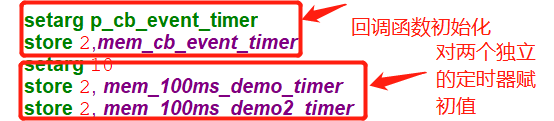
常用的定时器有两种：100ms定时器和0.625ms定时器。这两种定时器都是依靠蓝牙时钟计时，但是对定时器的查询时轮询的。

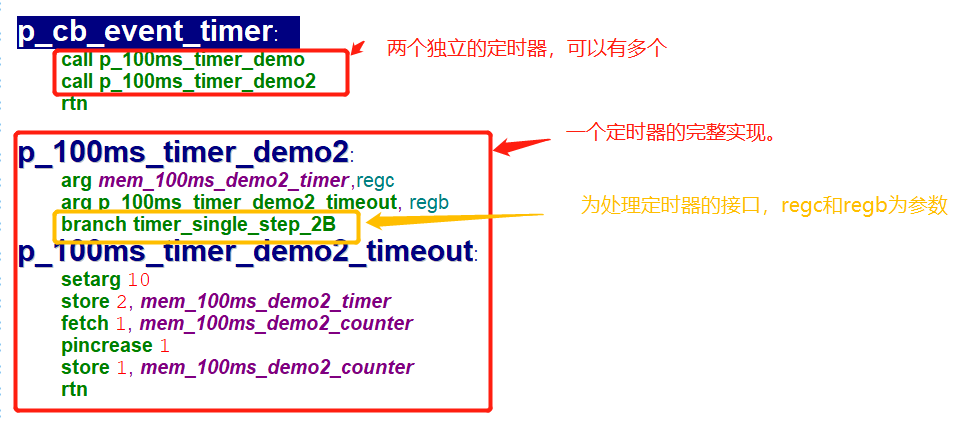
一、100ms定时器：

使用100ms定时器，需要定义一个回调函数（callback\_func），然后回调函数的地址赋值到变量“mem\_cb\_event\_timer”，程序每100ms访问一次callback\_func，并检查定时器是否超时。在callback\_func中可以有N个call，每一个call都是一个独立的100ms计数器，可以为每一个call申请一个变量（mem）对其进行计数。给mem赋初值x，就代表x\*100ms后，会超时，x是递减的。

示例：



*100ms timer 初始化*



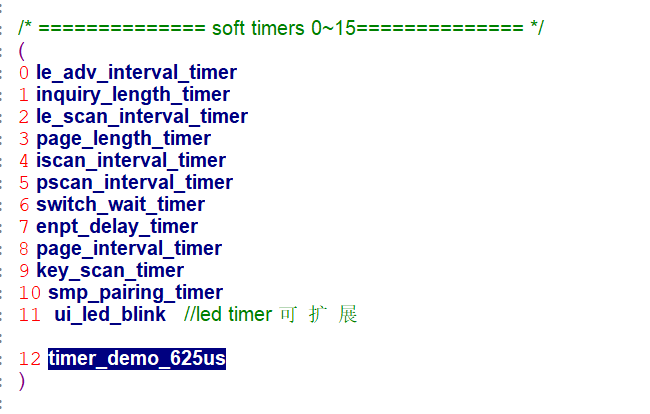
*100ms timer 函数体*

二、0.625ms定时器

使用0.625ms定时器,关键有两个函数”timer\_check”和”timer\_init”.

**timer\_init**用来初始化定时器, 调用这个函数以前要传两个寄存器参数,queue(选用哪一个定时器) 和 pdata(需要定时的时间)

参数1：queue



queue的取值可以是0~15。如上图，为了增加程序的可读性，我们最好把定时器标号以宏定义的形式，写在代码中。我们在选用定时器的时候，不要与蓝牙底层的定时器冲突即可。

参数2：pdata

根据需要给pdata赋值即可。

例如：需要定时1s, 因为1600 \* 0.625ms = 1s, 所以给pdata 赋值1600即可。

**timer\_check**用来检查初始化的时间是否跑完。

在调用timer\_check以前,需要传一个寄存器参数 queue(选用哪一个定时器), 这里要传递的这个queue参数,要与初始化的时候,传递的queue参数一致。

调用完这个函数，blank标志位的状态会改变，可以认为blank的值，就是这个函数的返回值。如果定时时间到， blank被置1；否则，blank置0。

例程： 使用0.625ms计时器，每1s，累加计数器。

