你得告诉我出了啥问题吧，光贴代码不知道你想表达啥问题。。。另，你的代码缩进好难受

对的，代码就是实现这样的效果了，代码我直接复制黏贴了，没注意看显示的效果了，不好意思，看了一下楼主的代码，是我自己理解错误了，按楼主的代码试了一下，效果很好，非常感谢了。。。

再问一下楼主，我**case0** 延时**3s**，**case1** 延时**2s**，**case2** 延时**1s**，是下面这么写吗，现象是**LED1**，**LED2**同时亮了，我是不是又哪里用错了

```
#define DELAY_ID0      0
#define DELAY_TIME0    1000

#define DELAY_ID1      1
#define DELAY_TIME1    2000

#define DELAY_ID2      2
#define DELAY_TIME2    3000

/* LED_DEMO_TASK */
/*-----
*功能：延时器测试闪灯，在超级循环中运行。
*参数：无
*返回：无
*备注：未测试。
-----*/

void led_demo_task(void)
{
    static uint8_t s_chStep = 0;

    switch (s_chStep)
    {
        case 0:
            /* 全灭 */
            /* 第一次用直接就是超时状态，因为还未设置超时时间 */
            if (zl_poll_delay_timeout(DELAY_ID2)) {
                zl_poll_delay_set(DELAY_ID2, DELAY_TIME2); //延时3s

                LED1 = 1;
                LED2 = 1;

                s_chStep = 1;
            }

            break;

        case 1:
            /* LED1 亮 */
            if (zl_poll_delay_timeout(DELAY_ID1)) {
                zl_poll_delay_set(DELAY_ID1, DELAY_TIME1); //延时2s

                LED1 = 0;

                s_chStep = 2;
            }

            break;

        case 2:
            /* LED2 亮 */
            if (zl_poll_delay_timeout(DELAY_ID0)) {
                zl_poll_delay_set(DELAY_ID0, DELAY_TIME0); //1s

                LED2 = 0;

                s_chStep = 0;
            }

            break;

        default:
```



5 | 5 | 0  
积分 | 莫元 | 飞币

新注册者  
发消息



```
break;
    }
}
```

回复

举报

发帖 ▾

回复

返回列表



友情提示：**标题不合格、重复发帖，将会被封锁ID**。详情请参考：论坛通告：封锁ID、获得注册邀请码、恢复被封ID、投诉必读

高级模式



发表回复

☐ 回帖后跳转到最后一页

up↑



本版积分规则