XVIII IMEKO WORLD CONGRESS
Metrology for a Sustainable Development
September, 17 – 22, 2006, Rio de Janeiro, Brazil

VIRTUAL REAL-TIME INSTRUMENTATION USING ETS CONFIGURATION

Piotr Bilski[1], *Wiesław Winiecki*[1]

[1] Institute of Radioelectronics, Warsaw University of Technology, ul. Nowowiejska 15/19, 00-665, Warsaw, Poland, +48 022 660, pbilski@ire.pw.edu.pl, W.Winiecki@ire.pw.edu.pl

Abstract- The paper presents the technique of Virtual Instrument design to obtain Real-Time mode. Application of the ETS approach to configure Real-Time instrument is presented and its requirements pointed out. Design of the virtual spectrum analyzer using the ETS technique is described. Detailed analysis of the designed instrument, considering ability to work in Real-Time mode, is conducted. The comparison between the ETS solution and formerly examined, traditional, approach is performed. Conclusions about the abilities of ETS RT virtual instruments and their practical applications are presented.

Keywords: Real-Time, virtual instrumentation, spectrum analyzer.

1. INTRODUCTION

The modern applications of the measurement systems and instruments cover broad spectrum of the technical domains. The most important are time critical tasks, where measurements and data processing must be performed in certain time instants. These are, for example on-line diagnostic [1] and monitoring systems [2]. To ensure determinism of the measurement process, strict hardware and software requirements must be fulfilled. Their existence describes the Real-Time (RT) mode, which is defined as the full determinism of the performed operations (Hard Real-Time - HRT) or their partial determinism, when time limits are not always kept (Soft Real-Time - SRT). In the virtual instrument (VI) design, the determinism refers to the synchronization between the data acquisition (DAQ) and signal processing operations. Multiple solutions for the RT mode exist; however, they require specialized hardware, such as PXI, compactPCI controllers, RT DAQ cards [3] or FieldPoint modules.

The possibility to obtain SRT mode in VI equipped only with the basic data acquisition hardware and general purpose operating system (GPOS) and LabVIEW programming environment [4] was also investigated [5]. The conclusions from these experiments were that in such a configuration SRT mode is obtainable, however, its quality is limited and determinism conditions are often violated. In practical applications more reliable solutions are required. Therefore, another platform for the VI design was needed. The description of the existing RT solutions is in Section 2, while Section 3 contains presentation of the analyzed instrument and hardware, on which it is deployed. In Section 4 there are results of the RT mode examinations and comparison with the traditional instrument. Section 5 contains conclusions and hints for the future RT VI designers.

2. SOFTWARE REAL-TIME SOLUTIONS

The basic RT solution tested before [5], based on the general purpose operating system, requires much effort for the optimization procedure and never assures determinism. Therefore sophisticated solutions involving RT operating system are required. The National Instruments (NI) company proposed two integrated solutions for the RT instrument design. These are: Real-Time Extensions (RTS) and Embedded Time System (ETS) [6], which expand abilities of the LabVIEW programming environment run on the personal computer (PC). Both solutions turn the target performing DAQ and data processing operations into instrument working under Real-Time operating system, so the determinism is assured. It is possible by dividing VI operations into time critical (DAQ and data processing) and non time-critical (presenting results, interaction with the user by the front panel). The former are run on RT target, while the latter are run on the PC under the general purpose operating system, such as Microsoft Windows™.

In the RTX approach, both RT target and no-RT instrument can be deployed on one computer and two operating systems are run simultaneously and share the same processor and memory. The disadvantage of such a solution is that for the communication between the DAQ hardware and RT system, specialized DAQ card with RT module is required. Without it, only low-level register programming (complicated and time-consuming) enables VI design.

The ETS approach is more universal, as even basic DAQ card is sufficient for the VI design. Two different computers are required here – one for time-critical, another for non time-critical tasks. The former can be PXI industrial computer or general PC, for which strict requirements (such as Pentium III processor, FAT32 hard disk partition and Intel 825xx family network chipset) must be met. Although ETS requires more hardware, it is a cheaper solution, because the price of two computers and basic DAQ card is lower than one computer and specialized RT DAQ card. Therefore it was selected for the experiments.

The ETS solution involves Ethernet network into the VI scheme. Although the RT target runs under RT operating system, communication between the target and the host is affected by the unreliable transmission protocols.

One of the key issues related to the RT processing is the priority queue. GPOS relies on the round-robin scheduling mechanism, granting every process in the main memory with computer resources for certain amount of time and does not ensure determinism. In the RT system, using combination of round-robin and preemptive scheduling determinism is possible by assigning VIs and threads with priorities. Influence of the priorities of the VI on its efficiency should also be tested.

3. HARDWARE AND SOFTWARE TEST CONFIGURATION

The virtual spectrum analyzer is the instrument processing data vectors, and RT mode requires that when the card acquires the next data packet, the previous one was yet processed and the processor is ready for another cycle. Two VIs were used for the experiments. The first analyzer was a simple version, performing power or amplitude spectrum calculation preceded by the initial data processing. The second analyzer was used before to establish RT conditions under GPOS [9] and can be used as the benchmark example. Its abilities include, windowing, filtering, power and amplitude spectrum calculation, linear and exponential averaging, as well as the cepstrum and histogram drawing. The hardware configuration for both instruments was the following:

- DAQ card: NI PCI 6023E, maximum sampling frequency – 200 kHz,
- Non Real-Time target: PC with Celeron 2,4 GHz processor, 256 MB of RAM, 80 GB hard drive, Intel 82559 network interface, Windows XP operating system,
- Real-Time target: PC with Pentium III 733 MHz processor, 256 MB of RAM, 40 GB hard drive, Intel 82558 network chipset, RT operating system.

The instrument was entirely designed on the non-RT target, and then deployed on the RT target, using LabVIEW 7.1. Real-Time module. As the communication protocol between the computers was used simple LAN connection and the protocol selected for the data transfer was TCP/IP. To store samples from the data acquisition, circular buffer [10] was applied. Its parameters were examined to determine the conditions to maintain RT mode. To compare ETS and traditional approaches, the sophisticated analyzer was run on the RT target computer, controlled by the general purpose and RT operating systems.

4. EXPERIMENTAL RESULTS

The conducted experiments are similar to the examinations performed before for the RT VI run under Microsoft Windows system [5]. The key factor of its efficiency was the circular buffer. This technique is also used in examinations presented in the paper. The time t_{proc} operating system has to perform necessary calculations (before the next samples vector is obtained) is related to the samples vector length n_{proc} [10]:

$$t_{proc} = \frac{n_{proc}}{f_s} \quad (1)$$

where f_s is the sampling frequency of the DAQ card. This equation is also valid for the VIs run under RT operating system.

4.1. Simple spectrum analyzer

Because the task scheduling mechanism in the RT target ensures determinism of the operations performed by the VI, the circular buffer was tested at the maximum sampling speed of the DAQ card. Every experiment was repeated 500 times to obtain the most reliable results. The experiments' results are in Tab. 1, where n_{proc} is the size of the processed data vector, n_{buf} is the size of the circular buffer, n_{RT} is the ratio of the non-delayed cycles, n_{max} is the maximum fill ratio of the circular buffer and n_{avg} is the average fill ratio of the circular buffer.

Tab. 1. Examinations of the analyzer for the sampling frequency of 200 kHz.

n_{proc}	n_{buf}	n_{RT} [%]	n_{max} [%]	n_{avg} [%]
512	4096	99,9949	26,02	2,62119
1024	8192	99,0118	17,65	2,1234
2048	16384	98,1225	18,51	3,3595
4096	32768	99,8703	19,86	2,7487
8192	65536	100	6,25	1,69064
16384	131072	100	1,93	0,85747

The ETS RT virtual spectrum analyzer delivers high accuracy and ensures determinism in most cases. Buffer fill ratio is low, even for the most pressing conditions. The size of the buffer, which optimal value $n_{buf} = 8 \cdot n_{proc}$ determined from the earlier examinations [10] is sufficient to hold RT conditions. Although RT target performs most of the VI software operations (on the non RT target only visualization was performed), in most cases the analyzer worked in the HRT mode. Because the communication between the deterministic and non-deterministic part of the instrument was not going through Internet, but in the private network consisting of only five laboratory computers, the delays related to the packet loss in the LAN were negligible.

The comparison between the virtual spectrum analyzers working under general purpose and Real-Time operating systems is in Fig. 1. The curves express the maximum length of the samples vector for the given sampling frequency, for which the RT mode can be obtained. The main difference between the two is that the instrument working under RT operating system fulfills the time requirements with a high accuracy and can be considered as working in the HRT mode. For the analyzer deployed under general purpose operating system (on the computer playing role of the Non Real-Time target) the percentage of the delayed cycles decreases for the relatively long data samples vectors (see Fig. 2). This is possible, because in such a PC configuration, spectrum calculation for the long vectors is faster than their acquisition. RT operating system assures stable work, independent of the sampling speed.

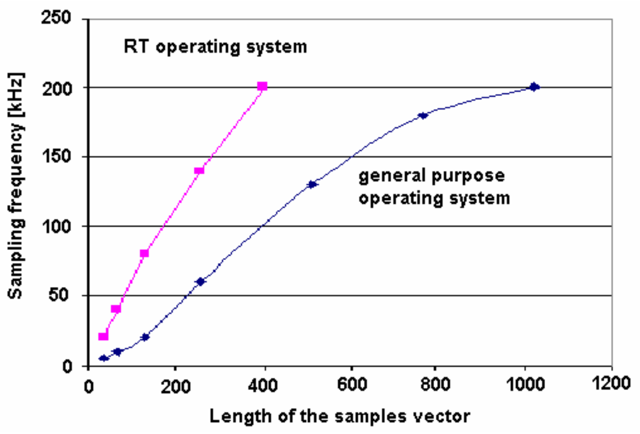

Fig. 1. Real-Time availability for the virtual spectrum analyzers under general purpose and RT operating systems

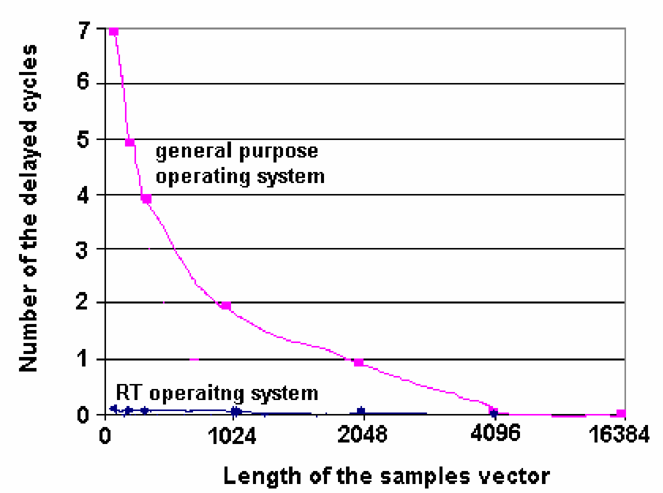

Fig. 2. Dependence between the delayed cycles rate and the length of the samples vector

4.2. Sophisticated spectrum analyzer

After experiments concerning basic VI, fully functional version of the analyzer was deployed on the RT target. It was also equipped with cyclical buffer to ensure parallelism of the data acquisition and signal processing. This analyzer consists of multiple subVIs responsible for the subsequent stages of operation: data acquisition, initial processing of the acquired samples vector, signal processing, presenting results on the front panel, and error handling with disk operations.

The first set of experiments was aimed at the comparison between the general purpose and RT operating systems. The analyzer performed basic signal processing operations (power spectrum calculation after uniform windowing, without filtering and averaging). The time of the operations duration was measured using the method presented in [5]. It resorts to the standard LabVIEW timer and requires performing every operation for multiple iterations (because LabVIEW timer has resolution of 1 ms, while typical cycle of the analyzer's work requires more accurate timer). The analyzer was run under both systems and identical parameters, i.e. maximum scan backlog s_{bck} (percentage of the cyclical buffer fill rate) as well as time of the operations t_{proc} were measured. The sampling frequency was set to 2048 samples pre second, giving range of 2 to 16 analyzer's cycles per second. Buffer's size n_{buf} was set to $8{\cdot}n_{proc}$. Results for different samples vectors are in Tab. 2.

Tab. 2. Comparison between the analyzer's parameters for Non-RT and RT operating systems

n_{proc}	s_{bck} [%]		t_{proc} [ms]	
	RT OS	Non-RT OS	RT OS	Non-RT OS
128	30,5	79,3	64,64	64,33
256	33,7	48,2	128,96	128,65
512	12,0	4,5	257,19	257,29
1024	0,3	0,1	513,21	514,58

The average time of the operations is similar in both cases. When there is no danger of the buffer overflow, analyzer's single cycle duration in both operating systems is determined by the sampling frequency of the DAQ card (if the software operations are finished before the acquisition, program waits idle until the next iteration). Therefore, for the longer vectors there is no buffer overflow and s_{bck} is close to zero.

Every VI run under RT system can get one of five priorities – from background (the lowest), through normal, above normal and high to the time-critical (recommended for data acquisition threads). By default, VI gets normal priority, which puts it far behind the time-critical priority processes. Therefore the experiment has similar outcomes for both operating systems. There is still threat of the cyclical buffer overflow, even in RT conditions, because the other processes easily expropriate VI's thread. Typical s_{bck} fill rate during the typical experiment (400 repeated signal processing operations in a loop) is presented in Fig. 3.

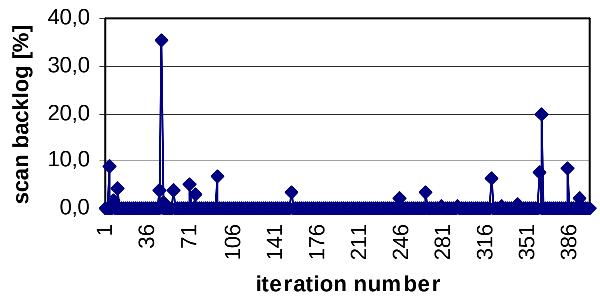

Fig. 3. Cyclical buffer fill rate during experiments for RT operating system with normal priority of the analyzer's thread

The difference between the operating systems is easy determinable, when the analyzer's priority was changed under RT operating system to time-critical. This way, its process and threads are not expropriated and the circular buffer is empty most of the time. The comparison between the general purpose and RT operating systems is in Fig. 4.

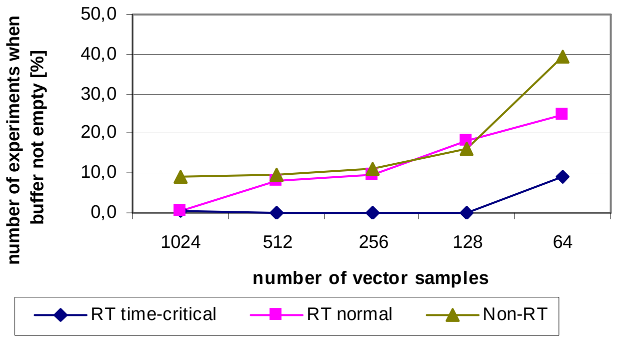

Fig. 4. Comparison of the cyclical buffer fill rate for RT and general purpose operating systems

The advantage of the VI with time-critical priority is easy distinguishable. Not only the analyzer works in RT mode, but also the buffer overflow is unlikely. Cyclical buffer is empty most of the time, even for the short sample vectors (i.e. 64), where Non-RT version rarely works without the buffer overflow. Therefore n_{buf} may be smaller, for example:

$$n_{buf} = 2 \cdot n_{proc} \tag{2}$$

There is another threat, related to the priority of the VI. If its duration is too long, another processes, required by the operating system may not get enough time and the RT system gets stalled [11]. To avoid such a situation, a time-critical VI must contain preset delays to allow other task to be completed. In our experiments, 5 to 10 ms was enough to avoid hang-up. The influence of the priority on the cyclical buffer fill rate (for f_s = 2048 Hz) is shown in Fig. 5. The curve for the 64-samples vector is the border, above which RT conditions are hard to obtain for the VIs with the lower priority. Acquisition of 1024 samples vector with the preset f_s leaves the operating system about 500 ms to perform the necessary calculations. Notice, that Windows NT family also offers prioritizing, however changing VI into high priority makes performing another tasks impossible.

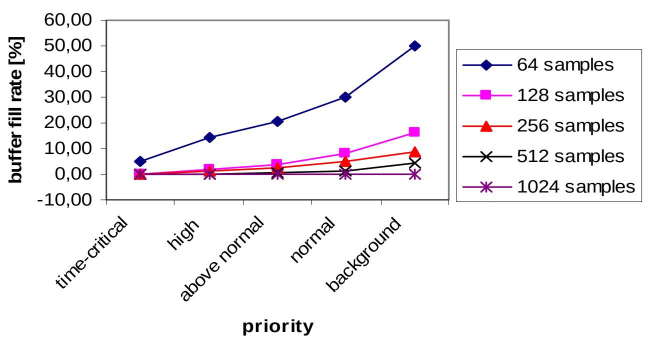

Fig. 5. Comparison between the efficiency of the VIs with different priorities

The time of the operation in RT system is measured and transferred to the host computer. It is then independent of the network congestion and can be compared to the values measured under general purpose operating systems. It should also be independent of the operations performed on the host computer. The experiments confirming this are presented in Tab. 3.

Tab. 3. Comparison of the operations' duration under both operating systems

n_{proc}	RT1 [ms]	RT2 [ms]	Non-RT [ms]
128	514,285	514,48	514,58
256	258,01	258,135	257,29
512	128,595	128,635	128,65
1024	64,38	64,415	64,33

Column RT1 contains the average time of operations for the RT system with analyzer of time-critical priority. Column RT2 contains the same parameter for the same system when there are additional operations run on the host computer (defragmenting hard disk, searching for the files, using spreadsheet). The obtained values are close to the ones for the analyzer run under Microsoft Windows ("Non-RT" column).

Another issue is the accuracy of the time measurement. For the RT system it can be performed by the function using the DAQ card's internal clock, giving better accuracy than the software function. Based on the hardware, it is independent of the scheduling mechanism and fluctuations related to the processes execution time. The time measurement function is presented in Fig. 6.

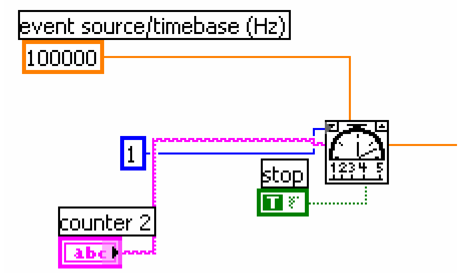

Fig. 6. LabVIEW time measurement function, using DAQ card clock

Both time measurement functions were compared. While for the long experiments, with multiple iterations, their results are comparable. For the RT conditions the more accurate timer can be required, especially when the exact operation time must be known with a high accuracy. Results for ten iterations of the VI's operation are shown in Tab. 4.

Tab.4. Comparison between the software and DAQ card clock based measurement functions

n_{proc}	software timer [ms]	DAQ timer [ms]
128	71,318	71,96
256	150,06	156,48
512	293,6	302,16
1024	583,78	567,03

The differences between the timer readings are related to the actual operation time and are diminished for the larger number of iterations. Both have identical resolution, but the DAQ card clock-based timer presents more reliable results.

For the RT operating system it is important that all the programs are stored in RAM. Therefore their size matters and the designer should focus on minimizing the size of the VIs deployed on the RT target. LabVIEW module allows disabling debugging options for every VI included in the project. In Tab. 5 there are presented sizes of the selected VIs uploaded to the RT target before and after suppressing their size. The size of the whole analyzer mainly depends on the number and size of the VIs selected for the project. There is also small number of the controls uploaded automatically to support the execution of the analyzer, but their size is irrelevant. In Tab. 5 the size of the whole instrument as well as its parts is presented.

Tab. 5. Illustration of the minimizing size of the constituent subVIs of the analyzer

VI function	Before size suppressing	After size suppressing
Selecting signal source	9,8 KB	8,46 KB
Saving results to file	28,49 KB	22,23 KB
Error handling	23,18 KB	17,19 KB
Data acquisition	4,73 KB	4,3 KB
Initial processing	5,16 KB	4,73 KB
Main instrument	284,43 KB	190, 44 KB

Results in Tab. 6 show the memory savings for controls (uploaded automatically by the LabVIEW and related to the used functions) and VIs. The size suppression is possible only for the latter and is about ten percent.

Tab. 6. Illustration of the minimizing size of the whole analyzer

Analyzer elements	Before size suppressing	After size suppressing
Controls	5,82 KB	5,82 KB
VIs	1177,37 KB	1063,02 KB
Whole analyzer	1183,19 KB	1068,84 KB

The size suppressing is better visible for the larger VIs, for which the saving of space in RAM is up to thirty percent (main instrument). The whole analyzer requires only about 1 MB in the main memory, while the RT system takes about 50 MB, so for the tested RT target there is a lot of free space for additional instruments. Therefore, for the VI deployed on the modern PC computer, its size is of the secondary importance, unless there are many VIs uploaded to RAM.

5. CONCLUSIONS

The examinations prove that virtual instrument designed using ETS technique with properly set priority level assures high determinism and can be used for more sophisticated tasks, than the VI based on the general purpose operating system. Its advantage over the traditional solutions is reliability even for the maximum sampling speed delivered by the DAQ card. The Real-Time operating system avoids buffer overflow and loosing SRT conditions. The optimal circular buffer parameters calculated in [5,10] are still valid, but now they can be stricter. This gives the designer's more freedom during software optimization. Practical applications of such an instrument are also wider, as there is are no limitations on the sampling frequency, as in the traditional solution. This makes ETS RT virtual spectrum analyzer a powerful tool for measurement and on-line diagnostics.

Design of the RT VI requires considering novel ideas, such as prioritizing the VI, or controlling the amount of memory used by the program on the RT target. The ETS configuration of the VI is expanding towards the distributed RT measurement system [8]. Obtaining deterministic transfer between the host and the RT target is the next step to the RT VI technique using Ethernet network.

Determining abilities of the network in the distributed measurement systems is a pressing issue [7], which is supported by the newest version of LabVIEW (8.0). The idea of establishing a subnet within the present network (similar to the Virtual Private Network solution), delivering deterministic behavior [8] requires only additional network cards. Experiments concerning new abilities of the LabVIEW RT module will be the next step of our research.

ACKNOWLEDGMENTS

This work was supported by the National Research Committee, grant no. 3 T10C 008 26.

REFERENCES

[1] M. Auboiroux, MVI Technologies, "LabVIEW Real-Time Manages an Endurance Test Bench for Renault Engine Components," from www.ni.com

[2] S.C. Chen and H.T. Yang, "Implementation of a Virtual-Instrument for Non-intrusive Intelligent Real-Time Load Monitoring," *Proc. NCI'03 Conf., (Cancun, Mexico, May, 19-21, 2003)*, pp. 174-178.

[3] "Getting Started with the NI 7831R," from www.ni.com

[4] www.ni.com/labview

[5] P. Bilski, W. Winiecki, "Time Optimization of Soft Real-Time Virtual Instrument Design," IEEE Trans. Instr. And Meas., Aug. 2005, Vol 54, No. 4, pp. 1412-1416.

[6] "Phar Lap Embedded ToolSuite (ETS)" in http://www.vci.com/products/low_power/ets.asp

[7] W. Nawrocki and M. Wawrzyniak, "LAN-Based Measurement Systems," *IMEKO'05 Symposium (Gdynia-Jurata Poland Aug. 12-15, 2005)*, pp. 261-265.

[8] "Using Time-Triggered Networks to Communicate Deterministically Over Ethernet with the LabVIEW 8 Real-Time Module," from www.ni.com.

[9] P. Bilski and W. Winiecki: "A Low-Cost Real-Time Virtual Spectrum Analyzer," *IMTC'05 Conf. (Ottawa, Canada May, 17-19, 2005)*, pp. 2216-2221.

[10] P. Bilski and W. Winiecki, "Software Approach to Real-Time Virtual Instrumentation," *IMEKO'05 Symposium (Gdynia-Jurata Poland Aug. 12-15, 2005)*, pp. 523-528.

[11] "LabVIEW Real-Time Module User Manual," National Instruments, 2004, from www.ni.com.