Applications of Timer of Data Sampling Software
Hongyan ZHUO*, Chengliang GE, Zhiqiang LIU, Jiaru ZHANG
Institute of Applied Electronics
CAEP, P.O. Box 919-1011
Mianyang 621999, China
*E-mail: 526756092@qq.com
** Corresponding author

Abstract—Based on periodic inspection and scanning, data sampling software usually used WM_TIMER and MM_TIMER. MM_TIMER is a multimedia timer. There are differences between WM_TIMER and MM_TIMER, such as implementation methods, PRI and precisions of time. Herein, the great differences are compared with experiments. Its precision will be reached at 1ms with applications of MM_TIMER.

Keywords— periodic inspection and scanning; precision of timer; MM_TIMER; WM_TIMER

I. INTRODUCTION

Up to today, there are two methods of data sampling system based on periodic inspection and scanning. One method is using embedded system to achieved microsecond precision by hardware clock. Another method is using A/D card to achieved millisecond precision. In the second method, most of users developed applied software based on Windows system. And for such multitask and multuser system of Windows, developers cannot operate the rock-bottom hardware. Then, developers achieved periodic data sampling by STK of applied software [1]. MM_TIMER and WM_TIMER are usually used timers of VC by developers. And there are great differences of precision between two timers. So, the methods and precisions of the two timers are presented in this paper.

II. DESIGN AND IMPLEMENTATION OF WM_TIMER

WM_TIMER is a timer which is provided by Windows. The method is easy and simple. Developer can transfer SetTimer() function to allocate a timer to application program. While a time interval is specified, Windows system sends a WM_TIMER message every specified time interval. Then the program can implement events in real-time.

The procedure of implementation is listed as initialing of SetTimer(), closing timer with KillTimer(), sending message queue, sampling data in response function of WM_TIMER.

Function of SetTimer is defined as below.

```c
void CTestView::OnButtonRun()
{
    // TODO: Add your control notification handler code here
    SetTimer(1, 20, NULL);       //sampling periods are 20ms
}
```

III. DESIGN AND IMPLEMENTATION OF MM_TIMER

MM_TIMER is a multimedia timer provided by Windows system. Within VC software, its difficult than WM_TIMER. The main procedure is to setup parameters of MM_TIMER, to transfer timeGetDevCaps() function to decide supported minimal precision and maximal precision, to startup timer to do applications, to release sources by deleting timer. The more used functions and means are listed below [2,3].

```c
//setup minimal resolution of timer
timeBeginPeriod(1); //setup minimal resolution of timer
```

The interface functions of MM_TIMER are included in DLL of mmsystem.dll. And VC++ provides response head files. The detail implementation methods are listed below.

(1)Setup resolution of Timer
MMRESULT timeGetDevCaps(lpTIMECAPS&ptc, UINTcbtc) transfers timeGetDevCaps() function to judge the minimal resolution and maximal resolution. Then it transfers timeBeginPeriod(UINTuPeriod) function to setup resolution of timer. The parameter UINTuPeriod is the value of timer resolution.

(2) Setup events of timer

MMRESULT timeSetEvent(UINT uDelay, //uDlelay: interval of sampling
UINT uResolution, // uResolution: precision of time, its default value is 1ms
LPTIMECALLBACK lpTimeProc, //lpTimeProc: user defined callback //function
DWORD dwUser, UINT fuEvent); //dwUser: user provided callback function
// UINT fuEvent: mode of event, triggered by TIMER period

(3) Declaring of whole callback function

Void CALLBACK TimerCallBackProc(UINT wTimerID, UINT mMsg, DWORD dwUser, DWORD dw1, DWORD dw2)

Wherein, wTimerID is sign of timer. mMsg is reserved. dwUser is parameter used by user. dw1 and dw2 are reserved.

(4) User defined message disposal function

The message disposal function defined by user is to receive messages from MM_TIMER.

IV. APPLICATIONS

A. Experiments

Above methods are applied in one big system. The curves of two timers are showed in figure 1 and figure 2. Figure 1 is curves of sampled data with applications of WM_TIMER. The precision of WM_TIMER is 15ms. But, there is 'Losing Second' phenomenon. The maximal losing second is about 2s-4s. As showed in figure 1, the real sampling time is 18s and figure 1 has only 16s. The 2s is lost.

Figure 2 is curves of sampled data with applications of MM_TIMER. The precision of MM_TIMER is 1ms. And, there is no 'Losing Second' phenomenon. As showed in figure 2, the real sampling time is 18s and figure 2 has data with time duration of 18s. Time is not lost.

V. CONCLUSIONS

By studying of applications, some conclusions are listed in table 1. From table 1, applications without requirements of high precision can apply WM_TIMER. And for an application with high sampling frequency and precision, the MM_TIMER is needed.

### TABLE I. COMPARISON OF TWO Timers

<table>
<thead>
<tr>
<th></th>
<th>WM_TIMER</th>
<th>MM_TIMER</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding</td>
<td>Concision and Conveniency</td>
<td>More codes</td>
</tr>
<tr>
<td>Precision</td>
<td>55ms in general</td>
<td>Maximal value is 16ms Minimal value is 1ms</td>
</tr>
<tr>
<td>PRI</td>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Sampling frequency</td>
<td>It is easy to appear 'Losing Second' while the sampling frequency is high.</td>
<td>It is not easy to appear 'Losing Second'.</td>
</tr>
</tbody>
</table>
REFERENCES


