DE GRUYTER OPEN

BULGARIAN ACADEMY OF SCIENCES

CYBERNETICS AND INFORMATION TECHNOLOGIES • Volume **17**, No 1
Sofia • 2017 Print ISSN: 1311-9702; Online ISSN: 1314-4081
DOI: 10.1515/cait-2017-0010

Data Receiving Method Based on Multimedia Timing in Real-Time System

Zhigang Ma[1], *Wenyi Liu*[2]

[1]*College of Information Science and Engineering, Shanxi Agricultural University, Taigu 030801, China*
[2]*Key Laboratory of Instrumentation Science & Dynamic Measurement, North University of China, Taiyuan 030051, China*
Emails: sxau_mzg@163.com liuwenyi@nuc.edu.cn

***Abstract*:** *Several methods, such as polling, multithread, timing, and so on, can be used in data receiving course. Low-efficiency and high level of system resource consumption may bring about data loss in polling and multithread methods when the data transmission rate is very high. Software timing methods are discussed and analyzed in Visual C*$^{++}$*. Timing method would improve the system resource availability and decrease the risk of data loss. According to the demand of a testing system, a real-time monitoring system based on memory sharing and multimedia timer is presented. After testing, the average timing error of the multimedia timer in the given instance is not bigger than 0.05%, so the continuity and integrity of data receiving can be assured under the conditions of high-speed data transmission.*

***Keywords*:** *Real-time monitoring system, data receiving, timing, multimedia timer, memory sharing.*

1. Introduction

Real-time monitoring system has been widely used in critical systems for industrial and agricultural production, environmental security, aerospace, medical, communication, defense equipment, and other domains. A large number of real-time monitoring applications include: fault diagnosis and maintenance of mechanical equipments [1], construction quality monitoring [2], meteorological and geological disaster prediction [3], oil, natural gas and other underground production environment monitoring [4], environmental monitoring and protection [5], medical care and monitoring [6], transportation [7], rocket launching parameter monitoring [8], and so on.

In order to meet the "real-time" requirements in monitoring systems, the execution efficiency of the application program must be guaranteed. The efficiency involves many aspects in the monitoring system, and relates with a variety of factors, such as: number of channels, sampling rate, data transmission rate, data

receiving method, data processing algorithm, performance of embedded processor or computer, development tool, and so on. Among of which, data receiving method will affect the performance of the whole monitoring system. Inefficient receiving algorithm may give rise to data loss, processing delay or other problems, and thereby affect the monitoring of critical parameters. For the purpose of developing a efficient and reasonable data receiving method, this paper is discussed on the basis of Visual C $^{++}$ 2008 (VC 9.0).

2. Real-time system and real-time monitoring system

In a Real-Time System (RTS), its function must be accomplished in the specified time, and give a response for external or internal, synchronous or asynchronous time. Correctness of the real-time system depends not only on the running result of the system, but also the time when the result is generated [9-12].

Real-time systems can be divided into “hard real-time” and “soft real-time”. In hard real-time systems, time requirement must be met; otherwise it will cause serious consequences, such as the application in certain key occasions of military, aerospace, nuclear industry and so on. In soft real-time systems, although the time requirement is proposed, occasionally violating the time demand will not cause serious effects in systems operation, such as widespread monitoring and control systems, information collection and processing systems. The Real-Time Monitoring System (RTMS) discussed in this paper belongs to soft one. In the soft real-time monitoring system, time demand is not very harsh, but because of plenty of channels and large amount of data, there are many difficulties in the design process.

A typical real-time monitoring system usually consists of the front-end (distal end), and one or more back-ends (proximal end), as shown on Fig. 1. Typically, the testing data is transmitted from the front-end to every back-end after data acquisition, framing and packaging. And data processing, analysis, solving and monitoring task will be completed mostly in back-ends. Therefore, the front-end and back-end are always called as acquisition module and monitoring module respectively.

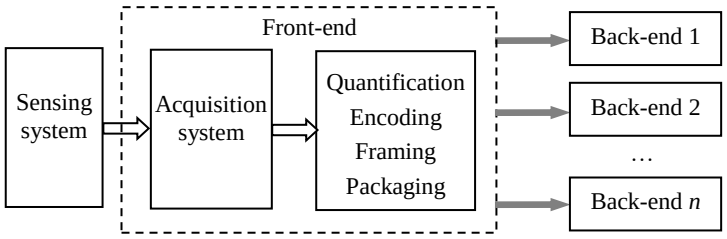

Fig. 1. Composition of real-time monitoring system

3. Data receiving method

In the real-time monitoring system, there are a large number of testing data transmitted from the front-end to the back-end. The obtained testing data must be real-time processed (analyzing, solving, curves, graphics, and so on) in the back-end, and polling, Multithreading, timing and other methods can be used for data receiving.

3.1. Polling method

Polling method is a kind of data receiving method with the easiest algorithm and program structure, which can be achieved by the loop control structure. There are some problems using this method for data receiving, mainly reflected in: because of the high occupancy rate of CPU, it is difficult to intervene in the program execution flow and likely to generate an endless loop. When developing the window application program, if the loop program segment cannot return promptly, it will prevent the continual execution of window message processing, and then lead to the operational difficulties or unresponsive in the software interface. In order to solving the above problem, the message mechanism can be adopted, that is calling the function similar with DoEvents in Visual Basic. Thus, control of CPU can be temporarily returned to the operating system, and which will avoid the endless loop. In fact, due to the characteristics of the cycle mechanism, the use of this method has a significant risk of losing data.

Here, we set two industrial PCs, named IPC-A and IPC-B. IPC-A and IPC-B were connected with a network cable. IPC-A transmitted data with the rate of 1Mbps~8 Mbps, and IPC-B was used to receive data.

Table 1 shows the relationship between data loss rate and data transmission rate. There is a need to explain, the testing is based on the following hardware and software environment: computer's hardware platform is 3.3 GHz of CPU frequency and 4 GB of memory; operating system is Windows XP SP3.

Table 1. Relationship between data loss rate and data transmission rate in polling method

Data transmission rate (bps)	1 M	2 M	4 M	5 M	8 M
Data loss rate (%)	0	0	0.05	0.26	4.18

It is thus clear that, the polling method is only suitable for the low-speed (no more than 4 Mbps) systems; for high-speed systems, this method has some limitations.

3.2. Multithreading method

Multithreading is a kind of concurrent execution technology for multiple threads achieved based on the hardware or software. Due to the hardware support, a computer with multi-threading capability is able to execute multiple threads at the same time, which can improve the overall processing performance of the computer. In essence, the use of multithreading can improve the availability factor of CPU. When there are multiple threads running in the system, the operating system will

provide a CPU time slice for every thread by polling way. Every thread will run in the corresponding time slice, because the time slice is very short, which looks the same that the multiple threads running simultaneously. Therefore, multithreading method can maximize the utilization of CPU resources.

In this paper, a real-time data receiving and processing procedure is designed base on multithreading method. There are two threads in the procedure, one is for data receiving and archiving, and the other one is used for data processing and analysis, including: channel data extraction, curve plotting, and so on. In this case, the relationship between data loss rate and data transmission rate is shown in Table 2.

Table 2. Relationship between data loss rate and data transmission rate in multithread method

Data transmission rate (bps)	5 M	6 M	8 M	10 M	12 M	15 M
Data loss rate (%)	0	0	0.09	0.21	0.86	1.19

As can be seen from Table 2, compared with the polling mode, data loss rate was reduced to a certain extent in multithreading method. However, when the data transmission rate exceeds 8 Mbps, the system still loses data.

3.3. Timing method

Timer is a commonly used data receiving method in real-time monitoring systems. Timing can be divided into hardware and software timing. Hardware timing has higher accuracy, but because of the additional circuit, the achievement of hardware timing is more complex. Timer in Windows operating system can be used for software timing, and high-precision timer can be adopted in precise timing applications. There are several ways of Implementing software timing in VC [13, 14].

(1) SetTimer function timing method

In VC programming, timing operation in multi-task system can be achieved by WM_TIMER message. When the timing task is completed, control is returned to the system in order to perform other operations. When programming, the timer number and interval are set by calling SetTimer function firstly, the former two parameters of SetTimer are timer number (ID) and timing interval (ms) respectively. For example, calling SetTimer (100, 500, NULL) will create a timer with ID of 100 and interval of 0.5 s. Then a timing response function OnTimer is programmed to complete the corresponding function.

Application of SetTimer timer is relatively simple, but its timing accuracy is low, only tens of ms. As a result, it is often used in applications with lower precision timing demand.

(2) GetTickCount function timing method

GetTickCount function can also achieve the timing operation. There is no parameter in GetTickCount, and its type of return value is DWORD, that is 32-bit unsigned integer. The execution result of GetTickCount is the elapsed time (in ms) from system startup to the time of calling this function. The timing accuracy of

GetTickCount is higher than SetTimer, about a dozen ms. Therefore, this method is also not suitable for the applications with high precision requirement.

(3) Multimedia timer

To meet the need of multimedia, Microsoft provides an application program interface (API) function for multimedia precise timing, which is timeSetEvent function. The prototype of timeSetEvent is:

MMRESULT timeSetEvent (UINT uDelay, UINT uResolution, LPTIMECALLBACK lpTimeProc, WORD dwUser, UINT fuEvent)

Where: uDelay specifies the timing cycle (ms); uResolution specifies the timing precision (ms), the smaller the value, the higher the timing resolution; LpTimeProc points to a callback function, which contains the program code need timing executed.

The timing precision of multimedia timer is about 1 ms, which is very reliable. However, the consumption of system resources is very large, so it must be released promptly after used.

(4) High-precision timer

The above-mentioned timing methods are commonly used in systems with timing precision over 1 ms. For more demanding timing or interrupt system, more precise functions QueryPerformanceFrequency and QueryPerformanceCounter are provided in VC. Among them, calling QueryPerformanceFrequency can get the frequency of high-precision timer supported by hardware, and QueryPerformanceCounter provides the current value of the high-resolution performance counter, if one exists.

When using, the computer's internal clock frequency (TimerFrequency) can be obtained by calling QueryPerformanceFrequency firstly; then calling QueryPerformanceCounter function before and after the timing event respectively, the two counts will be obtained; using the difference (CounterDiff) of the two counts and clock frequency (TimerFrequency), the event time can be calculated with CounterDiff / TimerFrequency.

Theoretically, the timing precision of high-precision timer can be achieved the magnitude of ns (10-9 s). in practical use, considering of the system hardware configuration, the influence of various factors, thread priority in operating system, timing randomness, the actual minimum resolution of high-precision timer is about 1 μs, timing precision can reach about 2 μs [14].

(5) Comparison of software timing methods

Table 3. shows the timing precision of the above-mentioned methods in Windows XP. The minimum resolution of timer created by SetTimer or GetTickCount method is from a dozen to dozens of ms, and the priority is lower in the multi-tasking operating system, so these two methods are only suitable for lower timing precision and real-time demand occasions. By contrast, the timing resolution and precision of multimedia timer is higher, which can be about 1 ms, and this method can reduce the impact to timer's running from the constraints of system

resource. The timing interval of high-precision timer can be set as any time above 2 μs, which is suitable for the systems with higher precision requirements.

Table 3. Comparisons of software timing methods

Timing method	Timing precision	System priority	Application
SetTimer	dozens of ms	lower	lower real-time demand
GetTickCount	a dozen ms	lower	lower real-time demand
Multimedia	1 ms	higher	higher real-time demand
High-precision	2 μs	higher	higher real-time demand

4. Application instance and analysis

For the requirement of a testing application, a real-time monitoring system based on memory sharing is developed, and the multimedia timing is applied to data receiving. In the testing system, a PCM demodulation card is inserted in the IPC, which is used for demodulating the received testing data from the front-end to PCM data, and then saving the PCM data to the prepared memory sharing region. In the design of real-time data processing software, the effective data in the sharing memory should be read out firstly, and then it will be separated, analyzed, resolved or displayed in curves according to the frame structure of PCM data.

In this testing system, the transmission rate of PCM data is in the scope of 1-15 Mbps, the size of sharing memory is set as (1 K + 5 M) bytes. Among of which, the former 1 K byte is the data attribution description block of the sharing memory, and the definition of it is shown in Table 4. Following the description block, 5 M byte space is used for storing testing data, and the data is written circularly in the entire space. The writing pointer in the description block is used to specify the position of the latest data; if the data space is filled, the pointer will be returned to zero.

Table 4. Data attribution description block of the sharing memory

Address	Bytes	Definition	Description
0~1	2	USHORT HeadLen	Length of property header
2~5	4	ULONG BufLen	Length of data
6~9	4	ULONG WrPt	Data writing pointer
10~1023	1014	UCHAR Reserved[1014]	Reserved bytes

4.1. Data receiving algorithm

Assuming the data transmission rate of PCM data is 5 Mbps, the sharing space will be filled in 8s approximately. When using multimedia timing method for receiving data, the timing interval can be set as 500 ms or 1000 ms; thus, the size of received data is 320 KB or 640 KB at a time. The algorithm process of reading and analyzing data based on multimedia timing method is shown on Fig. 2.

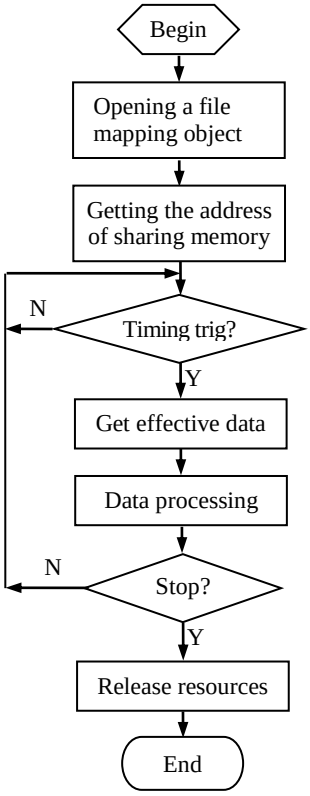

Fig. 2. Algorithm flow of timing data receiving and processing

Obtaining the latest data from the sharing memory must make use of the data writing pointer. Assume: [0 ~ MaxLen–1] is the entire range of data storage memory space; WrPt1 and WrPt2 represent the last and current position of writing pointer (position of the latest data) respectively. Because the timing interval is much less than the time filled the entire data space, for the wring pointer, it is certain that the "rings" situation never occur. According to the comparison of WrPt1 and WrPt2, there are following different cases, which are shown in Fig. 3.

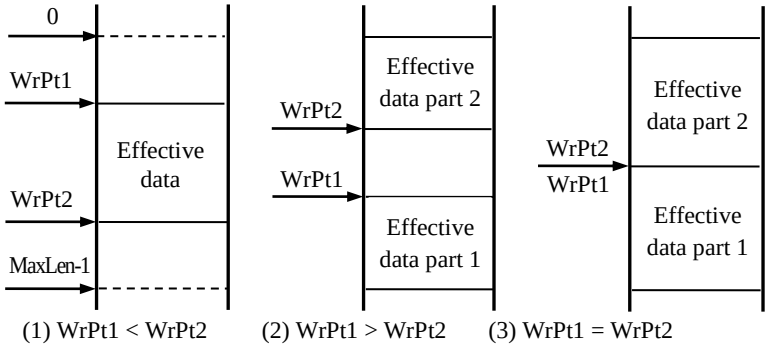

Fig. 3. Position of effective data

(1) If WrPt1 is less than WrPt2, the effective data is located in (WrPt1+1 ~ WrPt2).

(2) If WrPt1 is no less than WrPt2, the effective data consists of two parts: one is located in (WrPt1 ~ MaxLen–1), and the other one is in (0 ~ WrPt2). These two parts must be joined together to form an entirety before the following processing.

(3) In particular, if WrPt1 is equal to WrPt2, the entire space is the effective data, but the order needs to be adjusted from the position of WrPt1. However, the probability of this extreme situation is very low, almost negligible.

4.2. Timing error analysis

Because the above mentioned instance is a soft real-time system, its real-time demand is not very strict, so the multimedia timer can meet the requirement. By testing, this method can guarantee the data integrity in the case of 15 Mbps transmission rate. Moreover, timing error of multimedia timer has been tested, and the error curve is given in Fig. 4. In several timing testing experiments (sum: 1000), timing error occurred 18 times, the maximum error was 15 ms, and the average timing error was 0.048%.

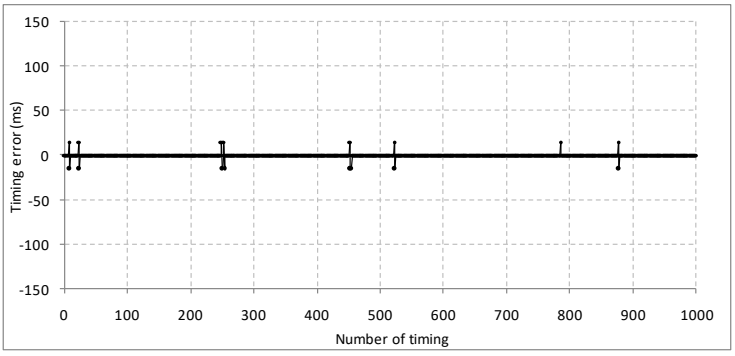

Fig. 4. Timing error curve of multimedia timer

There are two reasons for the timing error. Firstly, the timing processing procedure may be complex, so it fails to end before the next scheduled arrival. Additionally, the timer in the multi-task operating system sometimes cannot get a timely response. In general, when CPU is once occupied by some process or when system resource is limited, the message sent to the message queue will be temporarily suspended, which will lead to the timing errors sequentially.

5. Conclusion

Real-time monitoring is widely used in many fields. In order to achieve the real-time demand, the continuity and integrity of data receiving must be guaranteed. Because changes of some parameters occur frequently for an instant, data loss appeared during changing will lead to some serious consequences. In consideration of the drawback of low efficiency and excessive system resources occupation in

polling or multithreading method, multimedia timing method for data receiving is researched in this paper. Because the priority of multimedia timer is higher in the operating system, resource limitation has less impact on its timing function, and its timing precision is extremely high. In this paper, a real-time monitoring instance is introduced and the application of timing method for data receiving process is discussed. After testing, the timing error of multimedia is less than 0.05%, which can ensure no data loss under the circumstance of high data transmission rate.

***Acknowledgements*:** This work is supported by the National Natural Science Foundation (NNSF) of People's Republic of China (Grant No. 5127549).

References

1. U y s a l, R. B. Real-Time Condition Monitoring and Fault Diagnosis in Switched Reluctance Motors with Kohonen Neural Network. – Journal of Zhejiang University-SCIENCE C (Computers & Electronics), Vol. **14**, 2013, No 12, pp. 941-952.
2. D a i, Y., P. L i, Y. L i u, A. A s u n d i, J. L e n g. Integrated Real-Time Monitoring System for Strain/Temperature Distribution Based on Simultaneous Wavelength and Time Division Multiplexing Technique. – Optics and Lasers in Engineering, Vol. **59**, 2014, No 8, pp. 19-24.
3. S u n, Y., D. Z h a n g, H. T o n g. Research of Distributed Fiber Optic Sensing Technology in Monitoring of Majiagou Landside of Three Gorges. – The Chinese Journal of Geological Hazard and Control, Vol. **24**, 2013, No 4, pp. 97-102.
4. Y a n g, W., D. Z h a n g. Multi-Parameters Monitoring Underground Coal Mine Environment Using Mesh-Structured Wireless Sensor Networks. – Journal of Huazhong University of Science Technology (Natural Science Edition), Vol. **38**, 2010, No 10, pp. 70-74.
5. L e n g, X., J. C h e n, Y. Y e. A Distributed Environment Sound and Vibration Monitoring System. – Measurement & Control Technology, Vol. **32**, 2013, No 6, pp. 24-27.
6. Z h a n g, N., Y. Z h a n g. Development for a Remote Medical Real-Time Monitoring and Analysis Fronted Device. – Automation & Instrumentation, Vol. **25**, 2010, No 2, pp. 5-8.
7. F e n g, Y., J. H o u r d o s, G. A. D a v i s. Probe Vehicle Based Real-Time Traffic Monitoring on Urban Roadways. – Transportation Research Part C, Vol. **40**, 2014, No 3, pp. 160-178.
8. S h e n, S., S. Di, L. Q i n. Design of Real-Time Monitoring System Base on FPGA for Rocket Parameters. – Fire Control & Command Control, Vol. **36**, 2011, No 5, pp. 160-163.
9. W a n, Y., Z. X u, M. M e i. A Symbolic Execution Method for Conformance Test Generation of Real-Time System. – Acta Electronica Sinica, Vol. **41**, 2013, No 11, pp. 2275-2283.
10. A b d e s l a m, E., D. R a c h i d a, K. F e r h a t. Timed Wp-Method: Testing Real-Time Systems. – IEEE Transactions on Software Engineering, Vol. **28**, 2002, No 11, pp. 1023-1038.
11. K o p e t z, H. Real-Time Systems: Design Principles for Distributed Embedded Applications. New York, Springer, 2011.
12. B a i, X., M. W a n g, H. L u, W. T s a i. Verifying Timing Constraints in Real-Time Systems. – Journal of Tsinghua University (Science & Technology Edition), Vol. **52**, 2012, No 9, pp. 1286-1292.
13. G u o, Z., Y. M e n g, C. S u, H. W u. Windows Based Precise Timing Technology and its Engineering Applications. – Journal of Harbin Institute of Technology, Vol. **37**, 2005, No 12, pp. 1717-1720.
14. Y u, X., Y. W e i, M. H u a n g, X. Z h o u, C. Y a n g. Precise Timing of HLA-Based Hardware-in-Loop Simulation for Vehicles. – Journal of Zhejiang University (Engineering Science), Vol. **46**, 2012, No 7, pp. 1195-1200.