 什么时候我们需要用到SetTimer函数呢？当你需要每个一段时间执行一件事的的时候就需要使用SetTimer函数 了。使用定时器的方法比较简单，通常告诉WINDOWS一个时间间隔，然后WINDOWS以此时间间隔周期性触发程序。通常有两种方法来实现：发送WM\_TIMER消息和调用应用程序定义的回调函数。

**1.1 用WM\_TIMER来设置定时器**

先请看SetTimer这个API函数的原型

UINT\_PTR SetTimer(  
  HWND hWnd,          // 窗口句柄  
  UINT\_PTR nIDEvent,      // 定时器ID，多个定时器时，可以通过该ID判断是哪个定时器  
  UINT uElapse,         // 时间间隔,单位为毫秒  
  TIMERPROC lpTimerFunc     // 回调函数  
);

例如    
SetTimer(m\_hWnd,1,1000,NULL); //一个1秒触发一次的定时器  
在MFC程序中SetTimer被封装在CWnd类中，调用就不用指定窗口句柄了

于是SetTimer函数的原型变为：

UINT SetTimer(

  UINT nIDEvent,

  UINT nElapse,

  void(CALLBACK EXPORT \*lpfnTimer)(HWND,UINT ,YINT ,DWORD)

)

 当使用SetTimer函数的时候，就会生成一个计时器。函数中nIDEvent指的是计时器的标识 ，也就是名字。nElapse指的是时间间隔 ，也就是每隔多长时间触发一次事件。第三个参数是一个回调函数 ，在这个函数里，放入你想要做的事情的代码，你可以将它设定为NULL，也就是使用系统默认的回调函数，系统默认认的是onTime函数。 这个函数怎么生成的呢？你需要在需要计时器的类的生成onTime函数：在ClassWizard里，选择需要计时器的类，添加WM\_TIME消息映射，就自动生成onTime函数了 。然后在函数里添加代码，让代码实现功能。每隔一段时间就会自动执行一次。

例：

SetTimer(1,1000,NULL);

1:计时器的标识；

1000：时间间隔，单位是毫秒；

NULL:使用onTime函数。

当不需要计时器的时候调用KillTimer(nIDEvent);

例如：KillTimer(1);

**1.2 调用回调函数**

此方法首先写一个如下格式的回调函数(在SetTimer函数表示回调函数TIMERPROC不为空时，需要调用自定义的回调函数 )

void CALLBACK TimerProc(

  HWND hWnd,

  UINT uMsg,

  UINT idEvent,

  DWORD dwTime

);  
然后再用SetTimer(1,100,TimerProc)函数来建一个定时器，第三个参数就是回调函数地址。

**如果要加入两个或者两个以上的timer怎么办？**

继续用SetTimer函数吧，上次的timer的ID是1，这次可以是2，3，4。。。。

SetTimer(2,1000,NULL);

SetTimer(3,500,NULL);

嗯，WINDOWS会协调他们的。当然onTimer函数体也要发生变化，要在函数体内添加每一个timer的处理代码：

onTimer(nIDEvent)  

{  

switch(nIDEvent)  

{  

case 1:..;  

 break;  

case 2:..;  

 break;  

case 3:..;  

 break;  

}  

}

在使用定时器时注意SetTimer中表示定义时标识的nIDEvent参数，以及OnTimer或自定义的回调函数中nIDEvent参数，还有KillTimer中的这个参数，如果要用它们对特定的定时器进行操作时，注意保持它们的一致。

https://blog.csdn.net/cqingyu/article/details/4449593