Machine Control with CANopen and Real Time OS

Rainer Gallus, Woodhead Software & Electronics

Abstract

CANopen applications are often used inside machine solutions. This applications includes more and more control applications with time critical requirements. With this requirements real time OS are often the only way to fulfill this process requirements. Due to the fact that most of the readers know CANopen, this article is more focused to real time requirements. This article first provides a summary of the definition and requirements of real time OS and shows, based on a real project, how this features are used in the context with CANopen communication.

The goal of this article is to give developers of CANopen some background based on an example how he can use this interesting features in his own projects

1.0 Real Time Operating Systems

What is a real time OS

If you are at fairs or exhibitions, visitors often asks if the presented communication products supports real time requirements. During the discussion you could very fast realize that each of them has his own very different definition what real time means.

Let us start with an overview of the basic definitions for a real time System. A real time system is a system in which the correctness of the function depends not only on the logical correctness, but also on the time at which the results of the functions are produced. If the timing restrictions of the functions are not meet all requirements, logical and timig, a system failure occurred.

That means the value of a function in a real time system depends on how the time result is produced. For example if a robot has to pick up something from a moving conveyer belt in a small time slot and the robot is to late, the piece won't be there anymore. The robot has done his function incorrect, even though the robot went to the right place. You will find real time aspects in a lot of processes even nobody mentions this because the timing is not in such short time periods as we usually discuss it in the automation processes. For example think about the "Just in time" delivery of goods for production plants. The time of the delivery is scheduled and if the goods are at the appropriate time available the delivery has failed.

Problems will arise when there is competition for system resources and resources has to be shared among many activities. Then we start to begin to apply real time characteristics to operating systems. Implementing of any real time features means that the critical step in the process will be the determination of a schedule of activities, to make sure that all activities will be completed in time.

A real time system will contain all activities - real time activities that can be scheduled and non-real-time activities which cannot be scheduled. The real time system has to be able to handle the scheduled activities with preference to the non-scheduled activities to avoid any timing constraints.

Real time systems are characterized by the fact that serious consequences will result if the timing correctness properties of the system doesn't work. Typically a real time system consists of a controlling system and a controlled system. For example in an automated factory the controlling system is the computer with the human machine interfaces or soft PLC, who manage and coordinate the controlled system on the factory floor, like robots, conveyer belts, assembling station and so on.

It is very important that the state of the controlled system available at the control system has, has to be the actual state of the controlled systems. Otherwise the effects of the controlling system's activities may be disastrous.

Hard real time, what does it mean?

Hard real time operating systems guarantee that critical tasks will complete in a given amount of time. To guarantee this, all delays in the system must be prohibited. In hard real time systems, a process who completes correctly but takes longer than it's scheduled amount of time, fails. In hard real time systems the scheduler plays an key role.

Soft real time, what does it mean?

Soft real time systems are less restrictive than hard real time systems. In soft real time systems critical tasks are given a higher priority than to non-critical tasks. Delays need to be bounded so a critical task do not have to wait forever, but these restrictions are not so severe as in hard real time systems. In soft real time systems, if a process completes correctly but takes longer than it's given time the result may still be useful, depending on the process.

What's the difference between RTOS and a conventional OS?

The key characteristic that separates an real time Operating System from a conventional Operating System is the predictability that is included in all of the described requirements. Conventional operating systems, like e.g. Windows or Linux,. attempt to distribute the resources of the CPU well balanced among threads and processes. This sharing of the resources of the CPU gives all running applications the chance to make a progress, but doesn't allows one process to reserve all resources and to dominate over the rest of applications without the operating system itself. Likewise all priority information will be lost when a system service, normally performed by a kernel call, is being performed. With that behavior the timing of a thread or process is not predictable nor it can be guaranteed that the process will finish in the scheduled time.

In real time systems with micro kernel architecture the processes and threads are managed within the system and allows the communication between them. The scheduling will be performed on the thread level and threads will be scheduled according to their fixed priority. In case of priority inversion the threads will be handled with the new priority adjusted to the micro kernel. High priority threads, becoming the status “Ready-to-Run”, can pre-empt threads with lower priority.

In this operating systems all device drivers and operating system calls are existing as separate processes within the system.

What's the impact to the application?

The typical applications of real-time systems have changed over the years. In the early 1980s, most of the real time applications were used for control applications. Over the time, the number of real time applications increases. Real time Control applications still remain popular. With the development and availability of soft real time systems the solutions using soft real time systems increased steadily. One reason is that hard real time systems cannot support devices if their waiting time cannot be guaranteed. Due to the fewer constraints of soft real time systems these devices can be supported depending on the requirements of the process. This opens the way for additional applications in new areas.

Who needs a real time OS?

Hard real time operating systems were first used in mission critical environments where failure could have serious consequences for person or for the machinery. But more and more, real time systems also being applied in consumer devices, like complex systems with a highly demand on availability, like e.g. in Multimedia applications. This is useful when the application meet a high quality of service guarantees.

2.0 CANopen application using real time OS

The usage of CANopen in a real time project will be shown on an project for an injection moulding machine.

The control system based on two different main systems based on PCs. The first PC is dedicated to the HMI (Human Machine Interface) and the second PC is running a Soft-PLC, responsible for the machine control.

This two PC have to be connected via high-speed communication. The PC-based HMI system is using a graphic interface, developed with Visual C++. The PC performing the machine control can be a mono or a multi processor PC, depending on the complexity of the machine.

For the real time operating system a solutions was searched which offers real time capabilities under Windows NT or Windows 2000. This solution should offer the additional advantage for this project to use standard applications for the Windows environment and to have real time features for the control application running on the same PC. Another criteria for the decision was, that the product has to offer powerful real time features to the developers, all should be in a familiar Windows32 compatible interface, the developer are used to. This was important for the decision because the developer could be continuing to take advantage of the well known products available under Windows NT and Windows 2000. Another request was the availability for a good set of tools and utilities for building and executing real time programs, along with tools for measurement and tuning of the performance of both hardware and software.

Another important point was that the Win32 have to work seamlessly with the real-time subsystem to integrate Win32 and the real-time functionality.

Based on this requirements the decision was made to use RTX, VenturComs Windows real time extension, one of the products available on the market offering such features. This products adds a Real Time SubSystem, known as RTSS, to Windows NT and Windows 2000 systems. RTSS is conceptually similar to other Windows NT subsystems in the way that it supports it's own execution environment and API. RTSS differs in one important area. Instead of using the Windows NT or Windows 2000 scheduler, RTSS performs it's own real time thread scheduling. Furthermore, in a one-processor environment, all RTSS thread scheduling occurs ahead of all Windows scheduling, including Windows-managed interrupts and Deferred Procedure Calls (DPCs).

In the project an CANopen interfaces is used where the RTX driver is supplied with the product. For the project a RTX application was developed that controls directly the CANopen network variables in real time. Moreover, the control of the moulding system is based on three different CANopen networks, controlled by three CANopen interface boards running in one PC, all seen through only one driver.

The advantage of having different network areas are:

- Three different network transmission rates.
- To connect the high speed input/output in a fast network, the less important for administration in a slower network area.

The project requirement is to realize a link with the machine I/O without any limitation of performance in reading input and writing output data. This is another reason of the decision to implement three different CANopen networks, each one managed by one CANopen interface card. The data transmission rate are defined at 500 kbit/s and 1 Mbit/s. The transmission rate are depending on the required total length of the network segment and of the refresh cycle of the I/O.

Various I/O modules are connected on the CANopen network:

- Analogue and digital I/O modules
- Magnetostrictive transducer for position and speed of required modules, like moving plan, ejector, screw and so on.
- Smart devices, each one implements locally the control of 8 temperature zones for cooling and heating

For the CANopen interface a so-called intelligent interface board is selected, to decrease the PC host load from low-level CANopen protocol management. This solution offers the CPU more resources for other tasks and calculation. The usually complex handling of the network protocol has to be done by the interface board. All the standard CANopen functionality is used in this project together with some specific functionality required for this special OEM project.

Analogue data receive synchronously (SYNC) and the transmission is scheduled by remote frame (request by the central unit). For digital input the same functionality is used as for the anlalogue data, and additional in some cases the asynchronous mode.

The CANopen interface board had to be modified to adapt it to the project requirements in the most effective way. All this features has been realized through the creation of a virtual equipment, which will store the control variables: SYNC signal (generically overall the network), and RTR (maximum seven PDO with RTR for each). Due to this configuration, up to three PDOs with RTR can be read within a cycle time of 1 ms.

With this implementation it is possible to ensures ultimate control over the entire process. It provides accurate adjustments of the injection stroke, and optimizes the filling speed and holding pressure thanks to a servo valve system which allows automatic adjustment of both pressure and oil flow. At the machine, a user-friendly control panel will be installed.

The HMI user interface will be done by the customer itself.

3.0 Summary

If you intend to choose a real time system for your application the selection will not be easy. There are dozens of real time system available on the market for different platforms. Finding details on these systems can be difficult, because many manufactures simply state that their operating systems are real time. To verify which real time system will fit with your project requirement will cost you some time and work of verification. A good real time operating system is not only a good kernel. A good real time application should also offer good documentation, should delivery good tools to develop and test your application and also to tune your application. But the best test for your decision will be if you will have the first problems and you will need the support of the manufacturer of the real time system.

As you have seen in the project description even time critical CANopen OEM projects could be realized if your company has the support of all involved partners.

Regarding the communication interfaces you intend to use in your application verify what kind of real time operating systems will be supported by the manufacturer of your interfaces and how he could support you.

If your chosen manufacturer is able to support the most common real time operating systems, like VxWorks, QNX, VenturCom RTX or Real time Linux, as an example you will be sure that he has enough experience with real time application and the development of drivers for real time application. Besides this you don’t have to change the interface products or pay for the development of an driver anytime you have to use a different real-time operating system in one of your next projects.

Rainer Gallus
Woodhead Software & Electronics
Im Gässle 9
70711 LE-Echterdingen
Phone +49 711 782374-15
Fax +49 711 782374-11
E-mail Rgallus@applicom-int.com
www.woodhead.com