

Implementation of the Satellite Ground Station Control in Real-Time Under Windows

Mehdi Vakilfard^{a*}, Asghar Taheri^b, Mohammad Reza Salehifar^c

^a*MSc Electronic, Malek-Ashtar University of Technology, Tehran, Iran*

^b*Department of Electronic, Zanjan University, Zanjan, Iran*

^c*MSc Electronic, Science and Research Branch, Islamic Azad University, Tehran, Iran*

Abstract

As a result of crucial importance of control panels in satellite earth stations, for the purpose of implementation of this panel, a separate DOS operating computer is used and the rest of commands are executed by another computer running windows OS. Combined application of two computers and establishment of coordination between them, in addition to lack of suitability of the DOS operating system for users are the weaknesses of the mentioned method. Therefore, this article focuses on implementation of the entire operations on a single Windows XP operated computer. In this regard, the control panel is meant to be a real-time system. Making use of the software of MATLAB in combination with application of Windows NT and the Windows CE operating systems are the steps which have been taken in order to obtain the former goal. In this article, the software of RTX Venturcam is used for implementation of the real-time control algorithm in combination with implementation of some other side programs under Windows XP. RTX is approved by Microsoft and by making use of this application, the both advantages of real-time execution of tasks and a user friendly environment can be obtained.

Keywords: Real-Time System, Operating System, Control Panel, MATLAB Real-Time Execution Environment, RTX Software

Introduction:

In most applications, monitoring and data acquisition and control systems are supposed to accomplish their tasks in a limited range of time and in a random manner. This time range depends on the type of application and it could vary from a fraction of microsecond to multiple hours. The time spent by the control system for initiation and implementation of the intended service should be in a way that responding each sample is finished before the arrival of the next sample. Three main features of data acquisition and control systems include:

- 1) Having an adequate speed

- 2) Certainty in responding
- 3) Predictability of response time

By certainty in responding, we mean that the system should certainly respond to any external stimulation in a way that no event is left without a response. In applications with very short time ranges, for the purpose of guaranteeing the aforementioned three features, and as a result of validity and timeliness of the given response, the data acquisition and control systems should be designed in a way that they are able to fulfil the requirements of the intended application. In not a very long time ago, for the purpose of responding to tasks which required microsecond time ranges, special hardware which were designed through application of logic circuits or analogue circuits were used and these systems were not able to handle software implementations. Systems which were designed in this fashion were single-purpose and lacked any flexibility in terms of changing their mission. But the method which is nowadays implemented as the best method of design and development of data acquisition and control systems, is the use of the increasing power of Microprocessors. There are three major solutions for making use of microprocessors in control systems:

- 1) Making use of single-purpose micro-computers
- 2) Making use of common controllers such as PLCs
- 3) Making use of personal computers

Currently a large variety of computers are in disposal of users and according to each applied requirement, it is possible to make use of several different PC based solutions. By providing new computer based control services, PCs play an important role in terms of implementation of different software and hardware components produced by a large number of manufacturers. The groups of reasons which have led to attraction of attentions by PCs for establishment of a control system include the following points [4]:

- a) Worldwide accessibility
- b) Gaining valuable experience by implementation
- c) Daily innovations as result of numerousness of manufacturers
- d) Tight competitions between manufacturers and consequent balance of prices
- e) Shorter development cycles
- f) Smaller waiting times in terms of occurrence of problems as a result of benefiting from diagnosis and simulation processes
- g) Increased productivity
- h) Reduction of costs of maintenance, development and training
- i) Accessibility to local and worldwide networks
- j) Provision of a strong user interface
- k) Ability for exploiting different programming languages at different levels

As a result of existence of such features, nowadays industry has witnessed a large shift from traditional methods of control including using PLCs, to PC based controllers.

In order to supplement for the features of speed and certainty in PC based control systems, different solutions have been proposed. Among these solutions, the boldest are:

- a) Making use of a PC along a PLC
- b) making use of DSP in form of a cryptograph inside the PC known as Host/DSP
- c) making use of a single PC operating a single OS with real-time operability

Among the aforementioned methods, the last method is the newest and the most efficient. An OS with Real-time operability featuring different characteristics including multitasking and prioritization based timing guarantees a certain response in control based applications. In other words, the entire tasks including HMI and control algorithms will operate on a single processor. On this basis, the most suitable choice for implementation of a control system, is to use PC hardware running a real-time OS.

Real-time systems:

As it was mentioned before, in many control based tasks; the control system should be able to react in a predetermined time, in order to be sensitive to real world stimuli. In such applications, if the response is sent after the determined time, the response is totally invaluable and also it is followed by undesirable consequences. As an example it be referred to a missile's automatic navigation system. In such a system, decision making and responding processes of the control system related to gathered data from the target location should be completed in a suitable and short time with the pace of the target. Otherwise before the control system sends out the correct response as a set of information about the location of target, the distance between the missile and target location is increased to the extent that continuing the operation of navigation is no longer possible. Such applications are commonly called real-time systems and also systems which are capable of such operability is called a real-time system. However, real-time systems have some precise time limits. In this regard, processing must be completed in a certain amount of time in addition to production of correct result and, otherwise, the system will malfunction. In real-time systems, certainty of response and preparation of the correct response in a certain amount of time is more important than the speed of preparation of response.

Real-time tasks are divided into soft and hard groups. The task of soft real-time includes time based deadlines which are preferred and desired to be taken into account. In this case, even after the expiration of the time limit, the system can continue its operation and complete the task. In such systems, time limits are relatively precise but not guaranteed and lack certainty. Many of modern systems such as UNIX support a soft real-time operability. The task of soft real-time includes the following features:

- a) Delay in responding increases the costs
- b) Per each delay, the system performance is diminished. As an example of this application it can be referred to network interface subsystems.

In a hard real-time systems, time limits are strictly precise because otherwise, irreparable errors will take place. Hard real-time applications include the following rules:

- a) No delay in accepted under any circumstance
- b) In case of delay in suplication of calculation results, the results are spoiled
- c) In case of missing deadlines, disastrous failures will take place

An example for an application of a hard real-time system is the autopilot system of an airplane.

Design and simulation of the real-time system through the application of MATLAB software

For the purpose of simulation of a real-time system, the software of MATLAB could be a good choice. RTWT provides us with the possibility of using the core for real-time performance and execution of projects. In a real-time program, the code should by synchronized with real world time and also the time of sampling should be constant. The entire system uses different modules for implementation of the model. This part includes a code for defining, constructing and manipulating a model. The model code executes drivers for calculation of output and at the end, it updates time in a real-time fashion. In addition, the executive model includes reading the input from external hardware, calculating the output model and insertion of output in related hardware. RTWT is a computer based way for capturing patterns and testing real-time systems. This environment includes a single computer machine as the host in which the block model of Simulink is executed in State flow configuration. Upon completion of the model and consequent simulation of the model with Simulink in normal mode, the executive code could be produced by RT workshop, State Flow Coder, Open Watcom and C/C++ compiler. In this regard, ultimately the project is executed in a real-time fashion in external mode of Simulink. The advantages of implementing RWT include:

- a) Lack of requiring a separate operating system capable of real-time execution of the control algorithm
- b) Cheap prices of hardware and software required

- c) Capability of real-time execution of the control algorithm on Windows OS.
- d) Possibility of imposing rapid changes on the control algorithm
- e) Possibility for implementation of the control algorithm for testing

On the other hand, disadvantages of implementing RTWT include:

- a) Problems in the process of transferring codes between the OS and processes
- b) Nonstandard development environment
- c) Increased vulnerability of the OS
- d) Requiring a large amount of hard disk
- e) Creation of time excess
- f) Lack of dependability for execution of real project

An OS capable of real-time tasking:

An OS is a program which operates as an interface between the user and hardware. The OS provides an environment in which the user is able to execute different programs. The primary goal of an OS is to prepare the computer for easy use. However the second goal is to make optimal use of the hardware.

In a PC based control system, the real-time core is one of the most important components which requires extreme precision upon design and choice. Nowadays, different firms provide real-time operating systems including VsWorks, OS9, pSOS, QNX, Lynx OS and OSE and CX/UX. The provider of a guaranteed certainty in system performance is an OS capable of execution of real-time applications.

The most important criterion in discussing an OS capable of real-time tasking which directly impact the system's response time to external events, is a parameter named as interrupt latency. Interrupt latency is defined as the interruption time between occurrence of a hardware delay and the initiation of real-time processing of the delay. This delay depends on the method of an OS's servicing. The required features of an OS capable of real-time tasking include [4]:

- a) Being multipurpose
- b) The concept of purpose priority
- c) Pre-empting capability
- d) Supporting predictable synchronization mechanism
- e) Existence of an inheritance system

By real-time capability of a system, we do not merely refer to rapidness of the system. However, the important factor is the predictability of the maximum amount of time which is spent by the system on execution of a specific task. Moreover, the more important factor is that this amount of time should match the criterions of the specific real-time application. Real-time systems are not essentially the most efficient systems, rather they are able to perform some tasks with time certainty and guarantee. For example, the OS of XP windows is designed for maximum efficiency. In an OS capable of real-time utilizations, the higher priorities should be completed earlier than lower priorities. In addition, in special time intervals, the delays of Mouse and etc. are not considered for.

Table 1, comparing GPOS and RTOS

RTOS	GPOS
Optimize worst case	Optimize average case
Predictable schedule	Efficient schedule
Simple executive	Wide range of service
Minimize latency	Maximize throughput

Nowadays, different operating systems have been introduced by different firms each of which have their own advantages and disadvantages. For making industrial use of PC based control systems, a single standardized real-time system should be used so that in addition to making use of a suitable and proper environment, a matched set of hardware are used.

Windows NT is an OS which was used for obtaining this goal. Some of the advantages of this OS include:

- a) Existence of useful programming interfaces for 32bits windows
- b) Having better interfaces compared to RTOS
- c) Preparedness of communication devices
- d) Existence of several development tools

However, the Windows NT is not basically designed for responding to the requirements of a real-time system. In terms of prioritization, this system lacks the ability for substituting priorities which is an application of a real-time system. Windows NT is merely used in simple real-time systems under the following conditions:

- a) Soft real-time systems which are not strict about deadlines
- b) Simple systems with a reduced number of events
- c) Systems in which the CPU load is usually maintained in low levels

With daily increase in applications of PC based control systems, and utilization of these controllers in other smaller devices, the need for reducing the size of hardware as well as applied software forced Microsoft to develop an operating system named as Windows CE which possesses a completely different structure compared to the other products of Microsoft including Windows 9X and windows NT. This operating system is designed for fulfilment of a series of special needs:

- a) Requiring low storage space
- b) Supporting Microsoft APIs
- c) Operable on systems which lack hard-disks, keyboards and monitors
- d) Lack of dependence on a certain family of processors

On this basis a system with a modular structure was introduced which was only provide for customers through ordering.in contrast to other Operating systems, windows CE is not provided only based on a specific standard, rather it is designed in a modular way for providing high flexibility for responding to needs of a large number of products. This modules themselves are divided into components in a way that absence of a single component does not impose negative influence on the overall system. Therefore, with respect to the needs of the system, the minimum number of required modules could be used and therefore, the space required for operation of total modules is reduced. Windows CE empowers PCs to perform superior to common PLCs in terms of real-time control. In addition, the delay responding time in Windows CE is 50 times less than windows NT. While using a Windows CE, the user can have full control over timing system. Other features and characteristics of this system include: real-time tasking with 256 priority levels, custom style, supporting Microsoft APIs, operability on systems which lack a mouse, keyboards or monitors and hard disks and operable on embedded systems. Being modular, flexible and including cheap software and hardware with the ability of accepting multiple delays while being developable. On the other hand, a nonstandard environment and vulnerability are considered as disadvantages of Windows CE.

Implementation of real-time control system under windows XP and by using RTX

In general, windows XP is not sufficient for development of a real-time system according to the needs of users. However, by using the software of RTX which is developed by Venturcom co. this system becomes applicable even for difficult real-time applications. Windows XP features modern and advanced characteristics, a large number of hardware support it, is accessible and mass-production of its supported accessories and therefore, it is considered as one of the best choices for designers and users.

On the other hand, since the APIs of windows XP are not developed for real-time purposes, application of this OS for real-time tasks is strictly restricted and may seem impossible. Ardency's RTX solves this problem by adding real-time functions to the Windows XP. It also has developed an instrument for development and execution of real-time programs which could be well-used in control based applications. This programs provides a suitable hard dependable real-time environment which can be used to exploit a cheap platform for embedded and real-time applications. Although, however RTX has provided an excellent environment for real-time applications,. This response is still largely dependent on the type of system, designer and the user. Therefore, we are required to expend sufficient precision upon selecting the type of system with respect to the type of application. By the help of efficiency analysis tool of the RTX software, the designer and the user can both select a system which fulfills their highest expectations. In addition, the designer should execute the RTX for his or her customer in a correct fashion. In other words, RTX provides a primary context or strength only if the implementation is correct. The RTX API is based on a Win32 API. API is defined as a programming interface which is applied for controlling the programs of the operating system. This interface includes a set of functions and sub procedures which by being placed in a dynamic linear library (DLL), provide the possibility of accessing and manipulating the features of the Operating system through programming. WIN32 and RTSS both support RTX functions however, they possess different response times and efficiencies. On this basis we are allowed to run a single RTX code in both environments. In addition to supporting APIs, the WIN32 supports RTAPIs too however, some certain APIs are only supported in RTX. The RTX clock in both environments of RTSS and Win32 is accessible. RTX has three types of clock which are resulted from Real-time HAL which in turn has a resolution of up to one microseconds. One example of pathway is PCI which connects the sub-system of memory and processor to rapid devices. Other pathways including the keyboard and serial ports are relatively slow. Here, the PCI 1711 card is used. For implementing the PCI card in RTX environment, first it should be configured according to RTX devices. For changing the IRQ, first the accessible IRQs in the system should be observed. In delay histogram, the X axis shows the amount of delay which was previously defined as the difference between the desired and the measured time. On the other hand, the Y axis shows the number of delays which occur in a specific time cycle. In the diagram of maximum delay, the X axis shows the time and the Y axis refers to delay.

In the following example, the amount of delay in windows XP with an absent RTX and with a present RTX is measured and subsequently, reported on the related table.

Table 2, comparison of delay in Windows XP and RTX environments operating on a DELL Dimension 9200 Core2 Duo 2.66 GHz with 2GB of ram.

Operations	Windows Latency (ns)	RTX Latency (ns)
SetEvent (no thread switch): min / max	383/273048	135/1244
SetEvent□ WFSO: min / max	533/291868	210/447
ReleaseMutex□ WFSO: min / max	556/286842	259/635
ReleaseSemaphore□ WFSO: min /max	556/274191	240/1093
Yield: min / max	432/67018	127/578
Thread priority change: min / max	507/275943	233/601
Interrupt service thread dispatch: min / max	2000/250000	0/1000

Table 3, comparison of the Timer delay between Windows XP, Windows CE and RTX

Operation	Windows XP	CE 3.0	RTX 5.1
SetEvent (no thread switch): min/max in μ s	1.04 / 5000+	1.49 / 7.20	0.29 / 2.71
SetEvent→WFSO: min/max in μ s	1.38 / 5000+	2.46 / 10.7	0.60 / 2.96
ReleaseMutex→WFSO: min/max in μ s	1.49 / 5000+	3.51 / 10.5	0.70 / 3.26
ReleaseSemaphore→WFSO: min/max in μ s	1.39 / 5000+	3.00 / 9.40	0.61 / 3.43
Yield: min/max in μ s	1.11 / 5000+	1.32 / 8.34	0.33 / 3.37
Thread priority change: min/max in μ s	1.31 / 5000+	1.41 / 8.96	0.56 / 3.81
Interrupt service thread dispatch: min/max in μ s	4.3 / 5000+	4.3 / 26	2.0 / 19
Win32-to-RTSS SetEvent call: min in μ s	NA	NA	14

The histogram of the timer delay under Windows XP running RTX 5.1 on a system equipped with a Pentium III 8MHz processor in different environments of Windows XP and RTX is shown in figure 1.

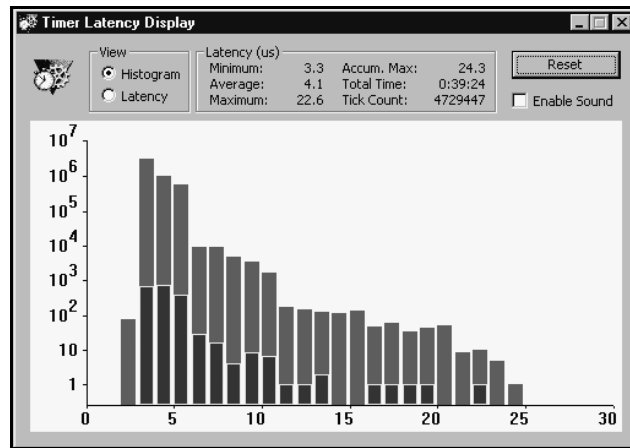


Figure 1, the histogram of timer delay in RTX

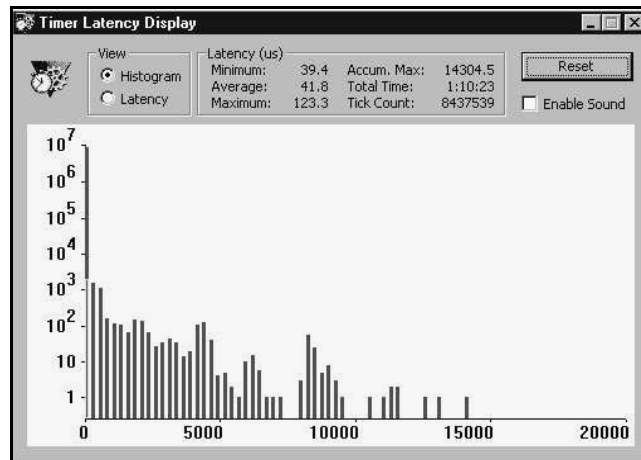


Figure 2, the histogram of Timer delay in windows

Results of implementation:

In the first test, the pulse that was created by the generator function is applied to the digital input of the PCI 1711 card. The pulse applied to the card is written in VC software and is directly conducted towards the digital output of the card. The delay time between the incoming and outgoing pulses of the card are measured. This time is a criterion for system delay. The same test was performed in windows, RTWT and RTX and results were compared. With respect to the fact that the worst answer is considered in real-time applications, the tests were ran while the CPU was processing a heavy graphics program.

Table 4, results of the test 1 in windows, RTWT and RTX

Windows XP	RTWT	RTX
>10 MS	≈2 MS	<150 μs

In the second test, a pulse with frequency of 10Ms was created by the Timer in VC software and the digital output of the Card was exposed to stimulation by the pulse. This test was also ran in windows with windows timer and in RTX with HAL timer and subsequently, the results are compared with each other. The reason that RTX uses the HAL timer with a resolution of 10μs is that the outgoing pulse is of extreme precision but the windows XP timer has a resolution of 10μs. As you can see in figure 3, the frequency of the pulse is 10μs and the system as 10μs for processing. However the time for processing should not exceed 10μs. therefore, in the written program, this point is taken into account. The test answer is highly dependent on T2 in windows XP and influences the output. However, in RWT as the T2 becomes closer to T1, the possibility of error increases but in the RTX test it was observed that changing T2 had no influence on the output signal.

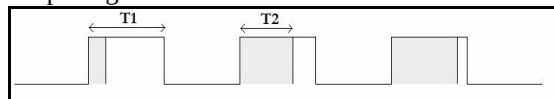


Figure 3, the second test

Table 5, results of the second test in RTX and Windows XP

Windows XP	RTWT	RTX
>10 ms	≈1 ms	<50 μs

Conclusion:

Using an RTOS in conjunction with an I/O card is the best solution for implementation of PC in control systems. Because the processing power of modern processors have guaranteed a certain response in control based applications. Interrupt delay is the most crucial criterion in studying an OS capable of real-time tasking. Windows NT is not developed for fulfillment of requirements of a real-time system. As a result of requiring low disk space, supporting Microsoft APIs, being customizable and etc... windows CE has applications in real-time systems.

By making use of the MATLAB software and the toolbox of RTWT a Simulink model in which input/output cards have been used, could be established and executed in a real-time fashion. This method is a suitable approach for simulation of control systems in scientific centers. However, as a result of making use of the heavily operated software of MATLAB, they are not practically used for implementation of control systems.

Windows XP is generally inadequate for fulfillment of requirements of a real-time system. Having advanced features is an advantage of the windows XP but the APIs of windows XP have not been developed for real-time applications. Therefore this system has limitations in real-time applications. Implementation of real-time systems is feasible under Windows XP and by the use of the RTX software.

The sub-system of RTX is able to support applications which require certain responses. The amount of delay in environments of WIN32 and RTX are explained in Histogram and maximum delay diagrams.

References:

- A. Taheri, "design and simulation of the satellite tracking antenna control system, with classical PID controller and matching", dissertation master, Malek Ashtar Industrial University, 2001.
- Teimuri, "Pc based controllers", master's thesis, Malek Ashtar Industrial University, 2002.
- Dave Faulkner, Ben Levy and Tim Gerner, "Open-Architecture Platforms", Circuit Assembly Magazine, 1999.
- Dr. Ir. Martin Timmerman and Ir. Jeen Christopher Monfert, "Windows NT as Real-Time OS?", <http://www.dedicated-systems.com/magazine/97q2/winntasrtos.htm> 1997.
- Advantech Co, "Advantech Hardware & Windows CE For Machinery Industry Embedded PC World Online", 2001, <http://www.advantech.com.tw/epc/newsletter/v51-07/wince.htm>.
- N. Costescu, D. Dawson and M. Loffer, "Qmotor 2.0 - A Real-Time PC Based Control Environment", IEEE 1998.
- N. Costescu and D. Dawson, "Qmotor 2.0 - A PC Based Real-Time Multitasking Graphical Control Environment".
- N. Costescu, M. Loffer, E. Zergeroglu and D. Dawson, "Qrobot - A Multitasking PC Based Robot Control System ", Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29634, USA.
- Markus S. Loffer, Vilas K. Chitrakaran and Darren M. Dawson, "Design and Implementation of the Robotic Platform", Department of Electrical and Computer Engineering, Clemson University, Clemson, SC 29631 - 0915, USA. Natinal Instrument Corporation, "Lab VIEW Real-Time User Manual".
- Ir. Martin Timmerman and Ir. Jean-Christopher Monfret, "Windows NT Real-Time Extension: an overview", <http://www.dedicated-systems.com/magazine/97q2/winntasrtos.htm>.
- Natinal Instrument Online News "Real-Time Control and Reliability With Windows/NT", <http://www.ni.com/news/01rt.htm>.
- Extending Windows NT for Embedded and Hard Real-Time Systems, Real-Time Magazine 1997.
- Implementation the win32 API over a POSIX real-Time OS", Real-Time Magazine 1997.
- Nat Frampton, John Tsao and Jerry Yen, "Hard real-Time Extension of windows NT Evaluation Report", General Motors Powertrain Group, 1998.
- Paul Yao, "Windows CE 3.0: Enhanced Real-Time Features Provide Sophisticated Thread Handling", MSDN Magazine, 2000.
- Real-Time System With Microsoft Windows CE 2.1", MSDN Library Microsoft Corporation, 2000.
- W. E. Dixon, D. M. Dawson, B. T. Costic, and M. S. Queiroz "Towards the standardization of a MATLAB-Based Control System Laboratory Experience for Undergraduate Students", Proceeding of the American Control Conference Arlington, VA June 25-27, 2001.
- real-Time Workshop User Manual", MathWorks Co.
- real-Time Windows Target for Use with Real-Time workshop", MathWorks Co.
- PCI-1710 Series User's Manual", Advantech Co.
- "Basses", ebook, <http://www.dcs.napier.ac.uk/~bill/books.html>.
- Hard Real-Time with IntervalZero RTX® on the Windows® Platform, June 2009.