Re: Timer with Micro seconds

Source:
http://www.tech−archive.net/Archive/WindowsCE/microsoft.public.windowsce.platbuilder/2007−11/msg00610.html

- *From*: "Helge Kruse" <Helge.Kruse−nospam@xxxxxxx>
- *Date*: Thu, 22 Nov 2007 10:14:12 +0100

In Chandra's code there is design flaw. Under some conditions the expression

liTimeOut.QuadPart += liDelay.QuadPart

can result in small value, when the addition overflows. The better way is to calculate the difference between current time and start time and compare it with the timeout.

I must admit that the probability for the overflow is small, but it's different from zero.

```
LARGE_INTEGER liDelay;

    // Query number of ticks per second
    if (QueryPerformanceFrequency(&liDelay))
    {
    // 1ms delay
    liDelay.QuadPart /= 1000;

    LARGE_INTEGER liTimeOut;
    // Get current ticks
    if (QueryPerformanceCounter(&liTimeOut))
    {
    // Create timeout value
    liTimeOut.QuadPart += liDelay.QuadPart;
    LARGE_INTEGER liCurrent;
    do
    {
    // Get current ticks
    QueryPerformanceCounter(&liCurrent);
    // Delay until timeout
    } while (liCurrent.QuadPart<liTimeOut.QuadPart);
    }
    }
```

```
LARGE_INTEGER liStart;
// Get current ticks
if (QueryPerformanceCounter(&liStart))
{
do
{
LARGE_INTEGER liCurrent;
LARGE_INTEGER liSpan;

// Get current ticks
QueryPerformanceCounter(&liCurrent);
liSpan.QuadPart = liCurrent.QuadPart − liStart.QuadPart;
} while (liSpan.QuadPart<liTimeOut.QuadPart);
}
```

/Helge

.