By Heinrich Munz,
Managing Director
LP Elektronik GmbH

LP-RTWin Toolkit

The LP-RTWin Toolkit uses a simple hardware expansion module that can be plugged into a PC to achieve Hard real-time deterministic responses in Windows based products. The real-time extension is based on a Non Maskable Interrupt ISR. Therefore deterministic response in the microseconds region can be achieved.

INTRODUCTION

"*Realtime though PC*"

These three words separated sum up the activities of LP Elektronik GmbH, Weingarten. But what do they actually mean?

Realtime

stands, in the widest sense, for measuring and automatic control applications in the industrial sector. LP products can also be used in other fields, however, such as laboratories and medical technology etc., for which LP Elektronik GmbH offers both internally developed products and a range of services.

though

This deliberately rather provocative word refers to the fact that we have developed solutions, or are able to develop new solutions for our customers, which are not available elsewhere. Our "realtime though PC" technology allows the industrial application of PCs on a hitherto unimaginable scale, and could potentially revolutionize the use of PCs for control applications.

The ISRs developed with the LP-RTWin Toolkit are guaranteed to react to react to external events in the matter of microseconds.

PC

Since being launched by IBM at the beginning of the 80's, the Personal Computer has established itself in every possible area of everyday life, both private and professional. Besides office and administrative applications, the PC is now used increasingly for measuring and control purposes in the above mentioned fields. This is due not least to the many advantages which a PC has to offer:

- availability
- standard
- price
- versatility
- wide range of models, e.g. desktop, laptop, industrial or VME PC, etc.
- secure future thanks to Windows NT or PowerPC technology
- connectivity (networks) and multimedia

MS Windows and realtime capability

IBM-compatible Personal Computers (PCs) were already introduced into factories some years ago, where they are now used for controlling machines, visualising processes, measuring electrical and non-electrical variables and processing images. Applications in which the PC has to react to external events (interrupts) within a specified "short" time have not been feasible under MS Windows so far, due to the fact that this graphic user interface does not have a real-time capability and that the necessary short reaction times are not possible.

According to the German industrial standard DIN, a computer system is regarded as having a realtime capability if it can be guaranteed to react to external events (interrupts) within a specifiable time. The standard does not stipulate how long this time must be. From this point of view, Windows can be said to have its own realtime capability on a scale of seconds or minutes. This theoretical statement has no practical relevance, however, as industrial control applications require reaction times within the microsecond range. For example, the typical reaction times of the "hard" realtime operating systems used mainly in the field of VME buses or embedded systems range from 5 to 60 microseconds.

A common solution up to now has been to use a co-processor board (e.g. LP's plug-in board LC20 with 68020 processor), which processes realtime tasks independently of the central PC unit. However, this architecture is too expensive for small to medium-scale control tasks and demand for it is declining due to the ever increasing computing power of modern PC hardware.

In order to convert a central PC unit to realtime, it is necessary to relay interrupts "instantly" to the processor by calling up a suitable interrupt processing program. It should not be relevant here whether the system has just disabled the "normal" interrupts or whether it is currently in the BIOS or operating system routines.

LP Elektronik GmbH has developed a technology which makes this possible. Instant interruption is possible with a simple hardware expansion module which does not require any modifications of the PC motherboard or any add-on boards. The secret lies in a simple programmable logic module (PLD). This module basically serves to switch between Windows and the real-time component. For this purpose, it uses a previously unexploited signal of the PC bus, the NMI (Non Maskable Interrupt).

PRODUCT SPECIFICATION

Features

The LP-RTWin Toolkit is based on the above described additional hardware and consists, in addition to

the hardware, of software components via which the user can utilise the realtime capability of Windows for his own applications. The user does not need any special knowledge of Windows programming, however.

In simplified terms, the product is a tool which can be used to create Interrupt-Service-Routines (ISRs) with realtime capability. In the ISRs, it is possible to access any hardware addresses and data areas of "normal" Windows programs. In addition, messages can be sent from the ISR to Windows programs. This means that control algorithms can be implemented in the ISRs which can also communicate with normal Windows programs if required.

The LP-RTWin Toolkit is a tool for developing hardware-oriented ISRs with realtime capability in the programming language “C” both via and for MS Windows. The developed programs can run independently, but can also interact with “normal” Windows applications, acting as a driver for the hardware to be activated.

The term “realtime capability” effectively means that the ISRs developed with the LP-RTWin Toolkit are guaranteed to react to external events in a matter of microseconds. External events are hardware interrupts triggered via the ISA bus of other PC I/O boards, for example.

"Hardware-oriented" means that the developed programs can access all hardware addresses of the 80x86 address space. This applies both to the I/O area, for all addresses within the first megabyte of address space (DOS real-mode area) and to the entire four gigabyte address space of the 80x86 processor, which also contains the "extended" address space of the ISA bus.

The user-developed programs are translated by the LP-RTWin Toolkit in-to program parts which allow hardware-oriented programming, using the corresponding operating system. In Windows 95 this program parts are the so-called virtual device drivers (VxD), in Windows NT the Kernel-Mode-Diver. This results in a translation of the self developed 32-bit programs which are executed in ring 0 of the 80x86 processor and which use the linear "flat model" as a storage model. Ring 0 is an operating level of the 80x86 processor in which hardware drivers are executed. This means that the user can access the hardware components of his choice directly, a function which can only be achieved by complicated means with normal Windows programs, if indeed at all. In the linear flat model, all processor registers and addresses have a 32-bit capacity and there is no selector offset addressing as in 16-Bit Windows programs. This means that it is also possible to dispense with the tiresome 64K-byte restrictions.

The user does not require any further knowledge of driver programming, however.

The main characteristics of the user-developed ISRs include:

- Guaranteed reaction to external interrupts within microseconds
- Access to any hardware addresses within the entire 4-gigabyte (linear) address space
- Access to all I/O addresses
- Access to data areas of "normal" Windows programs
- 32-bit flat-model programs
- Transmission of messages to "normal" Windows programs
- Runs under Windows 3.x, Windows 95 and Windows NT 4.0

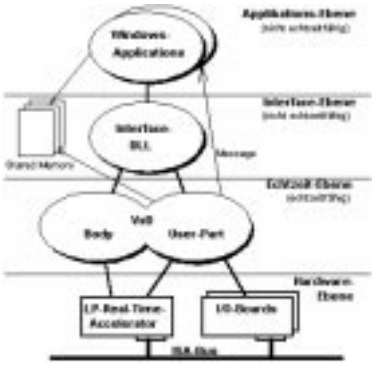

Figure 1. Overview of the components involved

Components of the LP-RTWin Toolkit

Fig. 1 provides an overview of all the components involved on various levels. The hardware layer features the LP-Realtime Accelerator and other I/O boards. These hardware components are activated by the realtime software in the layer abo-ve, and their interrupts can be pro-cessed correspondingly.

The realtime layer is divided into two groups. The body forms the frame and is included in the scope of delivery as a library component. The user part is written by the user in C and then translated using suitable tools from the LP-RTWin Toolkit. Once linked with the body, this produces a loadable Windows driver which contains the realtime user program.

The interface layer is required if the realtime program is to be used in conjunction with "normal" Windows programs or to initialise realtime programs which can otherwise run independently. This takes the form of a DLL (Dynamic Link Library) and serves as an interface between Windows applications and the realtime control program. This allows the Windows application to trigger processes in the realtime program or exchange data with it via the shared memory. These processes are subject to Windows restrictions, however, and therefore do not have a realtime capability, which is limited exclusively to the realtime level. As usual under Windows, this DLL can be called by any DLL-able Windows programs and so communicate and interact with the realtime program.

Hardware layer

The key feature of the LP-RTWin Toolkit is its guaranteed ability to suspend Windows at any time in response to any interrupt and branch within a matter of microseconds into an appropriate handling routine devised by the user. Since MS Windows does not have this property itself, i.e. it does not have a real-

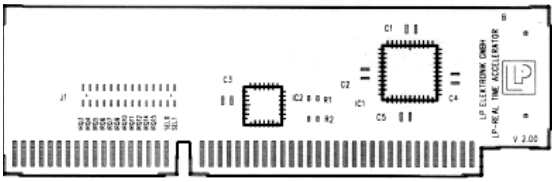

Figure 2. LP-Realtime Accelerator V2.0

time capability, a trick is required in order to get round this problem. This trick involves the use of a small additional hardware item, the so-called LP-Realtime Accelerator. This hardware is based on a programmable module or ASIC and simply requires the normal standard signals of the ISA bus.

There are three different possibilities for implementing the LP-Realtime Accelerator technology:

- The required module is located on our ISA bus-compatible plug-in board (Fig. 2). This allows commercially available I/O boards to be retrofitted with a real-time capability.
- The module (LP-RTAcc chip) can be licensed and used independently, i.e. without a board, for the user's own developments on an ISA- or PCI-bus.

  A corresponding data sheet is available for the chip.
- This possible application of realtime technology involves the use of suitable PC motherboards or backplanes which already have this technology "on board".

Seven of the eleven available interrupt lines of the ISA bus lead to the inputs of the LP-Real-time Accelerator. Any of these interrupts can be selected with the software via the data and address lines, thus causing the system to interrupt Windows immediately via the control signals and branch into the user's ISR.

In addition, the LP-Realtime Accelerator offers the possibility of programming a timer in the accelerator hardware to generate periodic realtime interrupts. The cycle time can be programmed between approx. 7 μ and approx. 27 ms via the software.

Finally, it is also possible to trigger a realtime IRQ via the software. This feature is very useful for switching rapidly from the normal Windows environment to the realtime environment.

This means that there are a total of 9 possible realtime interrupt sources which are prioritized in relation to each other. In other words: if the realtime software is currently processing IRQ 5 and the input for IRQ 6 or higher is set, the current ISR is immediately interrupted and the ISR of the higher-priority IRQ is called. After completing this ISR, the system returns to the lower priority level.

Realtime layer

Windows 3.x and Windows 95

Like any other operating system, Windows also has hardware-oriented device drivers. In the case of Windows 3.x and Windows 95, it is necessary to distinguish between two types, however. On the one hand,

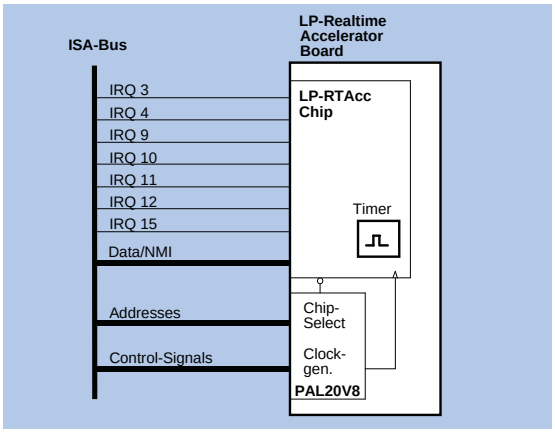

Figure 3. Block diagram of LP-Realtime Accelerator

there are the "normal" drivers (filename ending DRV) which have always been part of Windows. On the other hand, the introduction of Windows in 386 enhanced mode called a new type of driver into existence, the so-called virtual device driver, or VxD for short. In addition to their actual task of virtualizing hardware components for the various virtual machines (DOS boxes and Windows itself), VxDs offer a number of additional possibilities for realtime hardware control. The LP-RTWin Toolkit uses only these possibilities: the ability to virtualize hardware components is not required in this case.

The procedure for creating a new VxD has been largely automated, so that the user does not have to bother with VxD programming and can therefore concentrate fully on his control task. Once an error-free translation has been obtained and linked to a new VxD, the latter has to be loaded. For this purpose, it is necessary to reload Windows 3.1/95.

Windows NT

For Windows NT the procedure is analogous, except from the fact that it is not a VxD, but a Kernel-Mode-Driver, which besides can be exchanged dynamically to the running time.

The newly created realtime program is loaded during the booting process and waits to be activated by a "normal" Windows program, which triggers the initialisation of the realtime program via the DLL. Once an interrupt for the realtime program has occurred, the current Windows status is frozen and the system branches into the user's handling routine. Only when the realtime environment has completed its task Windows can be resumed at the point where it has been interrupted. In other words, although the realtime program can be said to run "under" Windows, it has an infinitely higher priority than other Windows programs.

The realtime programs are created in the form of 32-bit programs and are executed in the flat model. This results in a high working speed and efficiency on the part of the realtime component.

The entire realtime program consists of several parts:

- The body forms the frame of the program and is included in the scope of delivery as a library component. This part also contains the interfaces for the different hardware platforms and Windows-Operating-Systems. These interfaces are created by the respective hardware manufacturer and are not accessible to the end user.
- The user part is written by the user in C and is then translated using the appropriate tools of the LP-RTWin Toolkit. This part contains the user-specific ISR. Once linked with the body, this produces a loadable driver

Interface layer

The interface layer is required in order to allow the realtime program to interact with "normal" Windows programs or initialise realtime programs which can otherwise run independently. The DLL (Dynamic Link Library) serves as an interface between Windows applications and the realtime control program. This allows the Windows application to trigger processes in the real-time program or exchange data with it. These processes are subject to Windows restrictions, however, and do not therefore have a realtime capability, which is limited exclusively to the realtime level. As usual under Windows, this DLL can be called by programs written either in C, BASIC, PASCAL or other DLL-able programming languages. This allows all these Windows programs to communicate and interact with the realtime program.

The DLL-level is also responsible for managing the shared memory zones requested by the application. Once called, the application is given back one or two far pointers to the DLL-allocated memory and is thus able to access this memory. The addresses of the shared memory zones will also ignore the ISRs, so that access to the shared memory is possible from both sides. Both a 16-bit and a 32-bit DLL are available.

Development environments

Any DLL-compatible development tools can be used for developing the non realtime Windows applications (e.g. Visual C/C++, Delphi or Visual Basic). The realtime user part can only be translated with the current version of Microsoft's Visual C/C++ (version 4.0 or higher). This compiler can only run under Windows 95 or Windows NT, so that the development system has to use one of these two operating systems. For debugging, the tool SoftICE for Windows by NuMega is recommended. ■

Heinrich Munz is co-founder and Managing Director of LP Elektronik GmbH. After having terminated his studies as a Radio and Television set mechanical engineer, he successful graduated as an engineer in Electronics at the Fachhochschule Ravensburg-Weingarten.
Immediately after his studies in 1985 he founded together with his partner, Josef Leibinger, an engineering company for software and hardware development for the industrial automation and realtime market. In 1987, the current company LP Elektronik GmbH was founded. LP Elektronik announced in 1996 a revenue of 2,2 Mio of DEM and employs 14 people.