CODESYS Control RTE V3
Manual

Document Version 5.0

tech_doc_e.doc / V1.3

CONTENT

1 OVERVIEW 4

1.1 What is a real time extension? 4

1.2 The System in Details 5

2 INSTALLATION 6

2.1 Important steps of the Setup 6

3 THE USER INTERFACE 9

***3.1 Remote PLC* 9**

***3.2 PLC Configuration* 9**

3.2.1 The tab *Startup* 10

3.2.2 The tab *Component Manager* 11

3.2.3 The tab *Application* 11

3.2.4 The tab *Logger* 12

3.2.5 The tab *Scheduler* 12

3.2.6 The tab *File* 12

3.2.7 The tab *Target* 12

4 AVAILABLE DYNAMIC COMPONENTS 13

4.1 The hardware platform adaption 13

4.2 CmpSJACanDrv 13

4.3 CmpEt100Drv 13

4.4 CmpEt1000Drv 14

4.5 CmpRTL81x9Mpd 14

4.6 CmpRTL8169Mpd 14

4.7 CmpSercos3Master 14

4.8 CmpHilscherCIFX 14

4.9 CmpNetXCanDlDrv 15

5 APPENDIX 16

5.1 Appendix A: The registry entries used by the RTE 16

5.2 Appendix B: Real time behavior on special platforms 17

5.2.1 Check BIOS for suspicious settings 17

5.2.2 Check the settings in Windows 17

5.3 Appendix C: Methods to simply check the real time behavior 18

5.4 Appendix D: Examples for BIOS configurations 19

5.4.1 CELERON Device 19

5.4.1.1 ACPI Configuration 19

5.4.1.2 CPU Settings 20

5.4.1.3 Chipset Settings 20

5.4.1.4 Clock Configuration 21

5.4.1.5 USB Configuration 21

5.4.2 INTEL ATOM Device 22

© 3S-Smart Software Solutions GmbH

5.4.2.1 ACPI Configuration 22
5.4.2.2 CPU Settings 23
5.4.2.3 Chipset Settings 23
5.4.2.4 Clock Configuration 24
5.4.2.5 USB Configuration 24
5.4.3 INTEL DUAL CORE Device 25
5.4.3.1 ACPI Configuration 25
5.4.3.2 CPU Settings 26
5.4.3.3 Chipset Settings 26
5.4.3.4 USB Configuration 27
5.4.4 More common hints on hardware configuration 27
5.5 Appendix E: Startup of a PROFIBUS Master by Hilscher CIFX 50-DP(M/S) 28
5.5.1 Installation of a Windows driver 28
5.5.2 NetX firmware and NetX boot loader 28
5.5.3 Configure the RTE in “CODESYSControl.cfg” 28
5.5.4 Start runtime 29
5.5.5 Create CODESYS project with IO configuration 30
5.5.6 FAQ 33
5.5.6.1 If the fieldbus master is not green, how can I find the problem? 33
5.5.6.2 There are unresolved references 33
5.5.6.3 I’ve activated the Trace, but I do not see a new entry of the CmpHilscherCIFX component 33
CHANGE HISTORY 35

© 3S-Smart Software Solutions GmbH

1 Overview

CODESYS Control RTE is a softPLC system, a PLC in software, on Windows, which is programmable with CODESYS V3. Therefore a powerful PLC within the operating system based on a CODESYS runtime system is available.

Warning: For security reasons, controllers must not be accessible from the Internet under any circumstances. Specifically, the TCP/IP programming port of the controller (usually 1200 and 1210/1211, or the controller specific ports) must not be accessible. The following are the port numbers assigned by default (configurable!) (just the most important for the RTE system):

- **1740..1743 UDP block driver in gateway and runtime system**
- **11740..11743 TCP block driver in gateway and runtime system**
- **1217 Gateway, for the connection from CODESYS to the gateway**
- **8080 Web-Server**

In case Internet access is needed, a safe mechanism has to be used, like VPN.

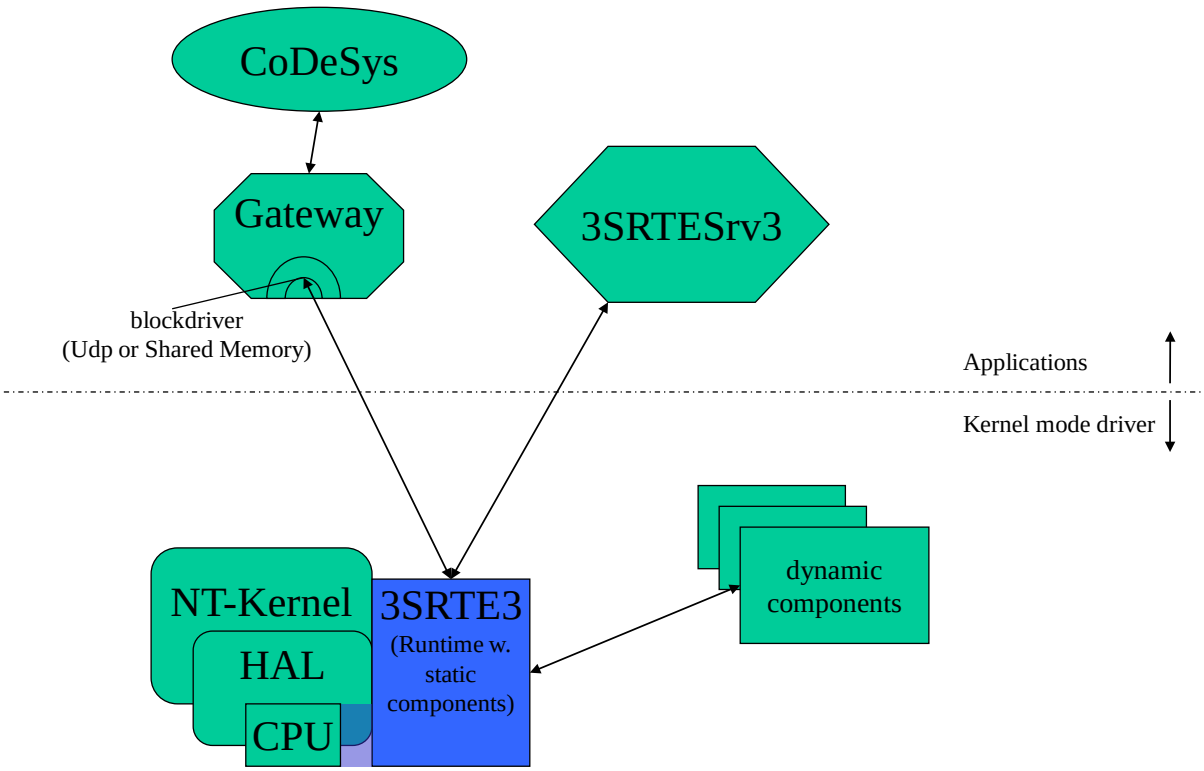
Within this document we assume the reader is familiar with the principles of a PLC and CODESYS. Here the special runtime system running on a Windows PC with hard real time properties, called RTE, is described.

1.1 What is a real time extension?

A real time system is defined by a deterministic time behaviour, with respect to its time response. This means, that if you command a realtime system (by configuration) to call certain routines within a given time schedule, this will happen within given time tolerances, otherwise it will be seen as a failure of the total system. Applied to a controller in the CODESYS environment this means that a task will be called within defined (previously known) tolerances, that is the task jitter will stay within predictable limits.

Experience shows, that these limitations of operating systems basing on Windows NT, are not met, which means that it is no hard realtime operating system. However, the hardware of a PC allows to set up a reliable task scheduling by the means of the software. This real time extension of Windows consists of a driver installing an interrupt service routine, which gets called cyclically by the timer tick of the PC hardware. This routine undertakes the task to call CODESYS-defined tasks and/or to allow the execution of the operating system further on or again.

1.2 The System in Details



The kernel of the real time extension CODESYS Control RTE of 3S-Smart Software Solutions GmbH consists of a user mode system service and a kernel mode driver. The standard hardware of the PC is used to call the task scheduler when needed. The scheduler can call own tasks and activate the time slice of the operating system. So the tasks of the PLC get interrupted by the operating system in fix intervals, once per millisecond form the operating system, with a configurable time scheduling. If the RTE tasks get executed on an own CPU kernel (this feature can be installed as from V3.5.3.0), the scheduler of the RTE will activate RTE tasks.

The dynamic components are inserted into a list in the configuration file of the runtime. These components provide the component interface to become a part of the CODESYS runtime. After being loaded by Windows, they can directly exchange data and calls with the runtime system.

It is not necessary to change the Windows kernel.

The RTE consists of runtime components that are loaded together with the RTE and some that are loaded “on demand”. What dynamic components are available and what functionality they cover is described later in this document.

Note:	The common properties of a 3S runtime system (means a 3S PLC) are described in the document „Runtime System Overview“ of 3S-Smart Software Solutions GmbH. There all variations and peculiarities of a runtime system are explained. So this technically detailed document is designed for development work.

2 Installation

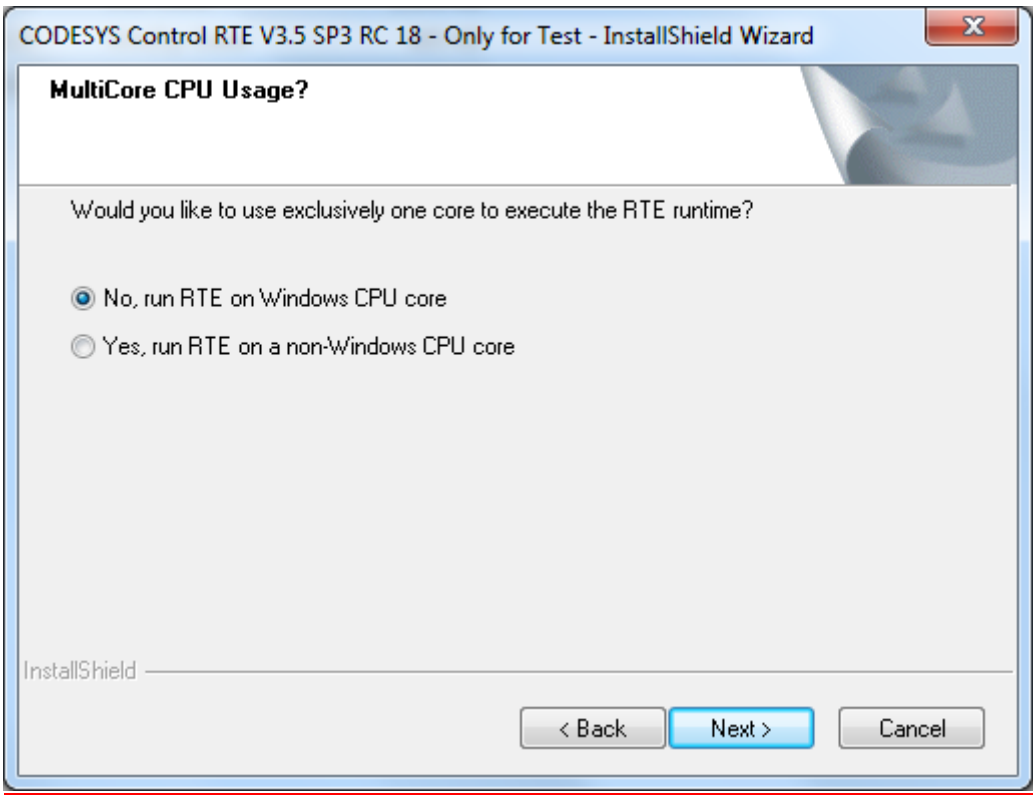
The installation of CODESYS Control (respectively Softmotion depending on the type of the runtime system) RTE V3 - in the following just referred to as "RTE" - is done by the Setup. Depending on the type of delivery, the setup is available as a single exe-file or in an archive file or on a CD/DVD.

The Setup installs all available drivers provided with the delivery.

The Setup detects the type of platform on which the system gets installed, mainly it detects whether it is an (old, hardly any more used) AT-compatible platform, or a system basing on the newer APIC standard. Depending on the platform, another component gets inserted in the file CodesysControl.cfg.

As from version V3.5.3.0 it is also possible to run the RTE on an own (reserved and then not any longer usable by Windows) physical CPU core. In this case the Setup inserts the "AMP"-platform adaption:
`Component.1=CmpDrvSchedulerAMP`
(AMP means "asymmectric multiprocessor system")

2.1 Important steps of the Setup

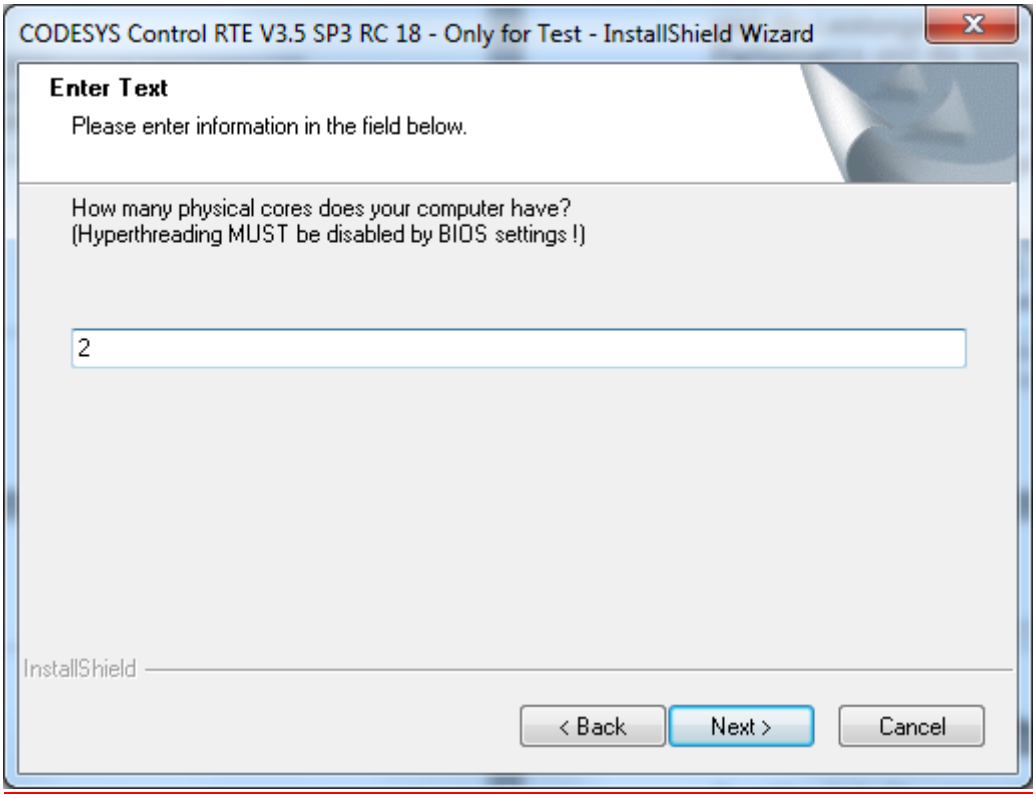


After you have started the Setup and run through some standard steps, this dialog will appear on systems with multiple CPU kernels or on systems, where it is not clear whether there is only one or several CP kernels. At this time you must know whether the system indeed has multiple CPU kernels and whether you want to use one of them exclusively for the RTE. This kernel then will not any longer be available for Windows and thus the performance of Windows will be reduced. However, of course at the same time the performance and temporal precision of the RTE will be increased.

If you choose the option "No, run RTE on Windows CPU core", the RTE will be installed in a way, that - like before in all versions of the RTE - the tasks of the RTE will be executed together with those of Windows on the first kernel and will share that kernel with the operating system in fix time slices.

If you choose the option "Yes, run RTE on a non-Windows CPU core", the RTE will be installed in a way, that the RTE tasks will run on an own CPU core which is reserved for this purpose only. In this case subsequently the following dialog will appear:

© 3S-Smart Software Solutions GmbH



Here you must regard the following:

- The number (in the example shown above: "2") is only a proposal of the setup. It is not always possible to detect the exact number of physical CPU kernels available in your system. So now you should know that number. Regard also that a CPU kernel, reported by Windows, could be a "logical" CPU kernel, not suitable for running a realtime system. (These CPU kernels i.a. are known as "hyperthreading kernels".)
- The dialog makes you aware that you should deactivate any features like hyperthreading by all means via the system configuration, because experience has shown, that they have bad influcence on the realtime behaviour.

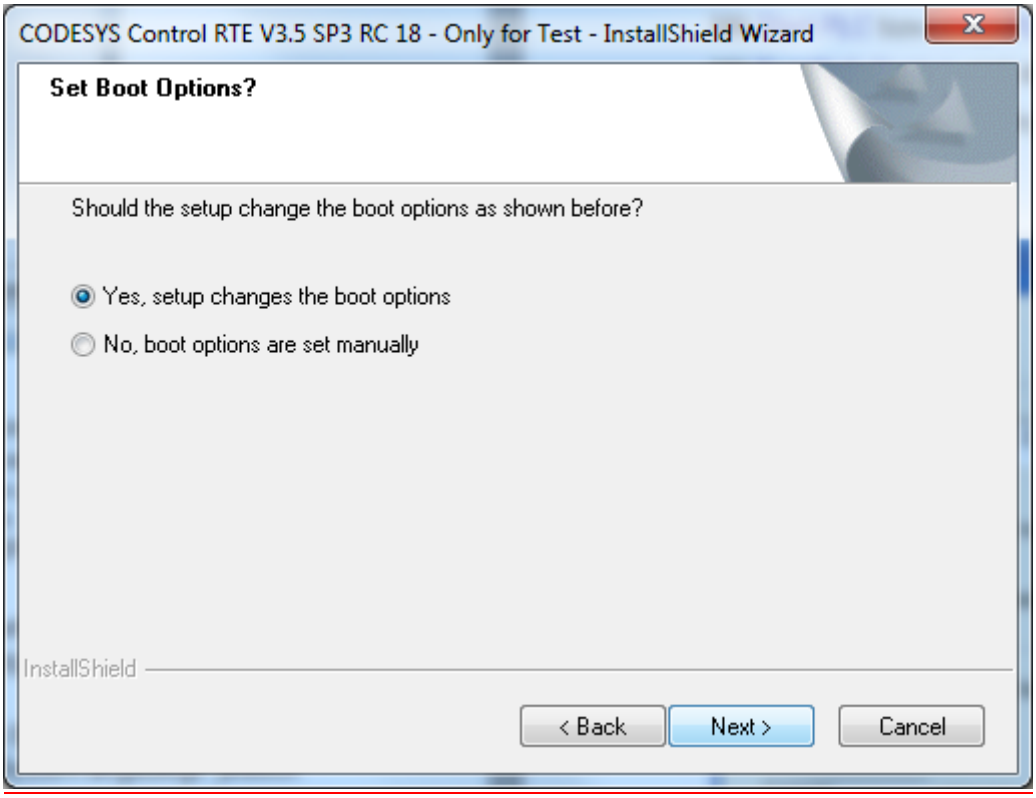
Later on the Setup will try to reserve a CPU kernel for the RTE by modifying the start configuration of the system. In the subsequent dialog this modification will be indicated, however not yet done.

Only after the next dialog, where you get asked whether the boot configuration should be changed by the Setup, actually the modification of the system start configuration will be done.

The modifications are (see also the text in the previous setup dialog):

- Limitation of the number of CPU kernels available for Windows, via the boot option "numproc", to <number of physical kernels> - 1.
- Deactivation of the processor feature "Data Execution Prevention".
- Switching off the processor feature "physical address extension (PAE)".

Note:	The processor features "Data Execution Prevention" and "PAE" not necessarily must be switched off. Just as well you can limit just the number of kernels for Windows by a manually defined start configuration. However 3S-Smart Software Solutions GmbH does not test such configurations!



The further Setup steps are standard steps. The RTE will be installed with the chosen options.

3 The user interface

Access to the user interface of the CODESYS Control RTE is obtained via an icon in the systray. The tray icon offers the possibility to edit some of the settings in the configuration file.

With a click on the icon of the RTE a menu is displayed. It consists of the entries

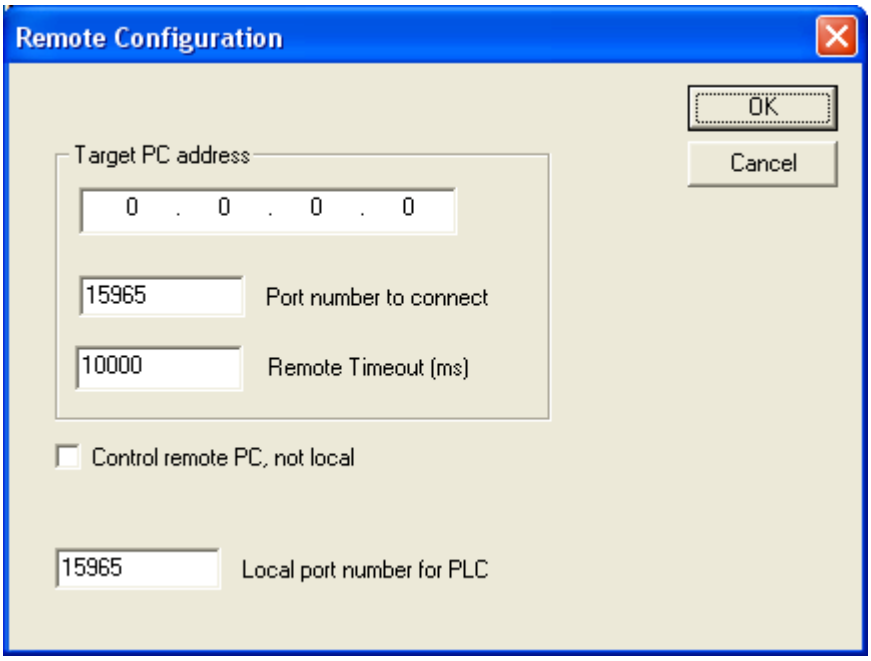
- *Start PLC*
- *Stop PLC*
- *Exit PLC Control*
- *PLC Configuration*
- *Remote PLC*
- *About…*

With *Start PLC* resp. *Stop PLC* the PLC is loaded respectively unloaded. This is equivalent to turning on/off power on a “real” PLC.

With *Exit PLC Control* just the tray icon disappears, the PLC keeps on running.

The *About …* dialog shows the icon and the version of the RTE.

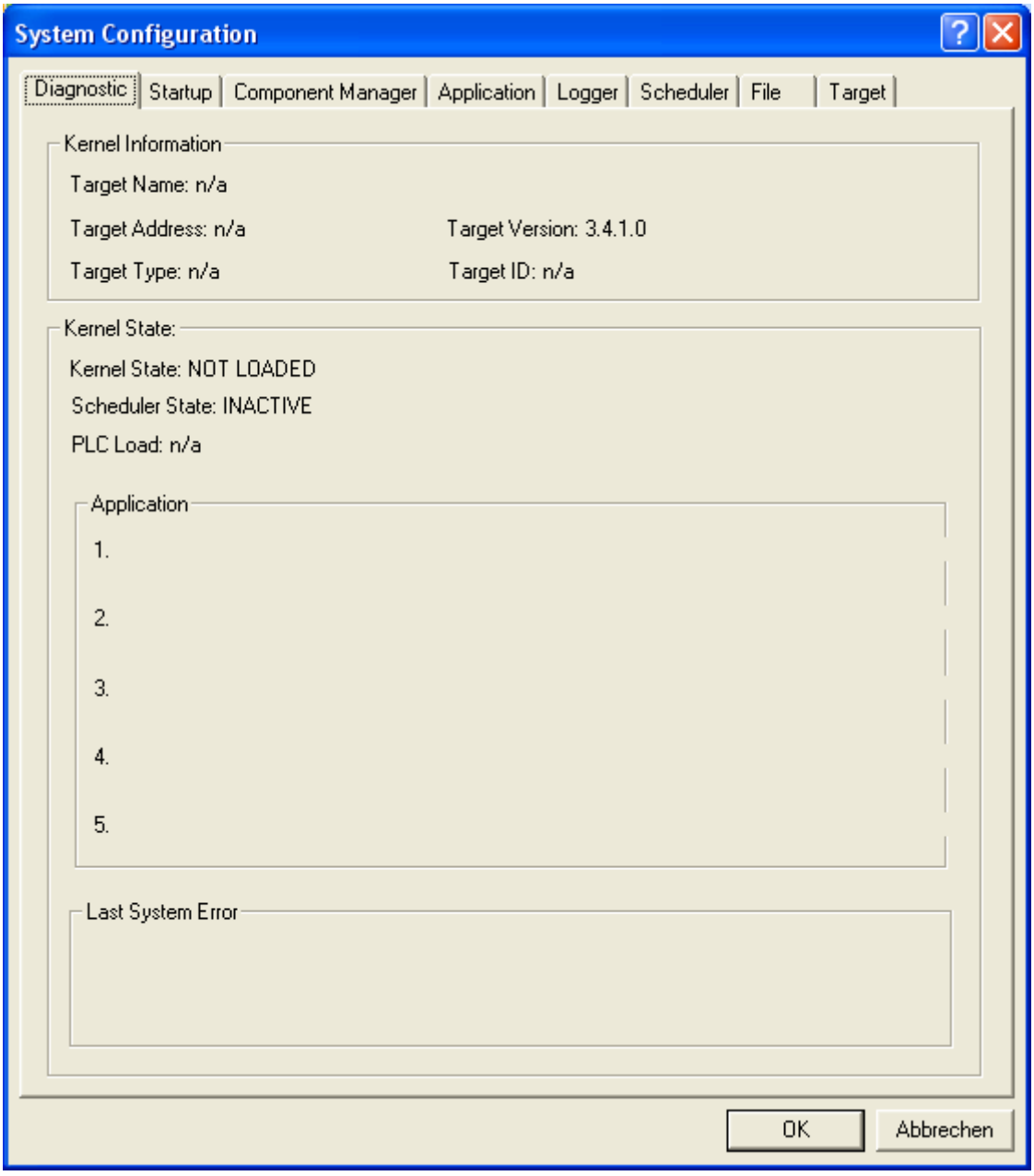
3.1 Remote PLC



With the menu item *Remote PLC* a dialog appears to determine whether the local RTE or a remote one shall be configured. The entire configuration done will then affect the locally installed RTE or the one running on the system with the IP-address *Target PC address*.

3.2 PLC Configuration

With *PLC Configuration* the configuration dialog of the RTE is shown, which allows to view some states and edit some common configuration settings.



The dialog appears with the tab *Diagnostic*.

Some states and IDs of the RTE are displayed here.

3.2.1 The tab *Startup*

In the tab *Startup* three options are available:

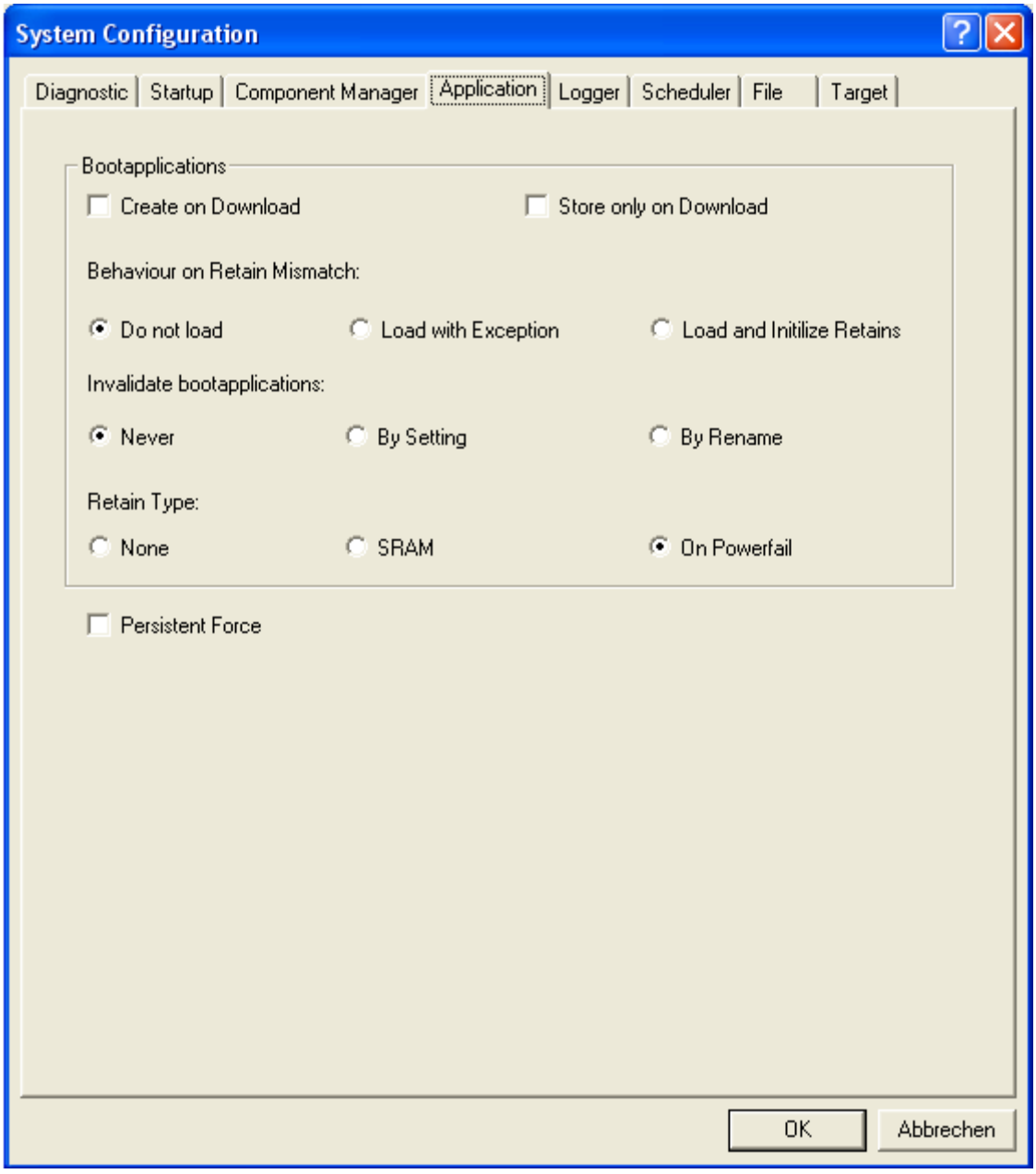
- *Start the RTE basic system service when booting*:
  Enabling this option leads to an automatic start of the system service („3S RTE Service V3“) of the RTE, no matter if a Windows user is logged on or not.
- *Start the PLC with the basic system service*:
  With this option the RTE is loaded (means, the PLC is started) together with the system service. Together with the option above, this leads to a PLC start at boot time of the PC.
- *Do not load boot applications:*
  This option prevents loading the boot application automatically. This is sometimes required for test purposes.

3.2.2 The tab *Component Manager*

This is made for adding dynamic components. Some of the available dynamic components are available in the drop down lists, some have to be entered manually to the lists by typing the name into the edit control.

3.2.3 The tab *Application*

Some settings for the application component can be edited here.



The options *Create on Download* and *Store only on Download* specify if the boot project is created implicitly on download. The additional option *Store only on Download* leads to not loading the project into memory. With this option enabled, the boot project is stored in the file system only and loaded when the PLC is started the next time.

The options for the handling of corrupt retain data specify how the PLC shall behave in these cases. With *Do not load*, the boot project is not loaded. With option *Load with Exception* the boot project will be loaded but not started. With the option *Load and Initialize Retains* the boot project will run with all retain data being reset to their initialization values.

To make sure the boot project is loaded only in case the PLC terminated correctly the last time, the possibility exists to invalidate the boot project after it has been loaded. The option *Never* prevents this.

The option *By Setting* leads to an entry in the configuration file that invalidates the boot project. With the option *By Rename* the app-file will be renamed.

The options under *Retain Type* specify how the RTE stores the retain data. *None* will disable the use of retain data. *SRAM* should not be set by the user directly. As this is hardware dependent, refer the hardware's manual or let an appropriate vendor specific component do this implicitly. The default retain handling of the RTE is *On Powerfail*.. In this case the retain data will be saved to a file if the controller is shutdown. To enable this, the system must be equipped with a UPS. Then potential power failures are taken into account, too.

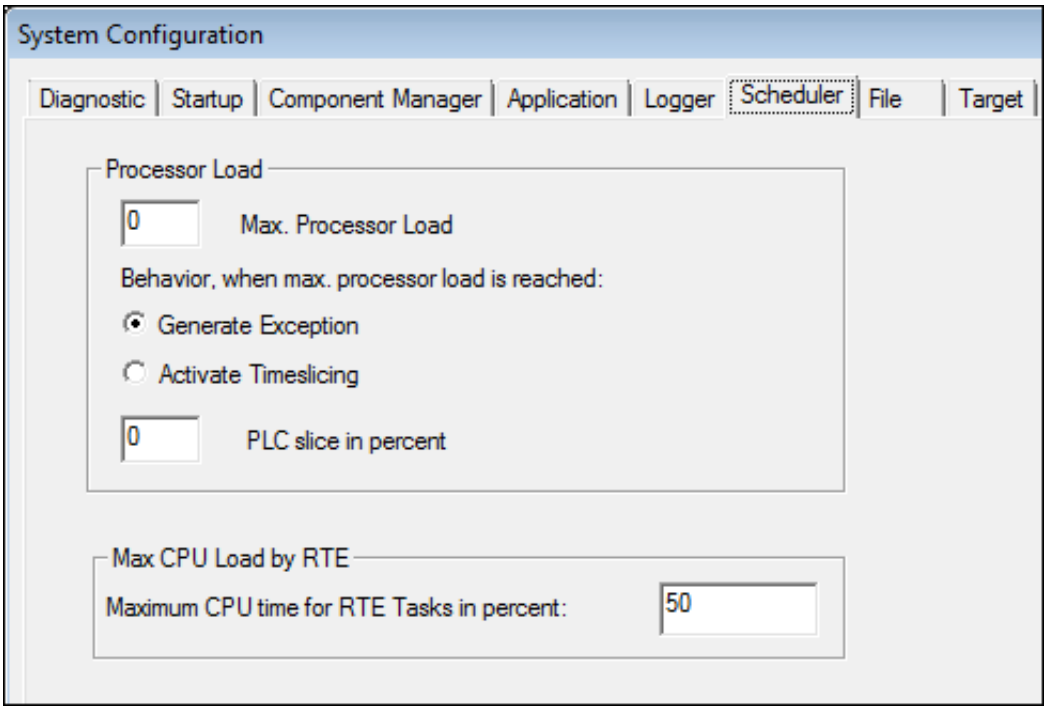
3.2.4 The tab *Logger*

Logging of log entries to a file is configured here.

Using the default values *5 Files*, *50000 Max File Size*, *1000 Max Entries*, for example, 5 files with a maximum size of 50000 Bytes or 1000 entries are created. The eldest file will then be overwritten. With a number of 0 file, everything is logged to files, no files are overwritten.

3.2.5 The tab *Scheduler*

The task scheduler within the PLC can also be configured. These entries are necessary in special cases only. You can provide settings here, for example, to prevent some task being no longer scheduled in case other tasks with a higher priority consume all the CPU time. You can also determine how the system should behave in case of such problems.



Max CPU Load by RTE: Default value is 50. If you change this value then the entry *MaxPLCTime* in the registry data base is set with the value entered here. This change will result in an adjustment in the CPU timing according to the value. This may be necessary if the RTE task is interrupted too often by Windows system tasks or vice versa. This option will be without any effect if the RTE runs on an exclusive CPU kernel.

3.2.6 The tab *File*

The file system of the PLC offers the feature to store files with a configurable extension to dedicated directories automatically.

3.2.7 The tab *Target*

By default, the RTE appears with the name of the computer in the PLC scan. The name can be configured here to a fixed name.

4 Available dynamic components

Optional drivers like PCI card drivers for field busses are available as IEC libraries or runtime components, where in case of the RTE these drivers are sys-files. If a driver is a component, it has to be entered to the configuration file, to make the runtime loading it. One can enter this by using the configuration dialog or a text editor.

The configuration file of the RTE is to be found at “C:\Program Files\3S CODESYS\CODESYS Control RTE3\CODESYSControl.cfg“ in case of standard installation.

Alternatively, you can also use the Tab *Component Manager* to tell the runtime system, which components to load at startup in addition to the RTE.

4.1 The hardware platform adaption

There are mainly two PC platforms available nowadays. The AT compatible, rarely used today and the newer one, the APIC platform. Both platforms are supported by the RTE. The necessary adaption to the platform is contained in the drivers CmpDrvSchedulerAT and –APIC. The setup detects the platform and the driver is entered as “Component.1” into the configuration file by the setup always. This component must not be changed ever.

These drivers store under their name in the registry path of the system services (HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services) in the key *Param* their measurements concerning the hardware speed at the first start. On any following system starts, these measurements are used. (In case the values are missing, a new measurement is initiated.)

4.2 CmpSJACanDrv

This driver is a generic driver for SJA1000 based passive CAN cards or on board hardware. The PCI cards by Peak, Ixxat and Automata are supported instantly. When installing such a CAN PCI card, one can use the appropriate driver from W2K_XP_Drivers, a subdirectory in the RTE installation path.

The CmpSJACanDrv can also use on board SJA1000. The properties of such hardware have to be configured in the cfg-file, in the CmpSJACanDrv section. Possible entries are:

```
NumDevices=1 <number of following controllers>
(optional: <DisableRetain=1> You can tell the driver, not to use possibly
available NVRAM as retain memory.)
0.Address=<physical address of the SJA1000>
0.MapAddress=1 <always 1 on a PC>
0.Interrupt=<Interrupt Vector of the SJA1000 >
0.IntEdgeTrig=<0 or 1. 0 hardware dependent>
0.XtalFrequency=<normally 16000000Hz, 24000000Hz is also possible,
depends on the hardware.>
0.BusType=<type of connection of the SJA to the CPU, 1 for ISA, 3 for
PCI, 0 for memory>
0.Alignment=<normally 1, hardwaredependent>
0.Name=<any name>
0.OutputCtrl=<valid values are for example 0xFE, 4C, 1A and others.
Hardwaredependent.>
0.PortAddress=<A portaddress the driver will once on startup write to.>
0.PortValue=<The value to write once on startup.>
```

4.3 CmpEt100Drv

This driver supports Pro100-compatible Intel Ethernet controllers. To use this component, the Windows network driver for this Ethernet controller has to be superseded by the one shipped with the RTE. To manage this, do the following:

- Open the Windows device manager.
- Choose *Update driver* from the context menu of the appropriate network card.
- Use *Manual Driver Choice* and choose the one in *<RTE-Installationdirectory>\CmpET100MPD*.

© 3S-Smart Software Solutions GmbH

In case the device manager refuses to install the driver, first uninstall the Intel driver and then delete the associated inf file from <SystemRoot>\inf. (SystemRoot is often c:\Windows) (The associated inf files refer to Pro100 and are normally called net557*.inf/pnf or net559*.inf/pnf. The pnf file is called like the inf file and has to be deleted also.) After the following reboot of the system, Windows is no longer able to automatically install the driver and the hardware wizard will appear. Now the driver from <RTE-Installationdirectory>\CmpET100MPD will be accepted.

4.4 CmpEt1000Drv

The installation of this driver is performed analogue to 4.3, but support Pro1000 compatible Ethernet chipsets by Intel.

4.5 CmpRTL81x9Mpd

This driver supports the Realtek RTL8139 (and compatible) Ethernet controller. The combi-chipsets 8100 and 8110 are included also.

The installation of this driver is performed analogue 4.3 also.

4.6 CmpRTL8169Mpd

Like 4.5, for the Gigabit capable Realtek controller RTL8169 and compatible chipsets, like 8111.

4.7 CmpSercos3Master

This driver supports the SERCOS III PCI card 700353x0 of Automata. After installing the hardware, used the driver from W2K_XP_Drivers\Automata_SERCOS3 directory in the hardware wizard.

4.8 CmpHilscherCIFX

This component supports the CIFX-cards, manufactured by Hilscher.

After the hardware installation use the driver shipped with the card or the SysDrv3S, which comes with the setup of the 3S soft PLC “CODESYS Control V3”, the soft-PLC without real time extension by 3S-Smart Software Solutions GmbH. The component has to be entered to the configuration file and additionally needs a section `[CmpHilscherCIFX]`.

The driver needs the entries

```
Device.0.BootloaderFilePath=NXCIF50-RTE.bin
Device.0.Channel.0.FirmwareFile=cifxdpm.nxf
```

Here in the example, 0 is the zero based index of the card, in case more than one is used. `Channel.0` has to be used always for CIFX PCI (or PCIexpress) cards.

The files referenced here normally are shipped with the card and have to be placed in the root file path of the RTE. The names of the files (or the entries in the configuration file) have to be adjusted.

See chapter 5.5 “

Appendix E: Startup of a PROFIBUS Master by Hilscher CIFX 50-DP(M/S)".

4.9 CmpNetXCanDlDrv

This component is a CAN mini driver, to support the Net-X (or CIFX) CAN cards natively. With this driver, the can interface of the CIFX CAN cards can be used with the 3S CanOpen libraries.

This component can only be used together with the CmpHilscherCIFX component, which is required to perform the firmware download to the card. A CAN firmware (master or slave) is required to make this component work. Then the CAN card is used as a native CAN interface, the protocol is completely managed in the libraries.

The component is configured with the following entries.

```
[CmpNetXCanDlDrv]
MaxNumberOfCanNetworks=0..n-> number of CAN networks (number of NetIds)
NetId.0.Channel=0    -> mapping of the NetId to board and channel of the
NetX
    -> see CmpHilscherCIFX
NetId.0.Board=0
```

5 Appendix

5.1 Appendix A: The registry entries used by the RTE

The behavior of the RTE is partially configurable in the Windows registry database. The following table lists all these values and shows the meanings and the default values. Only a part of these values can be edited using the configuration dialog.

All values are located in *HKEY_LOCAL_MACHINE\SOFTWARE\3S-Smart Software Solutions GmbH\CODESYS SP\3SRTE3*. (Or in a relative path, if a key is given in front of the name.)

Name	Type	Description	Default
AutoStartPLC	DWORD	0 or 1: With 1 the PLC will be started automatically by the system service.	0
AllowSetAppConfiguration	DWORD	0 or 1: Sets the write access to the tab *Application*.	1
AllowSetCMConfiguration	DWORD	0 or 1: <analogue the entry before>	1
AllowSetFileConfiguration	DWORD	0 or 1: < analogue the entry before >	1
AllowSetLoggerConfiguration	DWORD	0 or 1: < analogue the entry before >	1
AllowSetSchedConfiguration	DWORD	0 or 1: < analogue the entry before >	1
AllowSetTargetConfiguration	DWORD	0 or1: < analogue the entry before >	1
CallBuffersize	DWORD	Size of the buffer for asynchronous calls. (All functions performed by Windows like file services, socket functions …) The more simultaneous calls an application needs, the bigger this buffer should be, for performance reasons.	65536
CfgFile	STRING	This value points directly to the configuration file of the RTE.	<Installation-path of RTE>
DisablePlcAutoStart	DWORD	This value is used internally by the configuration dialog. It may be edited manually only in case of problems, for example in case the configuration dialog crashed and the variable stays 1 for ever.	0
ShowAppConfiguration	DWORD	0 or 1: Defines if the tab *Configuration* is shown.	1
ShowCMConfiguration	DWORD	0 or 1: < analogue the entry before >	1
ShowFileConfiguration	DWORD	0 or 1: < analogue the entry before >	1
ShowLoggerConfiguration	DWORD	0 or 1: < analogue the entry before >	1
ShowSchedConfiguration	DWORD	0 or 1: < analogue the entry before >	1
ShowTargetConfiguration	DWORD	0 or 1: < analogue the entry before >	1
LastExit	DWORD	<unused, reserved>	1
Path	STRING	This value contains the root path of the RTE file system.	<Installation-path of RTE>
SysTrayLocalPort	DWORD	TCP/IP port, used for the communication between the RTE and the tray icon.	15965
SysTrayRemoteAddr	DWORD	IP address of the remote system to be managed by the tray.	0

Name	Type	Description	Default
SysTrayRemotePort	DWORD	Like SysTrayLocalPort, but for the remote system.	15965
SysTrayUseRemotePLC	DWORD	0 or 1: Defines of the local or the remote PLC is the one to be managed by the tray.	0
KernelVersion	STRING	Version of the RTE, which was started the last time on this PC.	<Version string>
VersionSrv	STRING	Version of the system service of the RTE. Should be the same as the kernel version, may be different just in case of manual patches or hotfixes.	<Version string>
IO Driverpool	KEY	All subkeys in this key are displayed in the Drop-Down-List of the available components in the configuration dialog.	
StackSizePerTask	DWORD	Here you can define the stack size for each task if necessary (unexplainable crashes at complex function calls)	0x6000 (24KByte)

5.2 Appendix B: Real time behavior on special platforms

The RTE is dependent of the constancy of the CPU frequencies and the clock of the so named front side bus. In the following sections some settings in BIOS and in Windows are listed, which can influence the time behavior of the RTE. Also, methods are presented as one can check real-time behavior.

Note:

The RTE runs on the two platforms, AT compatible PC and APIC compatible PC (uni- and multiprocessor). The general adaptation on the specifics of the platforms is done in CmpDrvSchedulerAPIC und –AT. On the APIC platform, as usual on all modern platforms, a complex measurement is carried out to scale the clock frequency in microseconds at first start. The result of the measurement is stored in the Windows registry. See HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\CmpDrvSchedulerAPIC\Params, the values SetClocksPerUS and SetCountsPerMS.

If the values are set, no further measurement is done, when the RTE starts. When the system changes in a way that the clock frequency changes as well, then the values of the measurement have to be deleted. Then at the next start the clock frequency has to be scaled again. Below are described activities like changing of power save settings that lead to the fact that the clock frequency changes.

5.2.1 Check BIOS for suspicious settings

As most of all hardware platforms use different kinds of BIOS, we can show a brief overview here only. The mentioned options may be not at all or under a different name available in your BIOS.

- Everything that leads probably to a System Management Interrupt, like a self check of the BIOS, „USB Legacy Device Support“ and other things like that should be disabled. During a system management interrupt the system will not react at all and these interrupts may last some milliseconds (too long for the RTE).
- All energy saving options, like C-States or Intel Speed Step, should be disabled.
- MWAIT, if configurable, should be disabled. A HLT instead is no problem.

5.2.2 Check the settings in Windows

The active power scheme in Windows should be called *Always on*. Any energy saving mechanisms will lead to frequency throttling.

Automatic Updates should be disabled for industrial use. Otherwise the PC may for example reboot when you don't expect it.

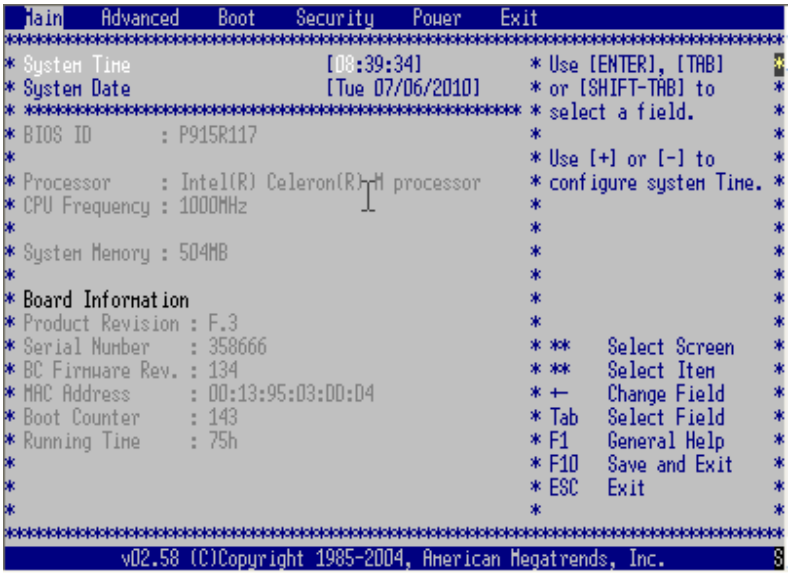
5.3 Appendix C: Methods to simply check the real time behavior

- Create a simple project containing a cycle counter. Open the task configuration and set the task cycle to be 1 ms. Run the project on the RTE and check, if the thousands increment once a second. If this is true, the time base is ok. Then check in the monitor tab of the task configuration for the jitter. If the hardware is good, it will range between +-100us.
- Add the library "SysPort" to your project and trigger a pin of the parallel or serial port. Then watch it with an oscilloscope.
- In case your hardware offers Ethercat support, means an Ethernet chipset supported by the RTE, you can also watch a digital output of an EtherCAT slave with an oscilloscope.
- Of support@3s-software.com one can obtain a project to determine the statistical distribution of jitter times.

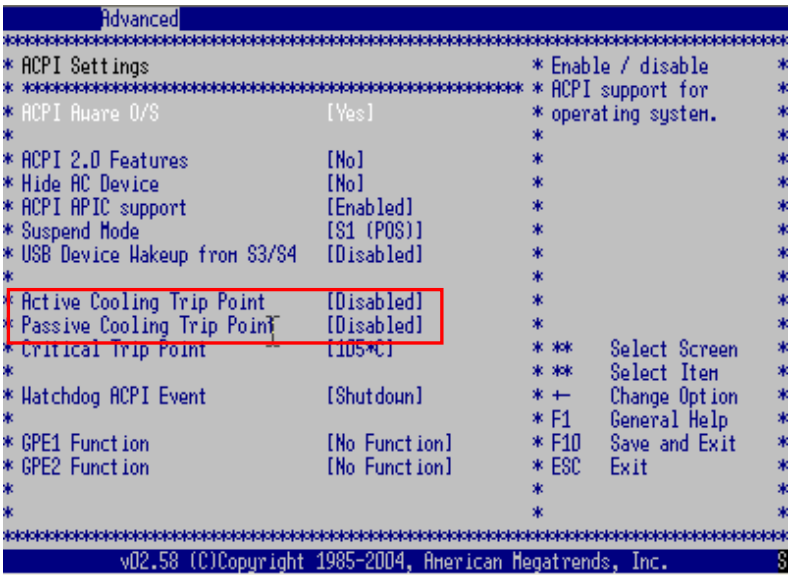
5.4 Appendix D: Examples for BIOS configurations

To make the RTE run on different platforms, we had to adapt some BIOS settings in the past. Here some examples are shown. This reflects, what we had to do on some platforms and can only be understood as hints.

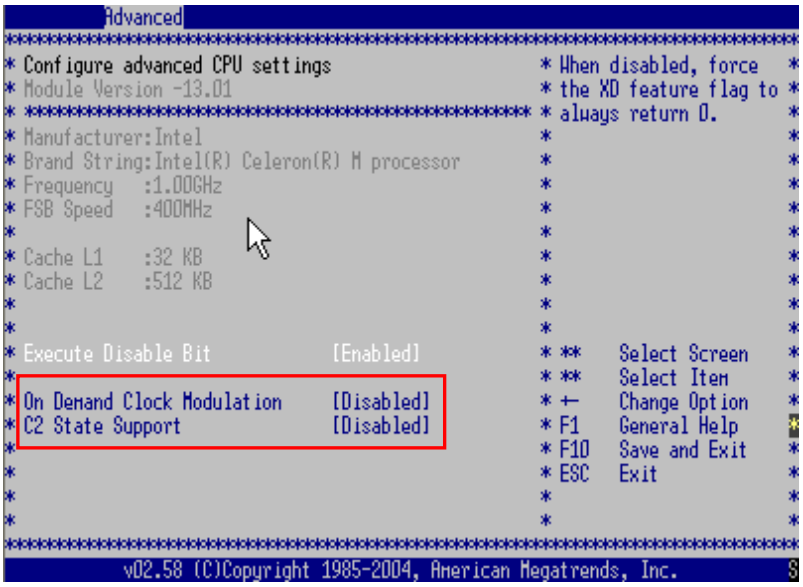
5.4.1 CELERON Device



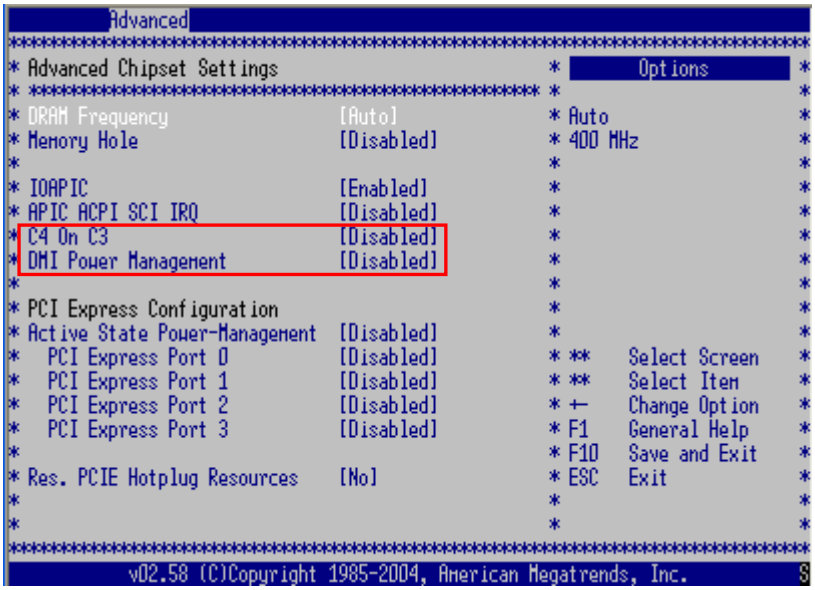
5.4.1.1 ACPI Configuration



5.4.1.2 CPU Settings

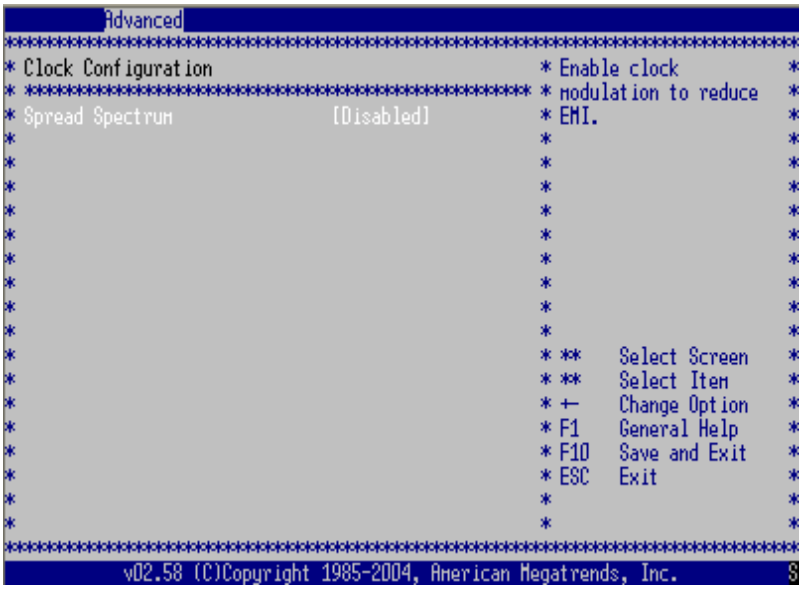


5.4.1.3 Chipset Settings



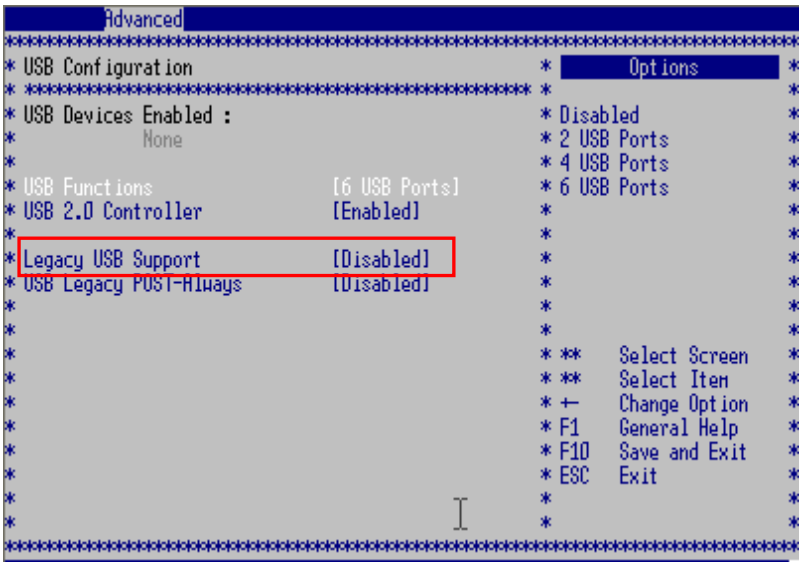
Enable APIC ACPI SCI IRQ ! (not correct in this screenshot!)

5.4.1.4 Clock Configuration



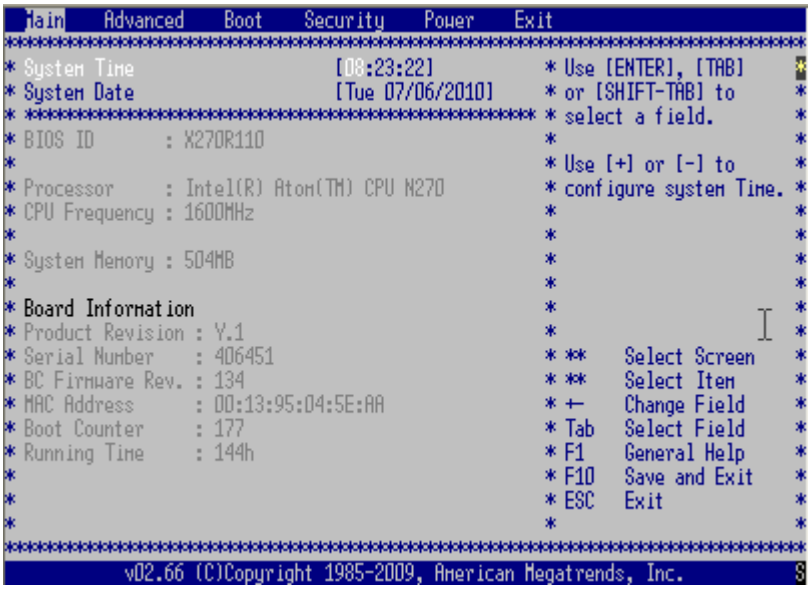
5.4.1.5 USB Configuration

“Legacy USB Support” should be deactivated on most platforms!

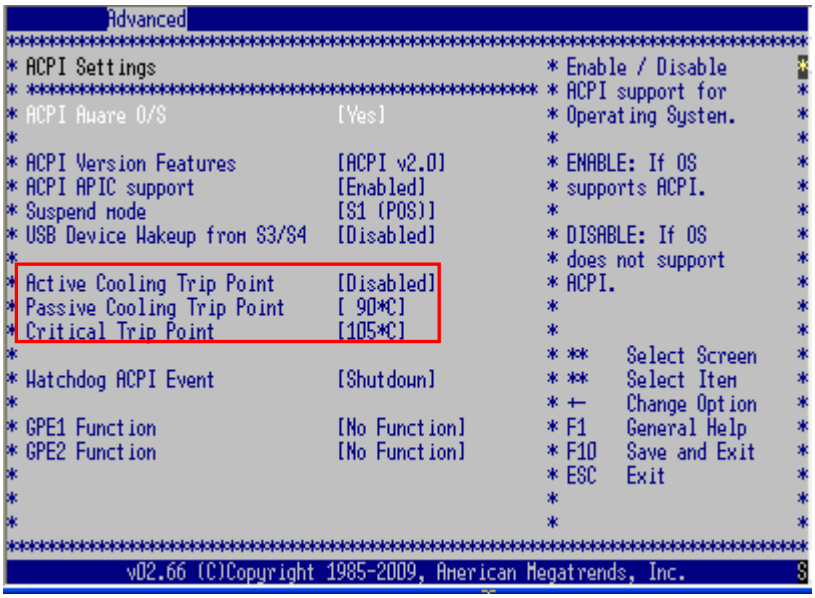


5.4.2 INTEL ATOM Device

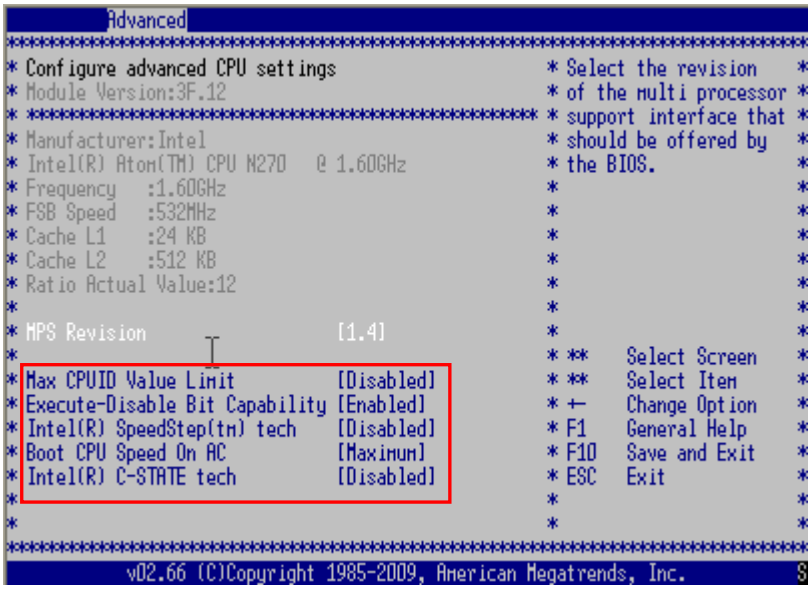
BIOS Settings



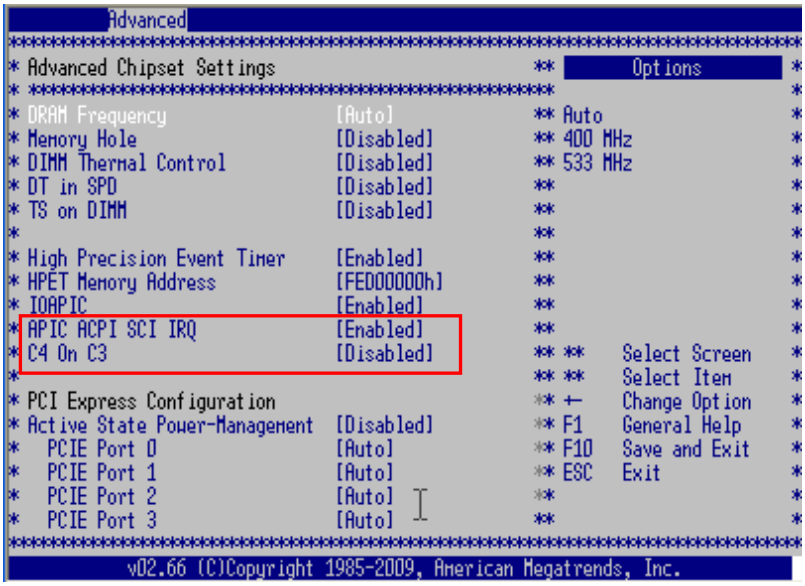
5.4.2.1 ACPI Configuration



5.4.2.2 CPU Settings

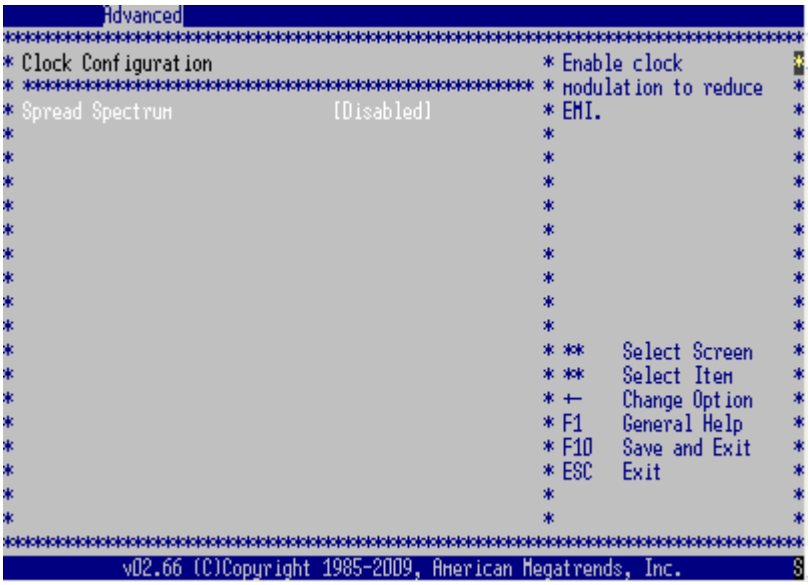


5.4.2.3 Chipset Settings



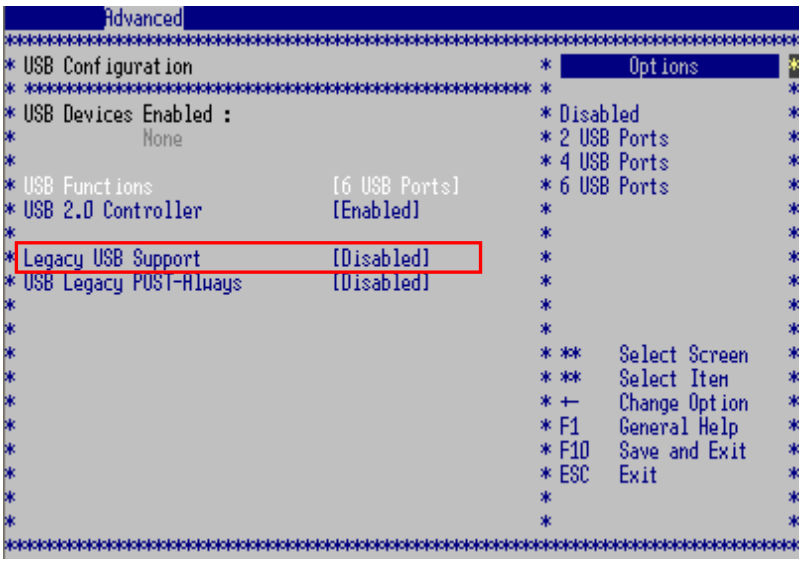
In any case enable APIC ACPI SCI IRQ !

5.4.2.4 Clock Configuration



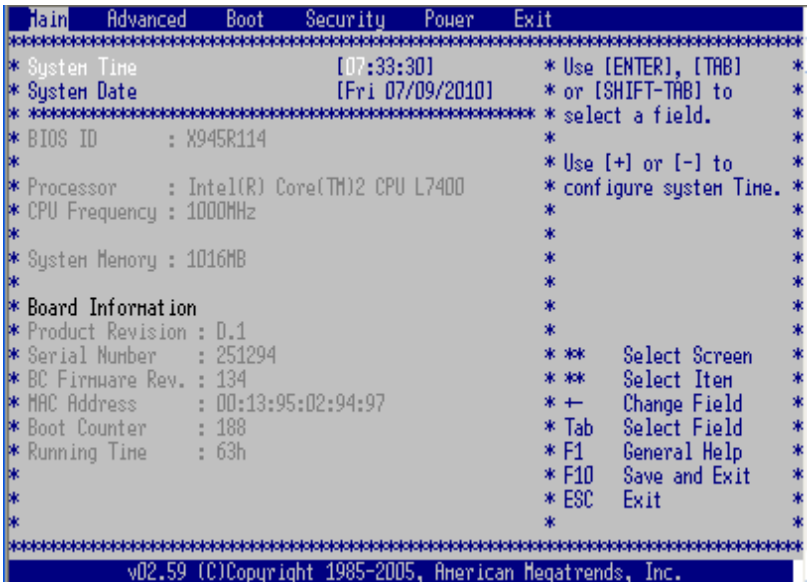
5.4.2.5 USB Configuration

"Legacy USB Support" should be deactivated on most platforms!

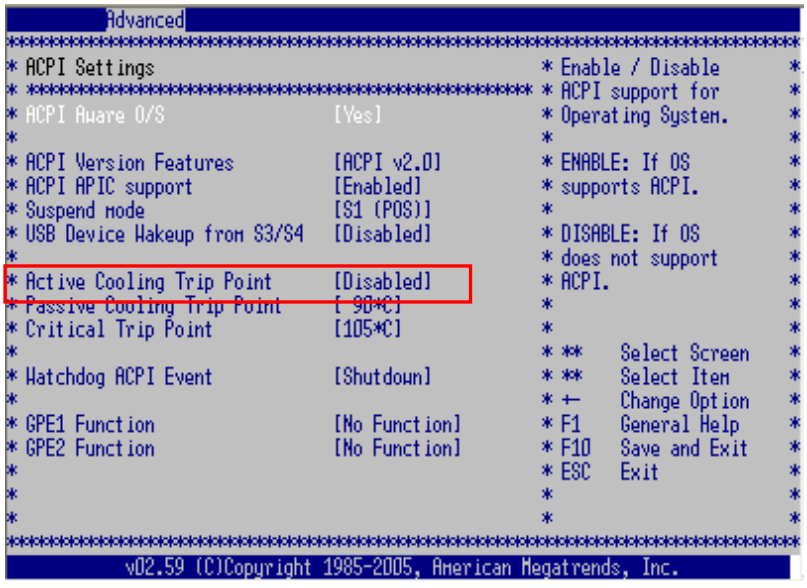


5.4.3 INTEL DUAL CORE Device

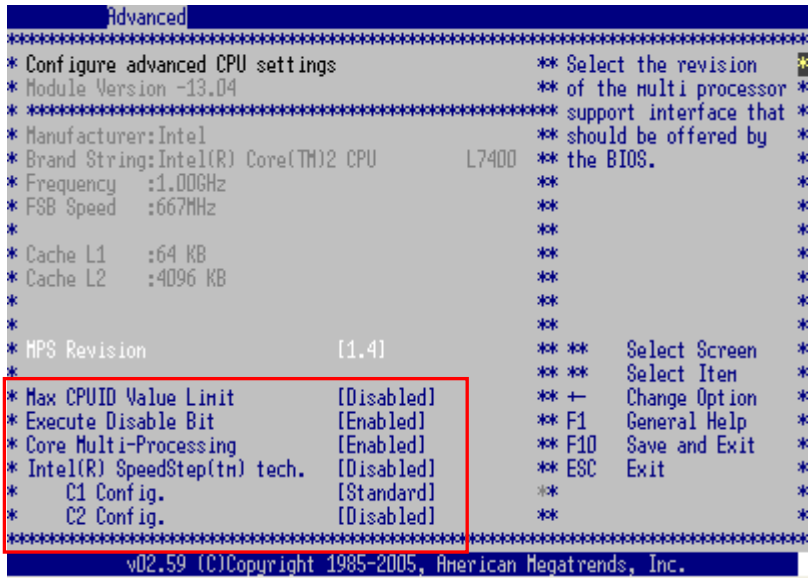
BIOS Settings



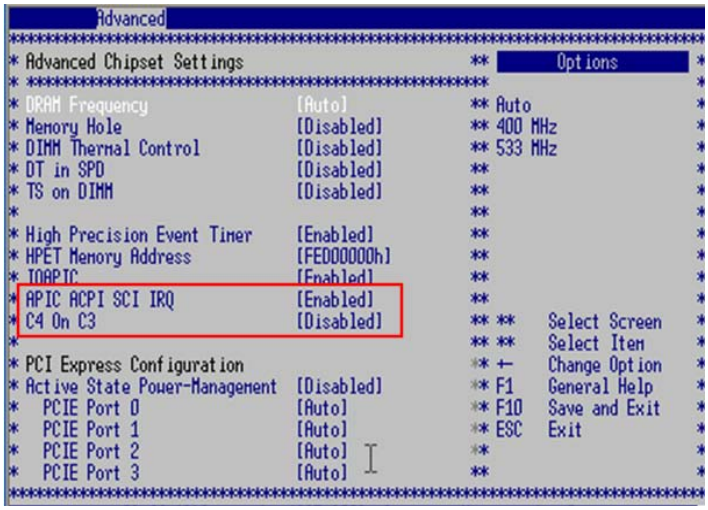
5.4.3.1 ACPI Configuration



5.4.3.2 CPU Settings



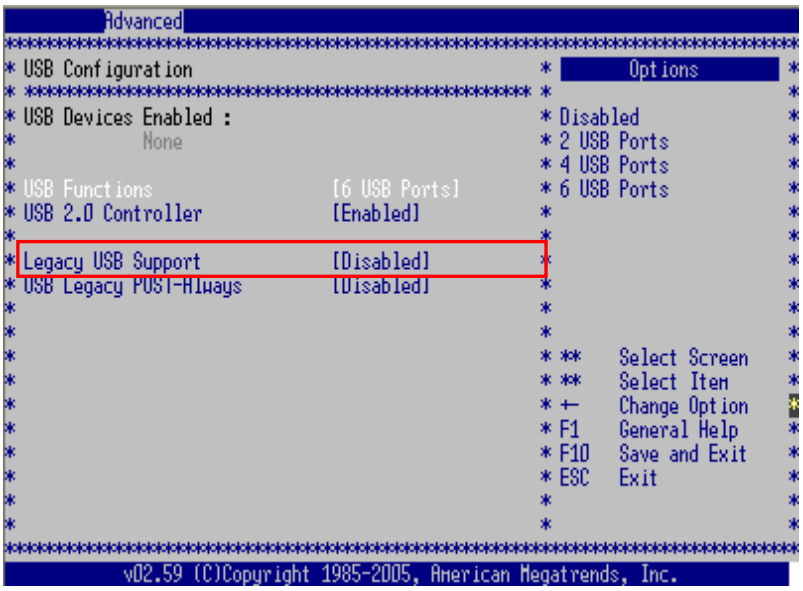
5.4.3.3 Chipset Settings



In any case enable APIC ACPI SCI IRQ !

5.4.3.4 USB Configuration

Disable USB Legacy support.



5.4.4 More common hints on hardware configuration

C-states play an important role in energy saving. But the higher the number behind the C-state, the more time it may take to wake up the processor. If it is not possible to disable these states in the BIOS, there are tools available in the internet, to maintain these things at runtime of Windows.

These tools are for example (just as hints, we did not test them nor do we recommend the use):

- RightMark CPU
- CPU-Z (CPUID)
- RM Clock
- And others …

Speedstep of INTEL or PowerNow of AMD have to be disabled, if possible, in any case.

In case none of the above actions leads to a good time behavior of the RTE, please contact the support of 3S Smart Software Solutions. It also can be necessary to send the hardware for time analysis to us.

5.5 Appendix E: Startup of a PROFIBUS Master by Hilscher CIFX 50-DP(M/S)

In the following the installation of a PROFIBUS-DP network controller with CODESYS Control RTE is described step by step. This card is equipped with the network controller NetX and the startup of any card with NetX technology is analogously.

5.5.1 Installation of a Windows driver

If you want to use the card with the runtime system CODESYS Control RTE, do not install the Hilscher drivers for the NetX cards, which are delivered by Hilscher. If you want to use the CODESYS Control RTE V3 with NetX, you have to install **WdmGeneric.sys** driver shipped with the card or **SysDrv3s.sys**. In case of standard installation the files are located in “C:\Programme\3S CODESYS\CODESYS Control RTE3\SysDrv3S” resp. “C:\Program Files\3S CODESYS\CODESYS Control RTE3\SysDrv3S”.

The RTE does not need a specific driver. You can even install no driver for the card. The most important thing is that the installed driver is passive. If you install an active driver, you will discover a lot of problems.

5.5.2 NetX firmware and NetX boot loader

Copy NetX firmware and NetX boot loader get delivered with the Setup and after installation are available in the subfolder "Hilscher CIFX\Firmware".

5.5.3 Configure the RTE in “CODESYSControl.cfg”

Edit „CODESYSControl.cfg“ located in the installation folder and add the component CmpHilscherCifX:

```
[ComponentManager]
Component.x=CmpHilscherCIFX
```

In section "CmpHilscherCIFX" depending on the number of CIFX cards activate the respective lines (remove ";" from the beginning of the line)

```
…
[CmpHilscherCIFX]
TraceLevel=0xFFFFFFFF
Device.0.BootloaderFilePath ... (see figure)
Device.0.Channel.0.FirmwareFile
…
```

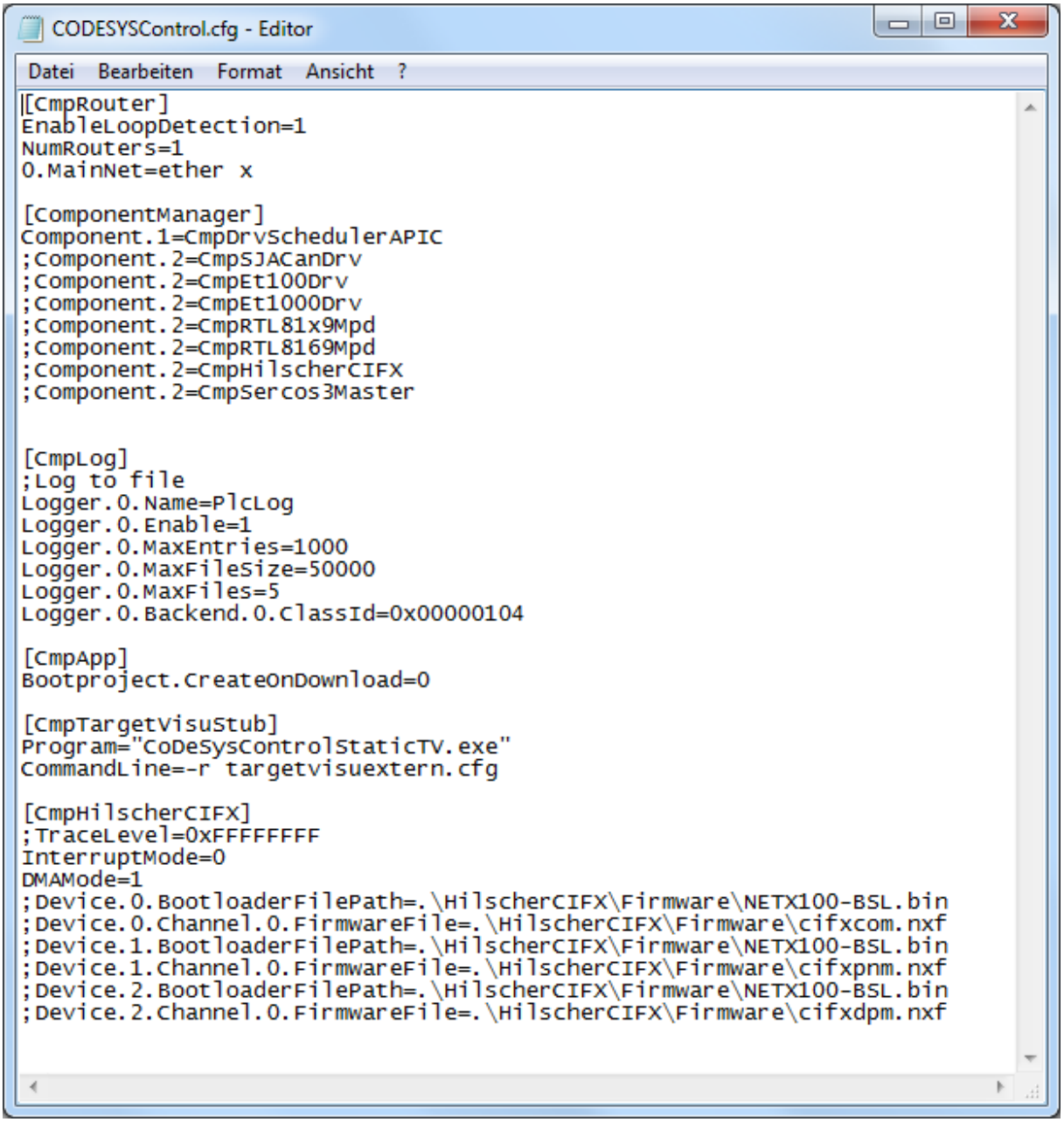
TraceLevel: This entry activates additional debug printouts in the CODESYS logger (dialog ‘Log’ in the device editor) or in the log files (“PlcLog.csv” or “PlcLog_x.csv”).

BootloaderFilePath: Specify the file path and filename of the NetX bootloader for each NetX controller. The relative path starts in the installation folder, an absolute paths can be used as well. If more than one NetX controllers are used, more than one file paths must be specified.

```
…
Device.0.BootloaderFilePath=NXCIF50-RTE.bin
Device.1.BootloaderFilePath=NXCIF50-RTE.bin
…
```

Device.X.Channel.X.FirmwareFile: Specify the file path and filename of the netX firmware for each netX and each channel. The relative path starts in the installation folder, an absolute path can be used as well. Most of the time channel 0 is used only. If more than one netX chips are used, more than one file paths must be specified.

```
…
Device.0.Channel.0.FirmwareFile=cifxdpm.nxf
Device.1.Channel.0.FirmwareFile=cifxdpm.nxf
…
```



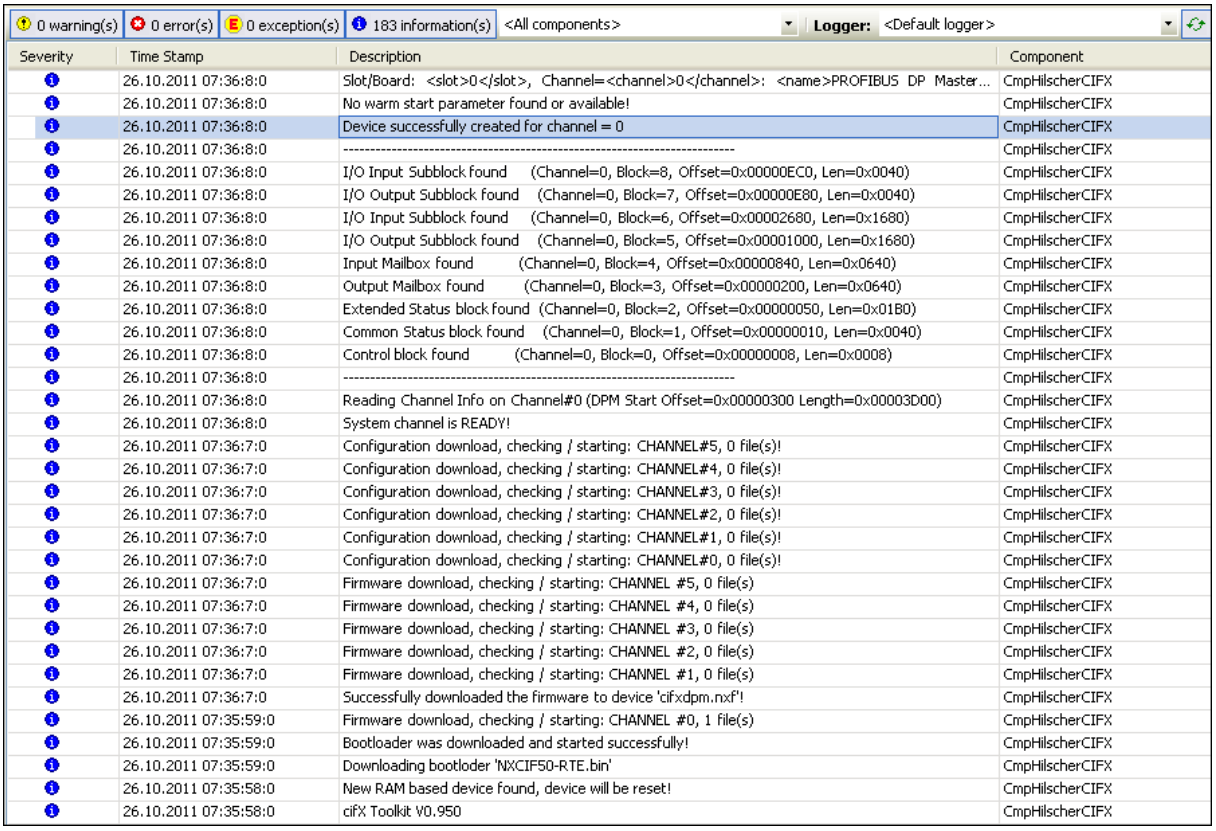
5.5.4 Start runtime

Start the RTE runtime system by use of ‚Start PLC' in the SysTray menu of the RTE. If the runtime is started, there must be logger entries in the runtime logger for CmpHilscherCIFX.

- The CmpHilscherCIFX component must be loaded
- The CmpHilscherCIFX component tells you the toolkit version (often asked by the Hilscher support)
- The firmware is loaded now. The logger tells you the name and the version of the firmware (also often asked by the Hilscher support).
- If the TraceLevel is set to 0xFFFFFFFF, the CmpHilscherCIFX prints out channel information for every configured NetX channel.

The logger entries are stored in LogPLCxx.csv located in the RTE directory or can be displayed in dialog ‚Log' of the device editor in the programming system.

© 3S-Smart Software Solutions GmbH



0 warning(s) | 0 error(s) | 0 exception(s) | 183 information(s) | <All components> | Logger: <Default logger>

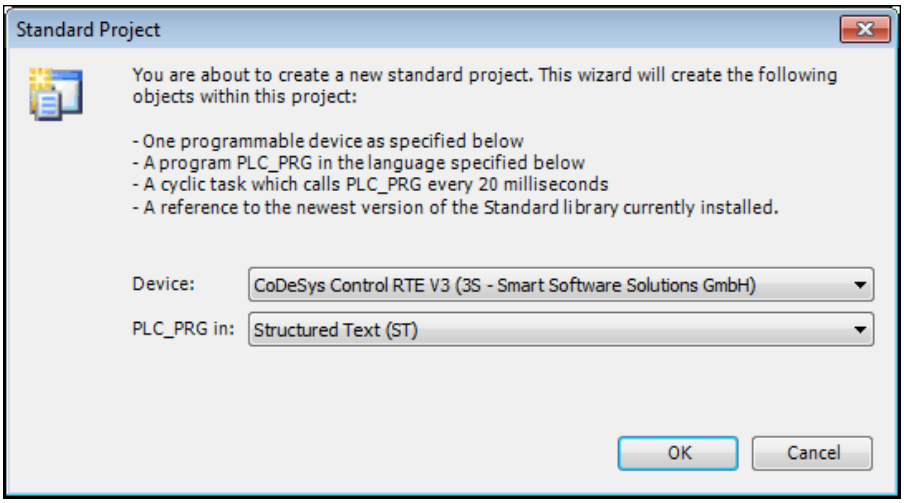
Severity	Time Stamp	Description	Component
	26.10.2011 07:36:8:0	Slot/Board: <slot>0</slot>, Channel=<channel>0</channel>: <name>PROFIBUS DP Master...	CmpHilscherCIFX
	26.10.2011 07:36:8:0	No warm start parameter found or available!	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Device successfully created for channel = 0	CmpHilscherCIFX
	26.10.2011 07:36:8:0	----------	CmpHilscherCIFX
	26.10.2011 07:36:8:0	I/O Input Subblock found (Channel=0, Block=8, Offset=0x00000EC0, Len=0x0040)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	I/O Output Subblock found (Channel=0, Block=7, Offset=0x00000E80, Len=0x0040)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	I/O Input Subblock found (Channel=0, Block=6, Offset=0x00002680, Len=0x1680)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	I/O Output Subblock found (Channel=0, Block=5, Offset=0x00001000, Len=0x1680)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Input Mailbox found (Channel=0, Block=4, Offset=0x00000840, Len=0x0640)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Output Mailbox found (Channel=0, Block=3, Offset=0x00000200, Len=0x0640)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Extended Status block found (Channel=0, Block=2, Offset=0x00000050, Len=0x01B0)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Common Status block found (Channel=0, Block=1, Offset=0x00000010, Len=0x0040)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Control block found (Channel=0, Block=0, Offset=0x00000008, Len=0x0008)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	----------	CmpHilscherCIFX
	26.10.2011 07:36:8:0	Reading Channel Info on Channel#0 (DPM Start Offset=0x00000300 Length=0x00003D00)	CmpHilscherCIFX
	26.10.2011 07:36:8:0	System channel is READY!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Configuration download, checking / starting: CHANNEL#5, 0 file(s)!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Configuration download, checking / starting: CHANNEL#4, 0 file(s)!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Configuration download, checking / starting: CHANNEL#3, 0 file(s)!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Configuration download, checking / starting: CHANNEL#2, 0 file(s)!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Configuration download, checking / starting: CHANNEL#1, 0 file(s)!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Configuration download, checking / starting: CHANNEL#0, 0 file(s)!	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Firmware download, checking / starting: CHANNEL #5, 0 file(s)	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Firmware download, checking / starting: CHANNEL #4, 0 file(s)	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Firmware download, checking / starting: CHANNEL #3, 0 file(s)	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Firmware download, checking / starting: CHANNEL #2, 0 file(s)	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Firmware download, checking / starting: CHANNEL #1, 0 file(s)	CmpHilscherCIFX
	26.10.2011 07:36:7:0	Successfully downloaded the firmware to device 'cifxdpm.nxf'!	CmpHilscherCIFX
	26.10.2011 07:35:59:0	Firmware download, checking / starting: CHANNEL #0, 1 file(s)	CmpHilscherCIFX
	26.10.2011 07:35:59:0	Bootloader was downloaded and started successfully!	CmpHilscherCIFX
	26.10.2011 07:35:59:0	Downloading bootloder 'NXCIF50-RTE.bin'	CmpHilscherCIFX
	26.10.2011 07:35:58:0	New RAM based device found, device will be reset!	CmpHilscherCIFX
	26.10.2011 07:35:58:0	cifX Toolkit V0.950	CmpHilscherCIFX

Logger entires in CODESYS, dialog ‘log’ of the device editor.

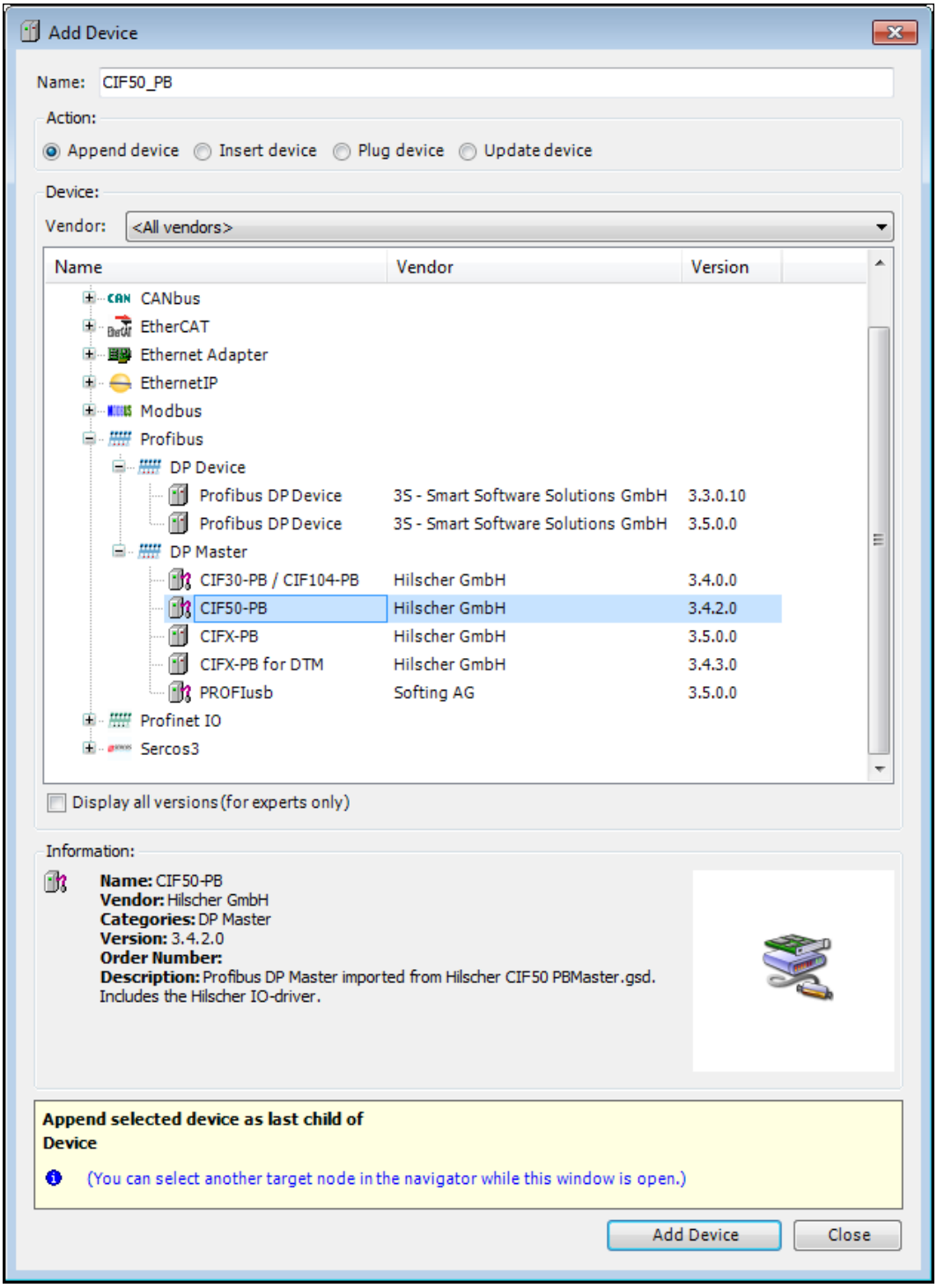
5.5.5 Create CODESYS project with IO configuration

Now you are able to create a simple application with a running field bus:

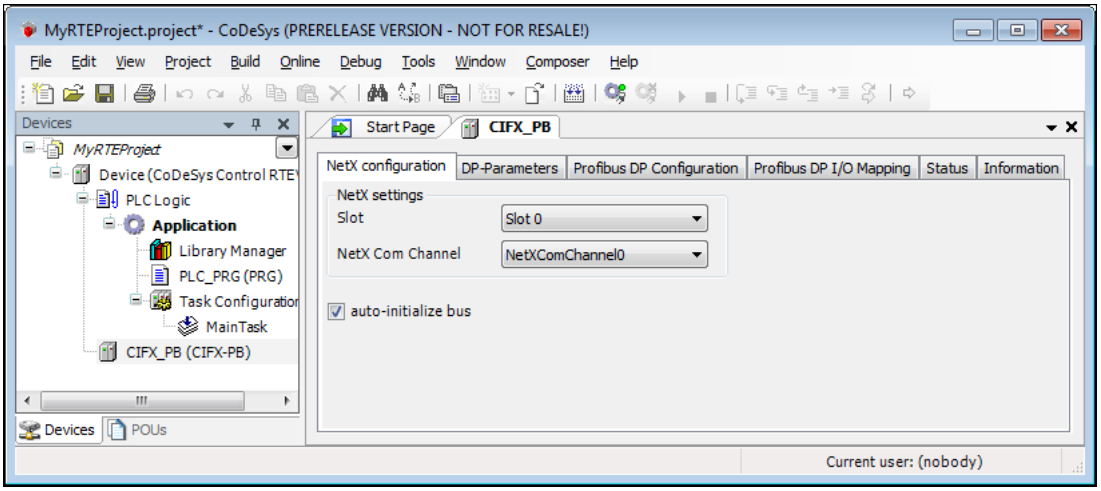
1. Create a new standard project
2. Add device CODESYS Control RTE V3.



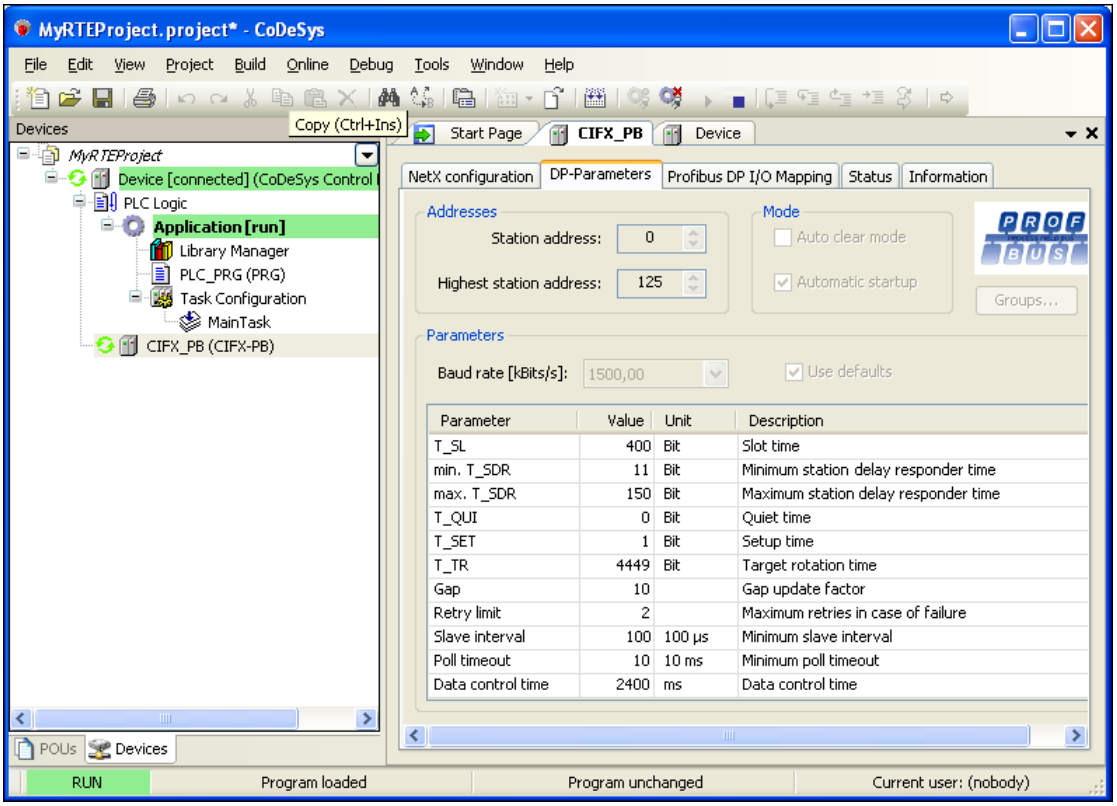
3. Add fieldbus master CIFX_PB.



4. Open the fieldbus master configuration dialog by double-click on the node in the device tree and switch to the tab ‚NetX configuration'. There, set ‚Slot' and ‚NetX Com Channel' in a way they are identical to the device number in the configuration file „CODESYSControl.cfg". Compare the value with the entries in the logger of the runtime system.



5. If you build the application, download it and go online, then any node in the device tree must be green.



5.5.6 FAQ

5.5.6.1 If the fieldbus master is not green, how can I find the problem?

Open the configuration dialog of the PLC (in this dialog you can find the communication settings, too), switch to the tab ‘Logger’ and click the green arrows. CODESYS uploads all log entries. Now you can look for errors and exceptions.

5.5.6.2 There are unresolved references

If there are unresolved references in the program, a runtime system component is not loaded or a wrong version of it is not loaded. Please check the CODESYS logger. The component CmpHilscherCIFX must be loaded.

If it is not loaded, please add it to the CODESYSControl.cfg file.

If it is loaded, it is possible to have got an old version of it. Please update the whole runtime system.

5.5.6.3 I’ve activated the Trace, but I do not see a new entry of the CmpHilscherCIFX component

If the TraceLevel is activated, but you do not see a new entry of the CmpHilscherCIFX, the runtime system does not find the NetX card/chip at all. Please check, if the card is plugged in correctly and if you have installed the correct driver. See chapter 5.5.1 “Installation of a Windows driver”.

Bibliography

Communication System NG Spec, 3S-Smart Software Solutions, 2004, V0.5

Runtime System 3.0, 3S-Smart Software Solutions GmbH

Change History

Version	Description	Editor	Date
0.1	HW Configuration Tips from ES added.	AF	28.7.2010
0.2	Review	RW	15.09.2010
0.3	Update according to review comments and review from AF thereof CDS-12397	AS	07.10.2010
1.0	Release after formal review	MN	21.10.2010
1.1	CDS-22297: NetX Example	AS	02.05.2011
1.2	Review	ESch	06.05.2011
1.3	CDS-22893 Ch. 3.2.5: “Max CPU Load by RTE“	AS	06.06.2011
1.3	Review	AF	06.06.2011
2.0	Release after formal review	MN	13.06.2011
2.1	Extension of chap. 5.5 Appendix E: Startup of a PROFIBUS Master by Hilscher CIFX 50-DP(M/S)	AS	26.10.2011
2.2	Review	ESch	25.11.2011
3.0	Release after formal review	AS	25.11.2011
3.1	CDS-29303	MN	17.09.2012
3.2	Security note added in Chap.1	TZ	30.10.2012
4.0	Release after formal review	MN	29.11.2012
4.1	Add chapter "Installation" (Chap.2)	AF	14.03.2013
5.0	Release after formal review	MN	15.03.2013

© 3S-Smart Software Solutions GmbH