CAN-API

Part 2:
Installation Guide

Software Manual

esd electronic system design gmbh
Vahrenwalder Str. 207 • 30165 Hannover • Germany
www.esd-electronics.com • Fax: 0511/37 29 8-68
Phone: 0511/37 29 80 • International: +49-5 11-37 29 80

esd electronic system design gmbh
Vahrenwalder Str. 207
30165 Hannover
Germany

Phone:	+49-511-372 98-0
Fax:	+49-511-372 98-68
E-mail:	info@esd-electronics.com
Internet:	www.esd-electronics.com

USA / Canada:
esd electronics Inc.
525 Bernardston Road
Suite 1
Greenfield, MA 01301
USA

Phone:	+1-800-732-8006
Fax:	+1-800-732-8093
E-mail:	us-sales@esd-electronics.com
Internet:	www.esd-electronics.us

Manual File:	I:\texte\Doku\MANUALS\PROGRAM\CAN\Schicht2\ENGLISCH\Univers\CAN-API-Installation_30e.en9
Manual Order no.:	C.2001.21 This order no. covers the two parts of the API manual: Part 1: 'CAN API, Function Description' and Part 2: 'CAN API, Installation Guide' (this manual).
Date of Print:	2007-07-27

Changes in the Software and/or Documentation

Chapter	Alterations in this manual versus previous version	Alterations in software	Alterations in documentation
1	Tables 1. - 4. revised	x	x
2.	Chapter completely revised	x	x
4.1	Chapter revised	x	x
5.1	Chapter '5.1 Installation Under VxWorks' completely revised and extended for VxWorks 6.x support.	x	x
-	The following modules are inserted: CAN-PCI104/200, CAN-PCIe/200, CAN-USB/2	x	x

Technical details are subject to change without notice.

This page is intentionally left blank.

Contents

Page

1. Guide to Your Installation

This document is valid for several esd-CAN boards. The boards are partly being sold installed into closed cases (e.g. CAN-USB-Mini), but sometimes also just as boards. In generally applicable contexts the term *modules* will therefore be used.

The installation descriptions given in this appendix are dependent on the module and operating system used.

For the hardware installation please consult the module-specific manual 'Hardware Installation and Technical Data'.

The CAN software is shipped on CD-ROM. The CD contains the CAN driver, the DLL, the configuration programs and the SDK (Software Development Kit) with the files which are required for developing and testing your own applications. The CD also contains CAN tools like the monitor program CANreal, CANrepro, CANscript and CANplot.

For installation, please follow the steps below (except CAN-Bluetooth and EtherCAN module):

1. **Installation of driver**
   Look for the column with 'your' operating system in the applicable table on the following pages. In the lines of the table you will find your CAN module. These two points lead you to the field which lists the chapters you need for your installation.

   The first chapters describe the installation of the driver.

   If you have a Windows operating system, you also have to carry out the second step and install the SDK. For all other operating systems this is not required.

2. **Installing the SDK (Software Development Kit - for Windows operating systems only)**
   After the driver has been installed, proceed to the installation of the SDK. This is described in the chapter starting on page 53. The SDK installation is only required for Windows operating systems.

Attention: The installation of the CAN Bluetooth driver is an exception of these rules! The CAN Bluetooth driver must always be installed last, that means after the installation of possible other CAN drivers and after the installation of the CAN-SDK!

The installation of the EtherCAN driver can be executed at any time.

Guide to Your Installation

The following tables shall be a guide to find the chapters that are important for your board and operating system:

CAN Module		Corresponding Chapter for ...						
		Non Real Time Windows Operating Systems					**Real Time Windows OS**	
		Windows NT	Windows 2000	Windows Server 2003, Windows XP (32-, 64-bit)	Windows Vista (32-, 64-bit)	Windows 95/98/ME	Windows CE.NET	Windows RTX
CAN-ISA/200	below called **'ISA-Bus Boards'**	2.5, 2.11	2.1.2 2.11	-	-	2.6.2, 2.11	-	-
CAN-PC104/200 (SJA1000 version)								
CAN-ISA/331		2.5, 2.11, 3.	2.1.2, 2.11, 3.	-	-	2.6.2, 2.11, 3.	-	-
CAN-PC104/331								
CAN-PCI104/200	below called **'PCI-Bus Boards'**	-	2.1.1, 2.4, 2.11	2.2, 2.4, 2.11	2.3, 2.4, 2.11	-	-	-
CAN-PCIe/200								
CAN-PCI/200		2.5, 2.11				2.6.1, 2.11	-	2.7, 2.11
CPCI-CAN/200								
CAN-PCI/266						-		
PMC-CAN/266								
CAN-PCI/331		2.5, 2.11, 3.	2.1.1, 2.4, 2.11, 3.	2.2, 2.4, 2.11, 3.	2.3, 2.4, 3.	2.6.1, 2.11, 3.	2.8, 2.11, 3.	2.7, 2.11, 3.
CPCI-CAN/331								
PMC-CAN/331							-	
CAN-PCI/360								
CPCI-CAN/360								
CAN-PCI/405		-	only 32-bit drivers: 2.1.1, 2.4, 2.11	only 32-bit drivers: 2.1.1, 2.4, 2.11	only 32-bit drivers: 2.1.1, 2.4, 2.11	-	-	
CAN-USB-Mini		-	2.1.1, 2.4, 2.11, 3.	2.2, 2.4, 2.11, 3.	2.3, 2.4, 2.11, 3.	2.6.1, 2.11, 3.	2.8, 2.11, 3.	-
CAN-USB/2		-	2.1.1, 2.4, 2.11, 3.	2.2, 2.4, 2.11, 3.	2.3, 2.4, 2.11, 3.	-	-	-
CAN-Bluetooth		-	2.9, 2.11	2.9, 2.11	-	-	-	-
EtherCAN		2.10, 2.11	2.10, 2.11,	2.10, 2.11	2.10, 2.11	-	-	-
CAN-PCC		2.5, 2.11	2.1.2, 2.11	-	-	2.6.2, 2.11	-	-

- ... No implementation available at the moment

Table 1: Guide to find the relevant chapters for the **Windows** software installation

CAN Module		Corresponding Chapter for **Unix Operating Systems**					
		Linux	LynxOS	Power-MAX OS	Solaris	SGI-IRIX6.5	AIX
CAN-ISA/200	below called **'ISA-Bus Boards'**	4.1.1	-	-	-	-	-
CAN-PC104/200 (SJA1000 version)							
CAN-PC104/200 (82527 version)							
CAN-ISA/331		4.1.1, 6.	4.2, 6.	-	4.4	-	-
CAN-PC104/331							
CAN-PCI104/200	below called **'PCI-Bus Boards'**	4.1.1	-	-	-	-	-
CAN-PCIe/200							
CAN-PCI/200		4.1.1	-	-	-	-	-
CPCI-CAN/200							
CAN-PCI-/266		4.1.1	-	-	-	-	-
PMC-CAN/266							
CAN-PCI/331		4.1.1, 6.	4.2, 6.	-	4.4	4.5	4.6
CPCI-CAN/331							
PMC-CAN/331							
CAN-PCI/360			-		-	-	-
CPCI-CAN/360							
CAN-PCI/405						4.5	
CAN-USB-Mini		4.1.1 6.	-	-	-	-	-
CAN-USB/2		4.1.1 6.	-	-	-	-	-
CAN-Bluetooth		-	-	-	-	-	-
EtherCAN		4.1.3	-	-	-	-	-
CAN-PCC		-	-	-	-	-	-
VME-CAN2		4.1.2	4.2	4.3	-	-	-
VME-CAN4		4.1.2	4.2	4.3	4.4	-	-

- ... No implementation available at the moment

Table 2: Guide to find the relevant chapters for the **Unix** operating systems

CAN Module		Corresponding Chapter for **Real Time Operating Systems**			
		VxWorks	QNX4	QNX6	RTOS-UH
CAN-ISA/200	below called **'ISA-Bus Boards'**	5.1	5.2.1	5.2.2	-
CAN-PC104/200 (SJA1000 version)	below called **'ISA-Bus Boards'**	5.1	5.2.1	5.2.2	-
CAN-PC104/200 (82527 version)	below called **'ISA-Bus Boards'**	5.1	-	-	-
CAN-ISA/331	below called **'ISA-Bus Boards'**	5.1, 6.	5.2.1, 6.	5.2.2, 6.	-
CAN-PC104/331	below called **'ISA-Bus Boards'**	5.1, 6.	5.2.1, 6.	5.2.2, 6.	-
CAN-PCI104/200	below called **'PCI-Bus Boards'**	5.1	-	5.2.2	-
CAN-PCIe/200	below called **'PCI-Bus Boards'**	5.1	-	5.2.2	-
CAN-PCI/200	below called **'PCI-Bus Boards'**	5.1	-	5.2.2	-
CPCI-CAN/200	below called **'PCI-Bus Boards'**	5.1	-	5.2.2	-
CAN-PCI-/266	below called **'PCI-Bus Boards'**	-	-	5.2.2	-
PMC-CAN/266	below called **'PCI-Bus Boards'**	-	-	5.2.2	-
CAN-PCI/331	below called **'PCI-Bus Boards'**	5.1, 6.	5.2.1, 6.	5.2.2, 6.	-
CPCI-CAN/331	below called **'PCI-Bus Boards'**	5.1, 6.	5.2.1, 6.	5.2.2, 6.	-
PMC-CAN/331	below called **'PCI-Bus Boards'**	5.1, 6.	5.2.1, 6.	5.2.2, 6.	-
CAN-PCI/360	below called **'PCI-Bus Boards'**	5.1, 6.	-	5.2.2, 6.	-
CPCI-CAN/360	below called **'PCI-Bus Boards'**	-	-	5.2.2	-
CAN-PCI/405	below called **'PCI-Bus Boards'**	-	-	5.2.2, 6.	-
CAN-USB-Mini		-	-	-	-
CAN-USB/2		-	-	-	-
CAN-Bluetooth		-	-	-	-
EtherCAN		-	-	-	-
CAN-PCC		-	-	-	-
VME-CAN2		-	-	-	*)
VME-CAN4		5.1	-	-	*)

- ... No implementation available at the moment
*) Note: For the operating system RTOS-UH drivers for host CPUs are available. Installation notes are not yet available.

Table 3: Guide to find the relevant chapters for the **Real Time** operating system

CAN Module	**Local Operating Systems** (driver running locally at the CAN module)			
	Linux	VxWorks (5.4, 5.5.x, 6.x)	QNX6	RTOS-UH
PMC-CPU/405	*)	5.1	5.2.2	*)
CPCI-405	4.1.1	5.1	5.2.2	*)
CPCI-CPU/750			-	-
EPPC-405	-	5.1	-	*)
EPPC-405-HR	-		-	*)
EPPC-405-UC	-		-	-

- ... No implementation available at the moment
*) Note: For these operating systems local drivers are available. Installation notes are not yet available.

Table 4: Guide to find the relevant chapters for the **host boards and embedded systems**

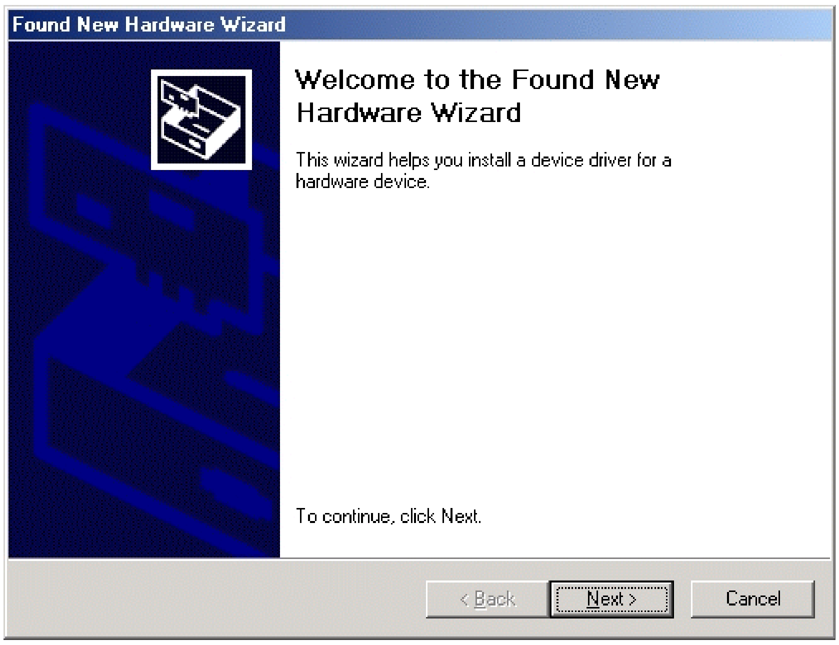
2. Windows Operating Systems

2.1 Windows 2000 Installation

In order to install the device driver for Windows 2000 the user must have administrator privileges. For these Windows versions non plug-and-play capable CAN modules (ISA, PC104 and Parallel Port) are supported by installing the Windows NT driver which is described in chapter 2.1.2.

2.1.1 Driver Installation of Plug-and-Play Capable CAN Modules

A non-USB module has to be plugged in before the Windows startup, an USB based CAN module may also be plugged in while Windows is running. Windows will indicate the detection of the new hardware by launching the *Found New Hardware Wizard*. Click *Next* to proceed with the installation.



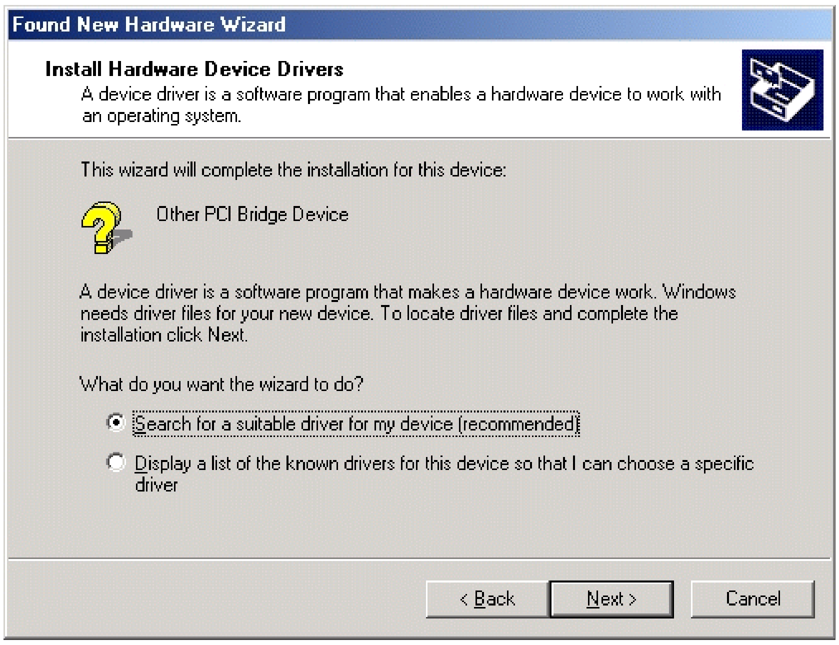
Found New Hardware Wizard

Welcome to the Found New Hardware Wizard

This wizard helps you install a device driver for a hardware device.

To continue, click Next.

Choose *Search for a suitable driver for my device (recommended)* and click the *Next* button. The reported device type in this and the following dialogues depends on the CAN module hardware.


Found New Hardware Wizard

Install Hardware Device Drivers
A device driver is a software program that enables a hardware device to work with an operating system.

This wizard will complete the installation for this device:

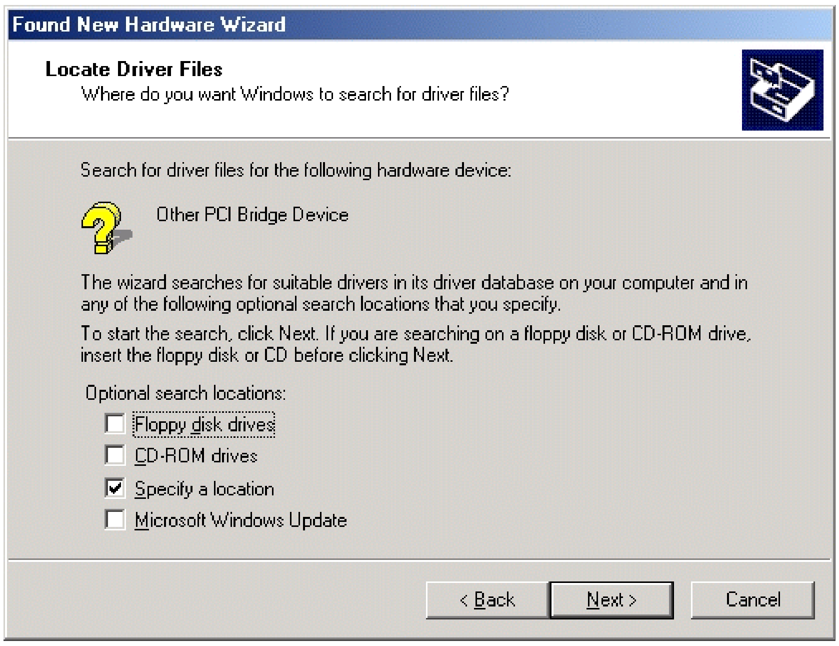
Other PCI Bridge Device

A device driver is a software program that makes a hardware device work. Windows needs driver files for your new device. To locate driver files and complete the installation click Next.

What do you want the wizard to do?

- [✓ Search for a suitable driver for my device (recommended)]
- Display a list of the known drivers for this device so that I can choose a specific driver

Check the box *CD-ROM drives* if you have the CAN driver CD inserted into your optical drive otherwise check *Specify a location*. Click *Next* to proceed with the driver installation.


Found New Hardware Wizard

Locate Driver Files
Where do you want Windows to search for driver files?

Search for driver files for the following hardware device:

Other PCI Bridge Device

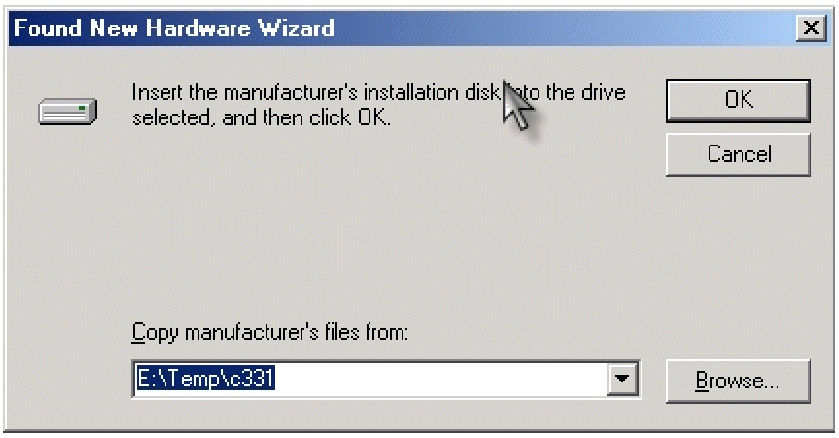
The wizard searches for suitable drivers in its driver database on your computer and in any of the following optional search locations that you specify.

To start the search, click Next. If you are searching on a floppy disk or CD-ROM drive, insert the floppy disk or CD before clicking Next.

Optional search locations:

- ☐ Floppy disk drives
- ☐ CD-ROM drives
- ☑ Specify a location
- ☐ Microsoft Windows Update

If you don’t install from the CAN driver CD you have to locate the driver files with the next step.

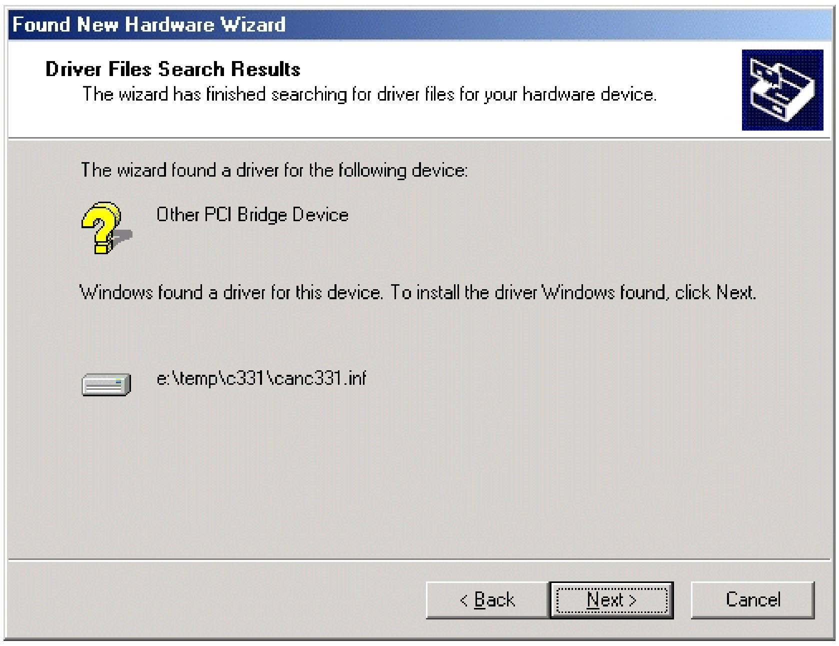


Insert the manufacturer's installation disk into the drive selected, and then click OK.

Copy manufacturer's files from:

E:\Temp\c331

The New Hardware Wizard automatically selects the correct `INF` file, which again depends on the CAN module hardware. Click *Next* to start the installation process.



Driver Files Search Results

The wizard has finished searching for driver files for your hardware device.

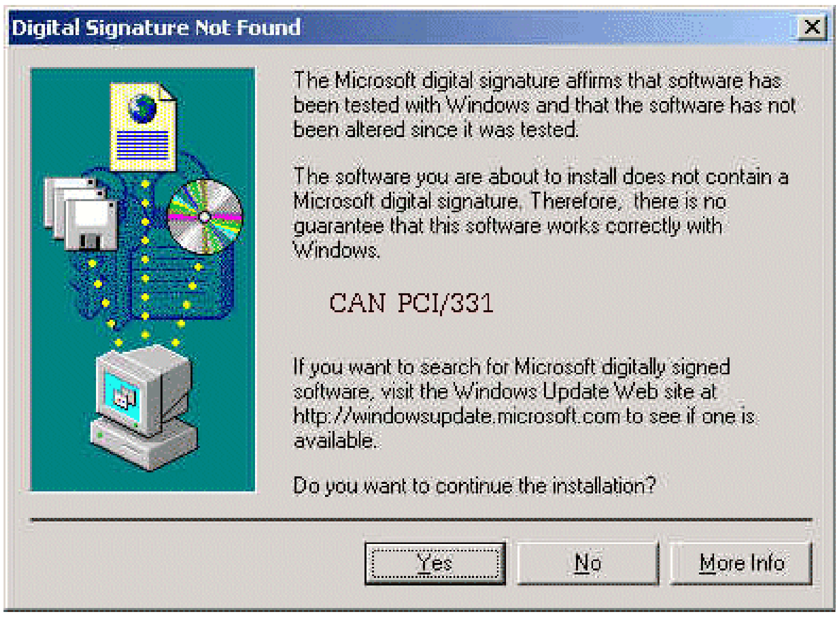
The wizard found a driver for the following device:

Other PCI Bridge Device

Windows found a driver for this device. To install the driver Windows found, click Next.

e:\temp\c331\canc331.inf

If Windows is configured to warn when unsigned (non-WHQL certified) drivers are about to be installed, a *Security Alert* dialogue will appear. The Security Alert dialogue has to be a answered with *Yes* to continue the driver installation. For more details about device driver and software signing refer to chapter 2.4.


Digital Signature Not Found

The Microsoft digital signature affirms that software has been tested with Windows and that the software has not been altered since it was tested.

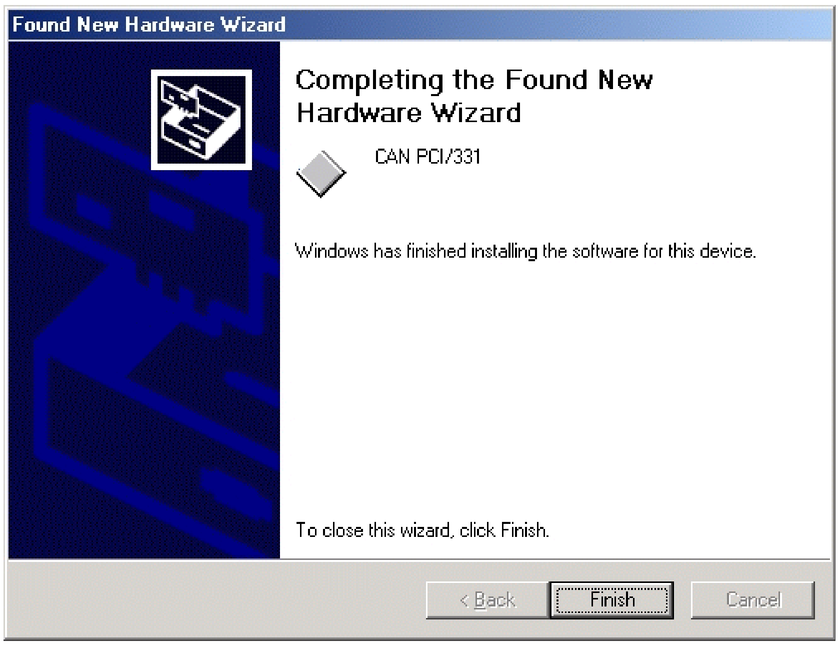
The software you are about to install does not contain a Microsoft digital signature. Therefore, there is no guarantee that this software works correctly with Windows.

CAN PCI/331

If you want to search for Microsoft digitally signed software, visit the Windows Update Web site at http://windowsupdate.microsoft.com to see if one is available.

Do you want to continue the installation?

Now all necessary driver files are copied to your system. Finally the Hardware Wizard should indicate the successful installation of the driver. Click *Finish* to complete the installation.


Found New Hardware Wizard

Completing the Found New Hardware Wizard

CAN PCI/331

Windows has finished installing the software for this device.

To close this wizard, click Finish.

After the driver installation has been completed, you may open the device manager to verify the correct installation and configure driver settings. Refer to chapter 2.4 for details.

2.1.2 Driver Installation of Non Plug-and-Play Capable CAN Modules

Note: For ISA-/PC104-bus based CAN modules and the CAN-PCC the Windows NT drivers are used. Please refer to chapter 2.6.2 for installation instruction.

2.1.2.1 CAN-PCC only: Stopping the System Driver for the Parallel Interface

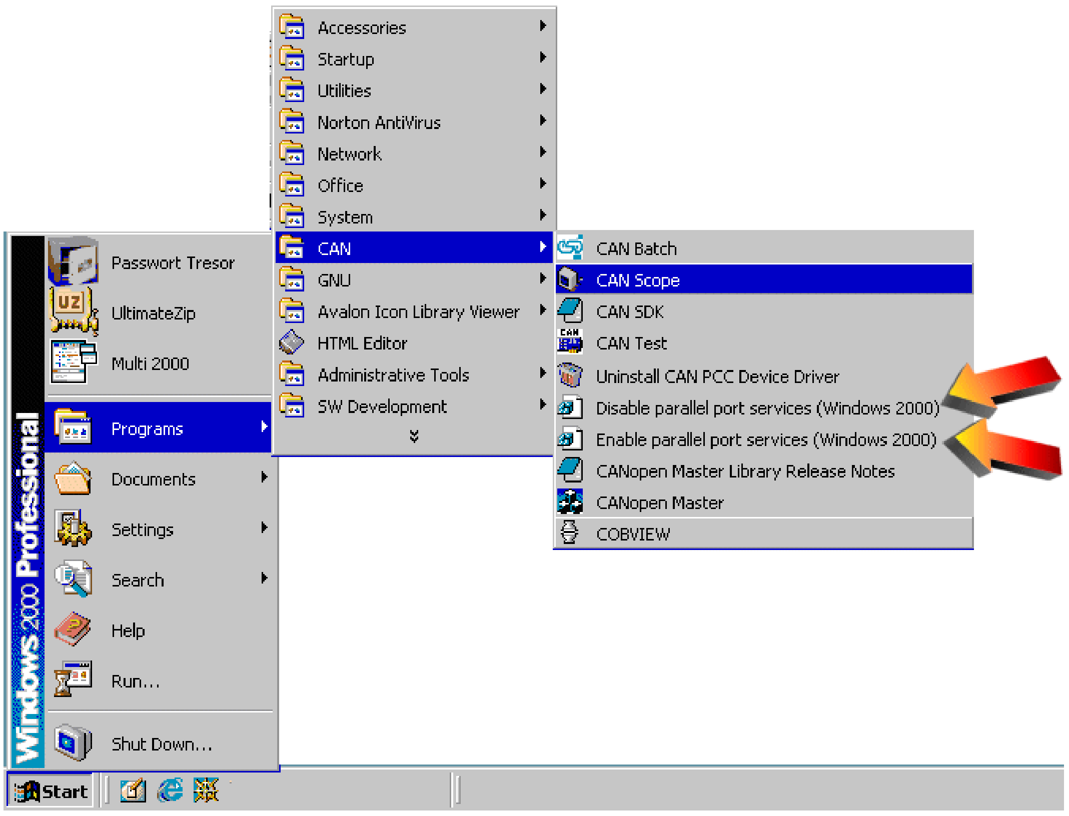
The CAN-PCC-module is run via the parallel interface of the computer. The Windows 2000 driver accesses this interface. To ensure that the CAN driver has got unrestricted access to the parallel port, the Windows 2000 driver has to be stopped **before the CAN driver is started**.

This can be accomplished by two calls which are available after the installation. They can be reached via *Programs/CAN*:

- **before** calling the CAN driver *Disable parallel port services (Windows 2000)* has to be called *and* the computer has to be rebooted

- if the parallel port is to be used for other matters (e.g. as print port), the CAN driver has to be stopped first *and Enable parallel port services (Windows 2000)* has to be called *and* then the computer has to be rebooted

Note: If starting the CAN driver has been changed in the Device Manager from 'Manual' to 'Automatic', this has to be re-changed, if the parallel port is to be used for other applications again and not for the CAN driver anymore.

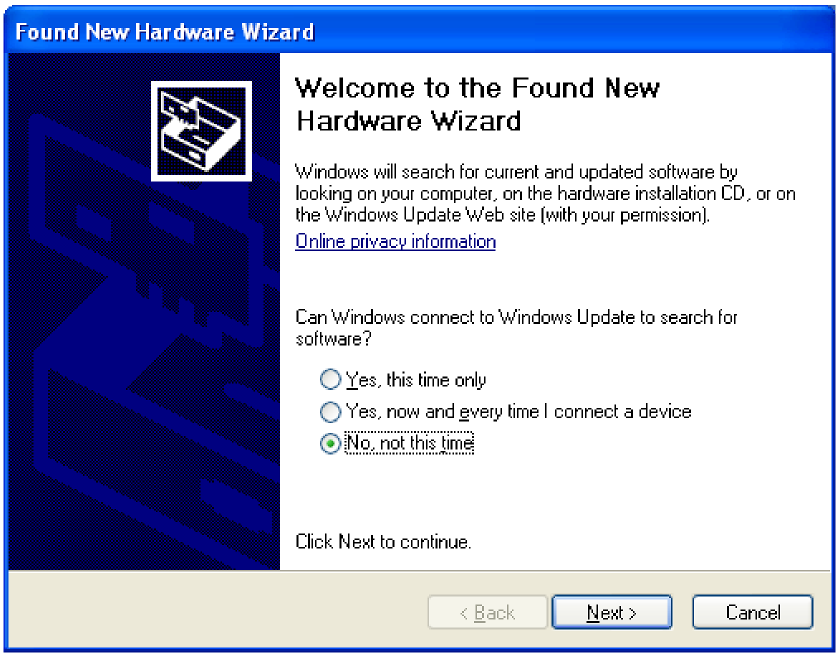


2.2 Windows XP/Server 2003 Installation (32- and 64-Bit Version)

In order to install the device driver for Windows XP and Windows Server 2003 (32-bit and 64-bit) the user must have Administrator privileges. For the 64-bit Windows versions only plug-and-play capable CAN modules (PCI, PCIe, CPCI, PMC, PCI104 and USB) are supported. For the 32-bit Windows versions the non Plug and Play capable modules are supported via the Windows NT driver. See chapter 2.1.2 for details.

2.2.1 Driver Installation for Plug-and-Play Capable CAN Modules.

A non-USB module has to be plugged in before the Windows startup, an USB-based CAN-module may also be plugged in while Windows is running. Windows will indicate the detection of the new hardware by launching the Found New Hardware Wizard. Select *No, not this time* from the available options and proceed by clicking the *Next* button.



Welcome to the Found New Hardware Wizard

Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).

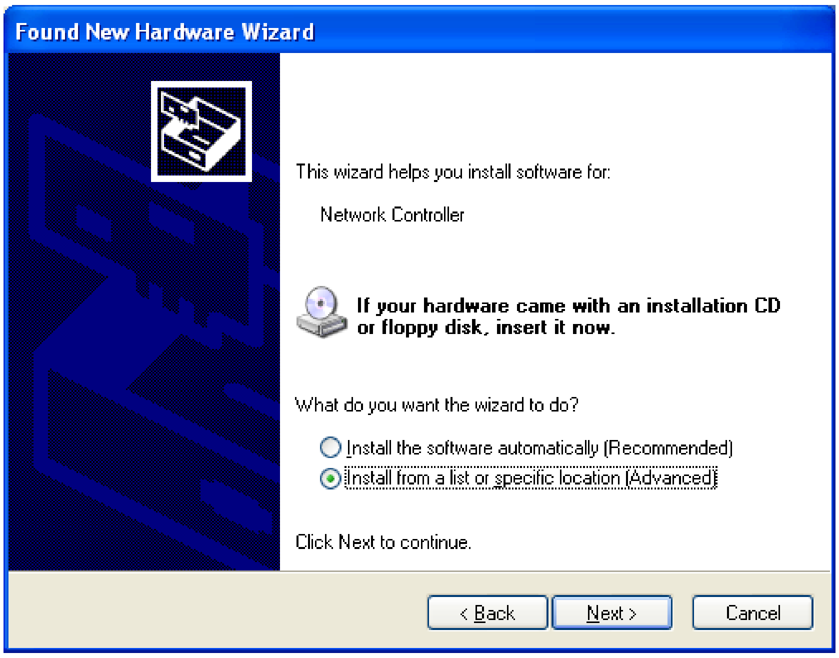
Online privacy information

Can Windows connect to Windows Update to search for software?

- ○ Yes, this time only
- ○ Yes, now and every time I connect a device
- [✓ No, not this time]

Click Next to continue.

In the following dialogue box choose *Install from a list or specific location (Advanced)* before you click the *Next* button. The reported device type in this and the following dialogues depends on the CAN module hardware.



Found New Hardware Wizard

This wizard helps you install software for:

Network Controller

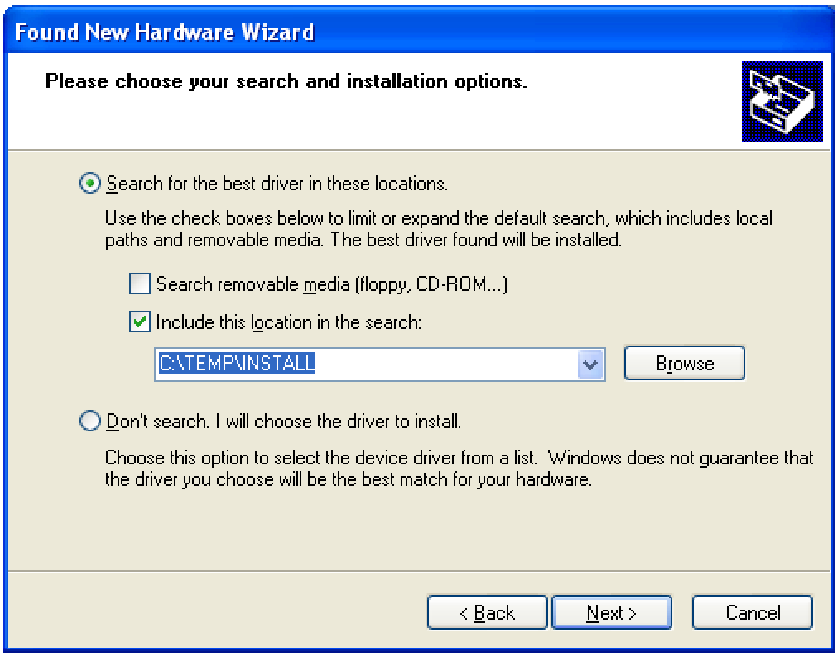
If your hardware came with an installation CD or floppy disk, insert it now.

What do you want the wizard to do?

( ) Install the software automatically (Recommended)
(●) Install from a list or specific location (Advanced)

Click Next to continue.

Select *Search for the best driver in these locations* and choose *Search removable media* if you install from the CAN driver CD or choose *Include this location in the search* and browse for the location of the driver files. By pressing the *Next* button the Wizard starts installing the device driver.



Found New Hardware Wizard

Please choose your search and installation options.

(●) Search for the best driver in these locations.

Use the check boxes below to limit or expand the default search, which includes local paths and removable media. The best driver found will be installed.

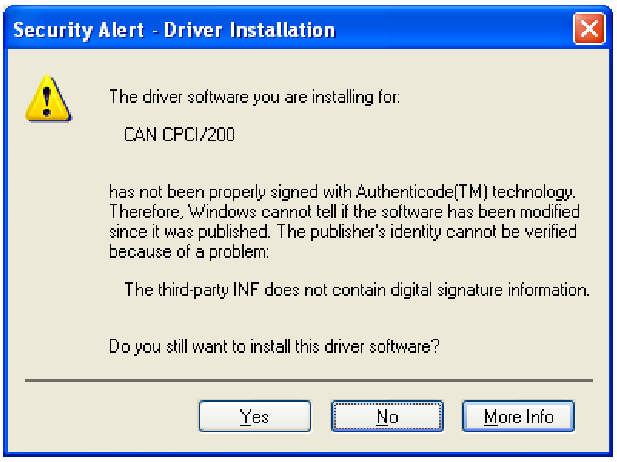
☐ Search removable media (floppy, CD-ROM...)
☑ Include this location in the search:

C:\TEMP\INSTALL

( ) Don't search. I will choose the driver to install.

Choose this option to select the device driver from a list. Windows does not guarantee that the driver you choose will be the best match for your hardware.

If Windows is configured to ignore file signature warnings the installation process is started and no further dialogue will appear. If Windows is configured to warn when unsigned (non-WHQL certified) drivers are about to be installed, a *Security Alert* dialogue will appear. The dialogue text depends on the Windows version (32- or 64-bit).
The complete CAN driver package for Windows 64-bit versions is digitally signed and you will see a dialogue box, which indicates that Windows has verified that the device driver is released by **esd** and hasn't been tampered. As both (32-bit and 64-bit) driver packages are not signed by the WHQL (Windows Hardware Quality Labs) the Security Alert dialogue has to be a answered with *Yes* to continue the driver installation. For more details about device driver and software signing refer to chapter 2.4.


Security Alert - Driver Installation

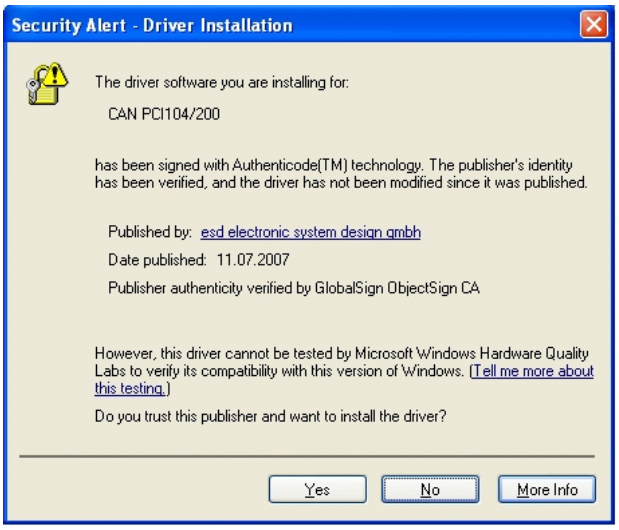
The driver software you are installing for:

CAN CPCI/200

has not been properly signed with Authenticode(TM) technology. Therefore, Windows cannot tell if the software has been modified since it was published. The publisher's identity cannot be verified because of a problem:

The third-party INF does not contain digital signature information.

Do you still want to install this driver software?


Security Alert - Driver Installation

The driver software you are installing for:

CAN PCI104/200

has been signed with Authenticode(TM) technology. The publisher's identity has been verified, and the driver has not been modified since it was published.

Published by: esd electronic system design gmbh

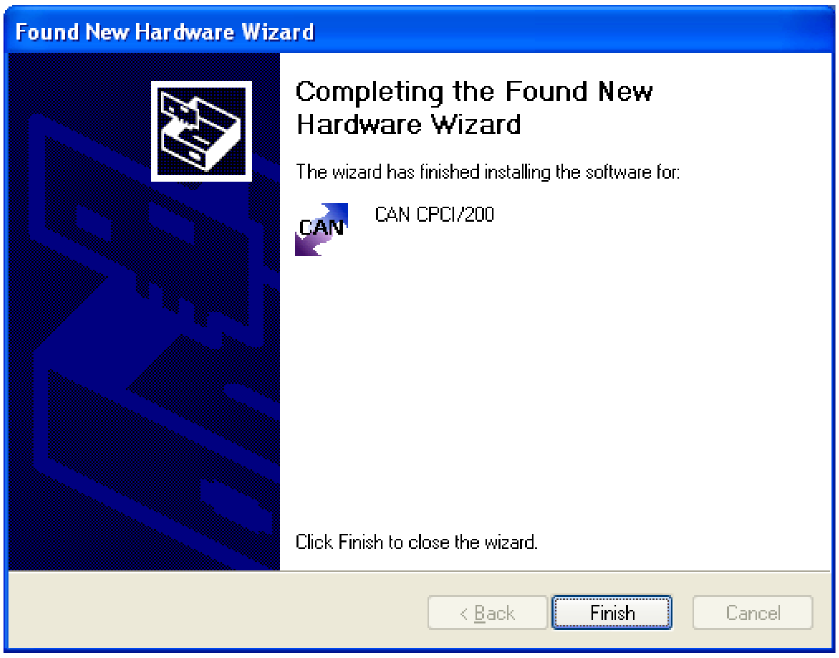
Date published: 11.07.2007

Publisher authenticity verified by GlobalSign ObjectSign CA

However, this driver cannot be tested by Microsoft Windows Hardware Quality Labs to verify its compatibility with this version of Windows. (Tell me more about this testing.)

Do you trust this publisher and want to install the driver?

After copying the necessary driver files to the system Windows displays a message indicating that the installation was successful. Click *Finish* to complete the installation.


Found New Hardware Wizard

Completing the Found New Hardware Wizard

The wizard has finished installing the software for:

CAN CPCI/200

Click Finish to close the wizard.

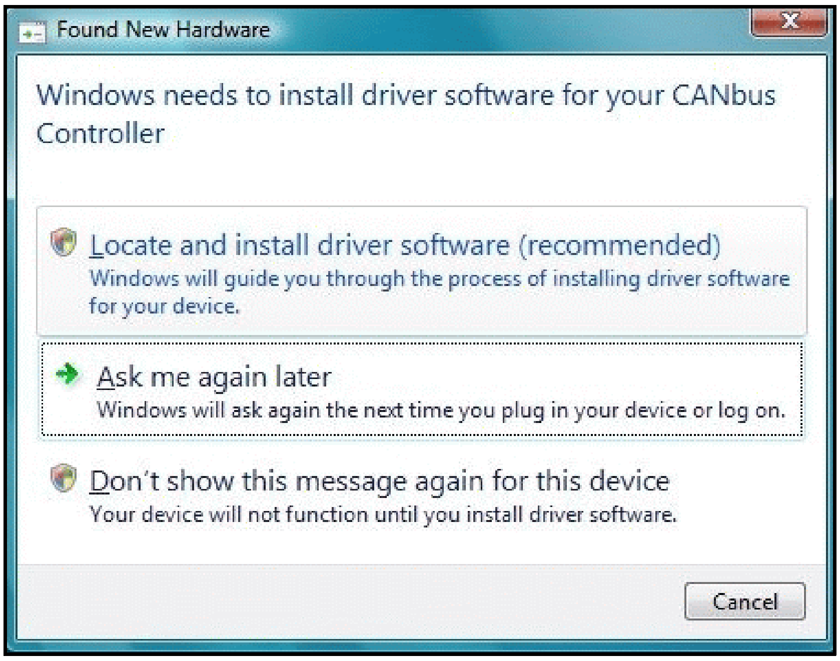
After the driver installation has been completed, you may open the device manager to verify the correct installation and configure some driver parameter. Refer to chapter 2.4 for details.

2.3 Windows Vista Installation (32- and 64-Bit Version)

In order to install the device driver for Windows Vista the user must have administrator privileges. For Windows Vista only plug-and-Play capable CAN modules (PCI, PCIe, CPCI, PMC, PCI104 and USB) are supported.

2.3.1 Driver Installation for Plug-and-Play Capable CAN Modules.

A non-USB module has to be plugged in before the Windows Vista startup, an USB-based CAN-module can also be plugged in while Windows Vista is running. Windows Vista will signal the detection of the new hardware by opening the dialogue box below. The text in the dialogue box may differ depending on the CAN module:



Found New Hardware

Windows needs to install driver software for your CANbus Controller

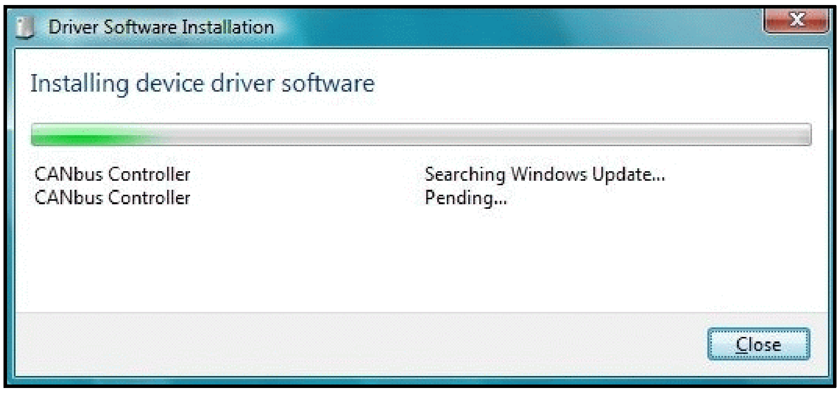
Locate and install driver software (recommended)
Windows will guide you through the process of installing driver software for your device.

Ask me again later
Windows will ask again the next time you plug in your device or log on.

Don't show this message again for this device
Your device will not function until you install driver software.

Choose *Locate and install driver software* to continue the device driver installation. As driver installation requires administrator privileges the UAC (User Account Control) opens the dialogue below, which has to be accepted.

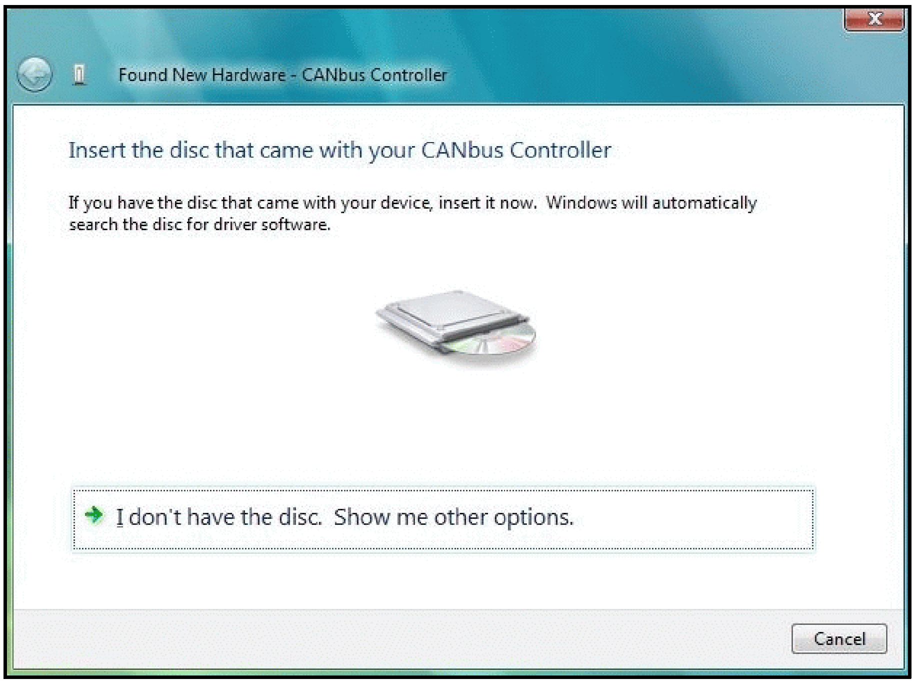
Depending on the configuration of the *Windows Update Driver Settings,* which can be changed via the *Hardware* tab of the *System Properties* dialogue which is opened if you select *Control Panel/System/Advanced System Settings*, Windows Vista will ask you if you want to search for a suitable driver online. If such a dialogue appears choose *Don't search online*, otherwise wait until the online search on Windows Update is completed which may take up to one minute.



Installing device driver software

CANbus Controller Searching Windows Update...
CANbus Controller Pending...

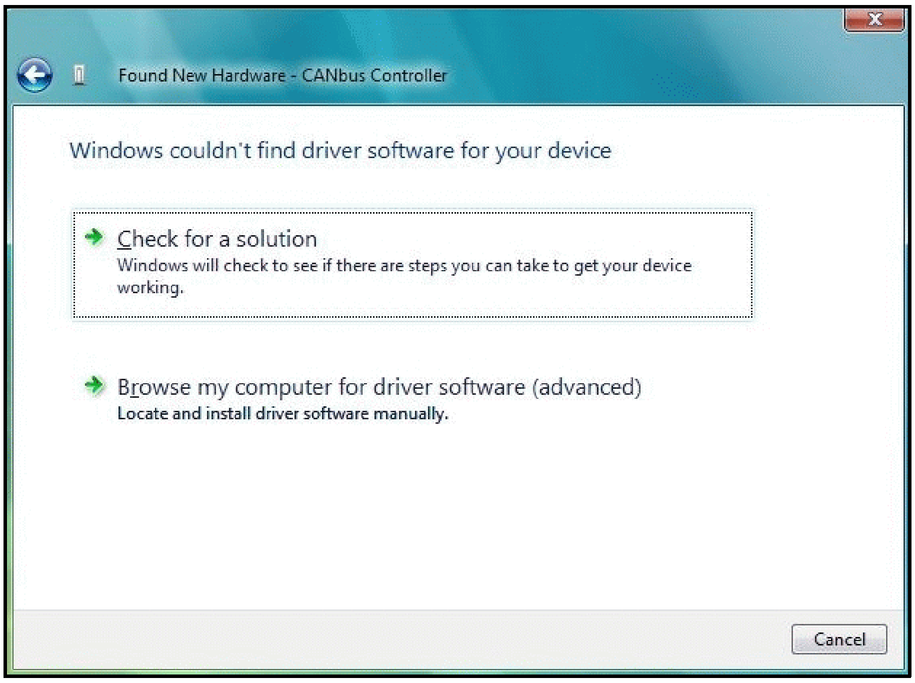
Insert your esd CAN driver disc and the installation process will continue. If the driver files are located on your hard disc choose *I don't have the disc. Show me other options*.


Found New Hardware - CANbus Controller
I don't have the disc. Show me other options.
Cancel

Insert the disc that came with your CANbus Controller

If you have the disc that came with your device, insert it now. Windows will automatically search the disc for driver software.

In the latter case choose *Browse my computer for driver software (advanced)* which is followed by a file dialogue to locate the driver files.



Found New Hardware - CANbus Controller

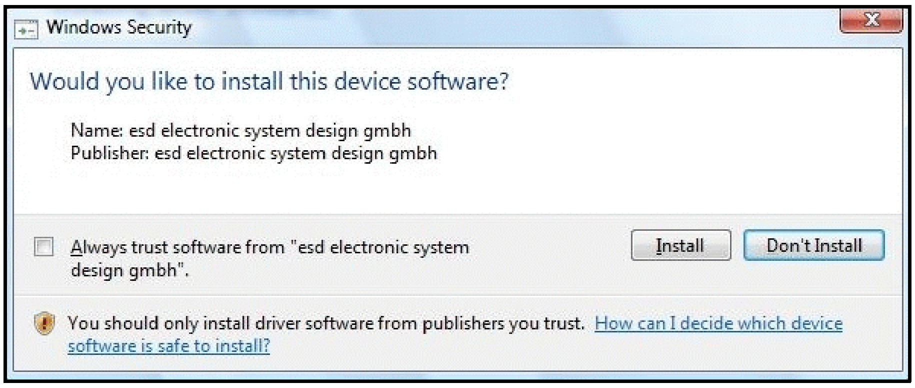
Windows couldn't find driver software for your device

Check for a solution
Windows will check to see if there are steps you can take to get your device working.

Browse my computer for driver software (advanced)
Locate and install driver software manually.

Cancel

The following *Windows Security* dialogue depends on the Windows Vista (32- or 64-bit) version. The complete CAN driver package for Windows Vista 64-bit is digitally signed and you will see the dialogue box below, which indicates that Windows Vista has verified that the device driver is released by **esd** and hasn't been tampered. You may select the *Always trust software from* ... check before selecting the *Install* button to prevent this dialogue installing other signed driver packages by **esd**.



Windows Security

Would you like to install this device software?

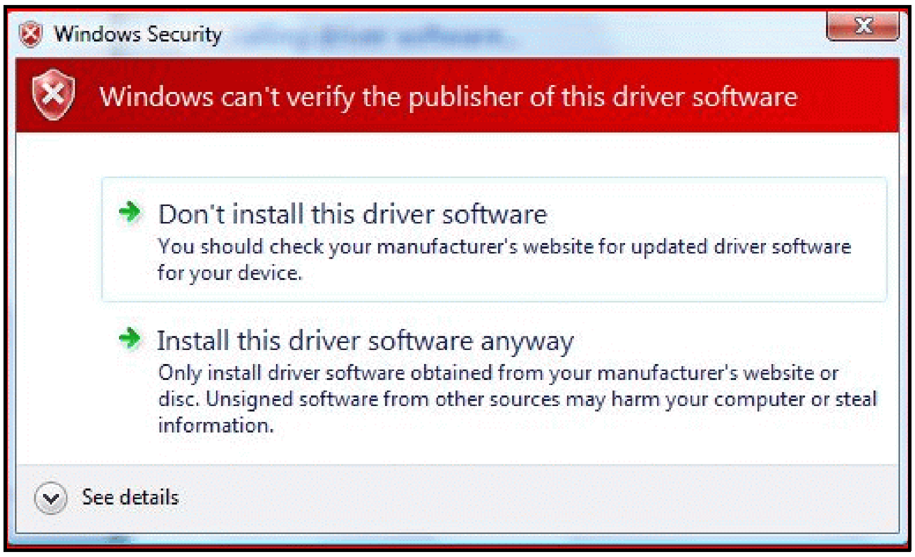
Name: esd electronic system design gmbh
Publisher: esd electronic system design gmbh

Always trust software from "esd electronic system design gmbh".

Install
Don't Install

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

The CAN driver package for Windows Vista 32-bit isn’t digitally signed and you will see the dialogue box below. Select *Install the driver software anyway* to continue the installation. For more details about device driver and software signing refer to chapter 2.4.


Windows Security

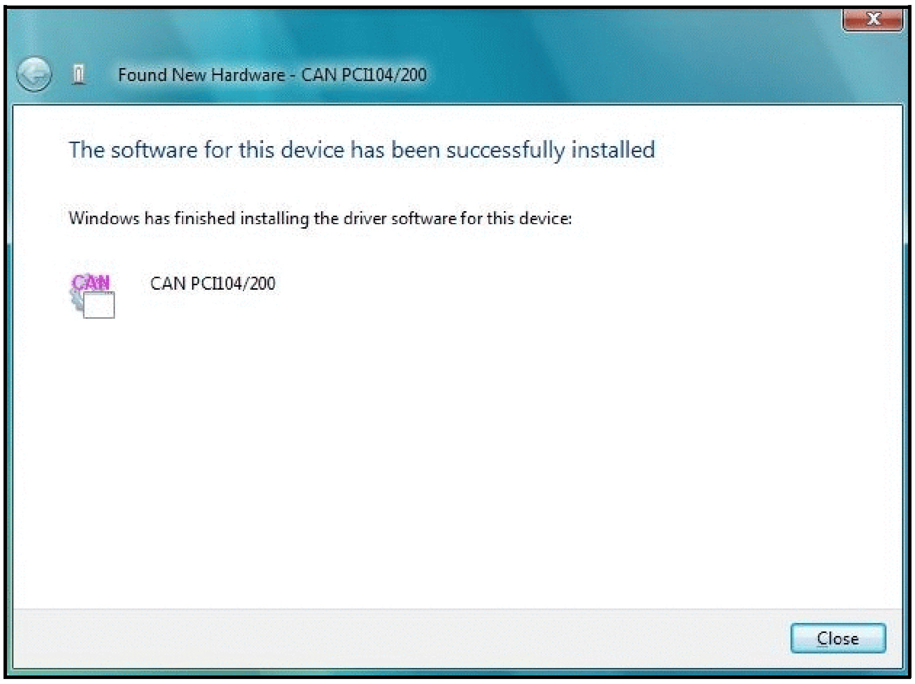
Windows can't verify the publisher of this driver software

Don't install this driver software
You should check your manufacturer's website for updated driver software for your device.

Install this driver software anyway
Only install driver software obtained from your manufacturer's website or disc. Unsigned software from other sources may harm your computer or steal information.

See details

After copying the necessary driver files to the system Windows displays a message indicating that the installation was successful. Click *Close* to complete the installation.


Found New Hardware - CAN PCI104/200

The software for this device has been successfully installed

Windows has finished installing the driver software for this device:

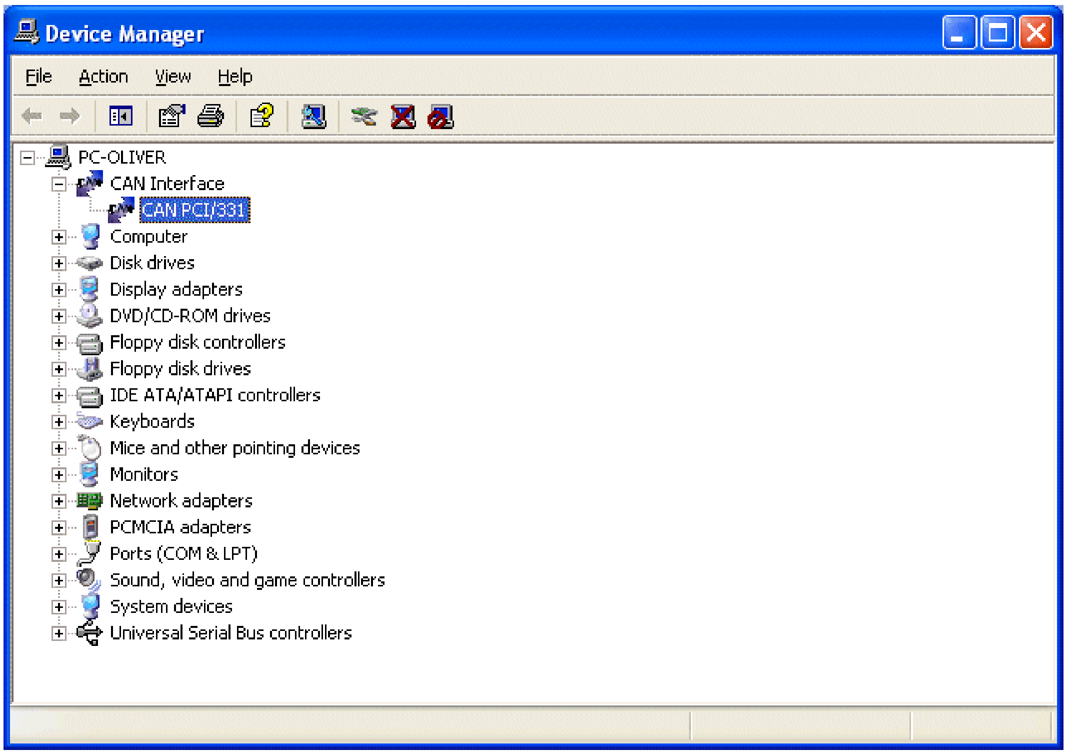
CAN PCI104/200

Close

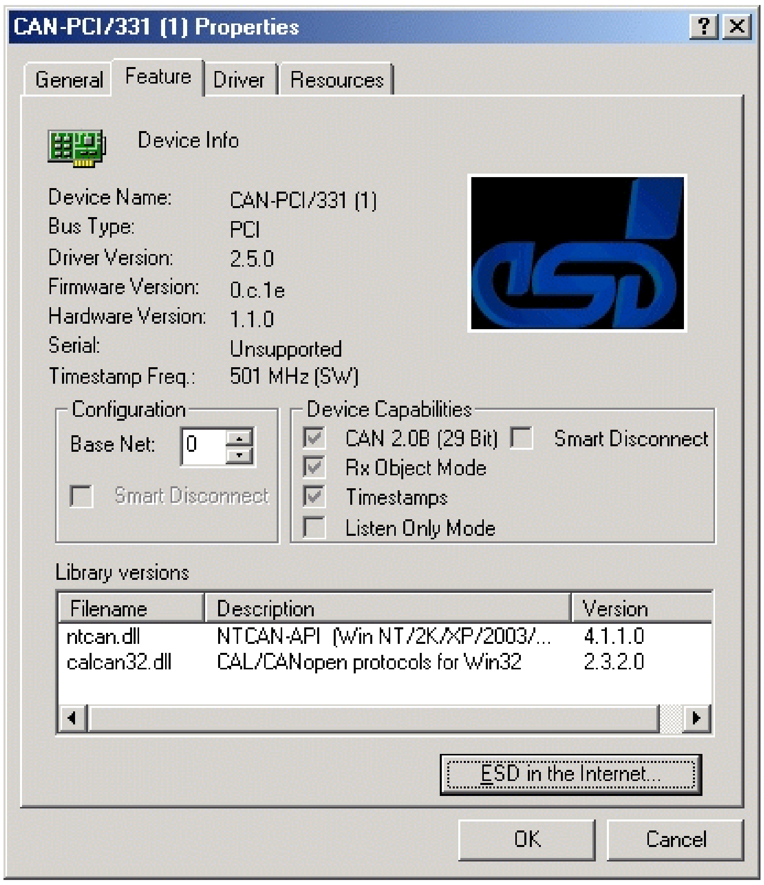
After the driver installation has been completed, you may open the device manager to verify the correct installation and configure some driver parameter. Refer to chapter 2.4 for details.

2.4 Driver Configuration

This chapter is valid for the operating systems Windows 2000, Windows XP/Server 2003 and Windows Vista, described in the preceding chapters.
To configure some settings of the device driver or check the installed version you have to open the Windows Device Manager. All device drivers for **esd** CAN modules are installed in the CAN Interface class as shown below.



To configure the driver you have to double-click the device instance to open the *Properties* dialogue of the device. Select the *Feature* tab.


CAN-PCI/331 (1) Properties

General | Feature | Driver | Resources

Device Info

Device Name:	CAN-PCI/331 (1)
Bus Type:	PCI
Driver Version:	2.5.0
Firmware Version:	0.c.1e
Hardware Version:	1.1.0
Serial:	Unsupported
Timestamp Freq.:	501 MHz (SW)

Configuration
Base Net: 0
☐ Smart Disconnect

Device Capabilities
☑ CAN 2.0B (29 Bit) ☐ Smart Disconnect
☑ Rx Object Mode
☑ Timestamps
☐ Listen Only Mode

Library versions

Filename	Description	Version
ntcan.dll	NTCAN-API (Win NT/2K/XP/2003/...	4.1.1.0
calcan32.dll	CAL/CANopen protocols for Win32	2.3.2.0

The dialogue box shows the following information:

- Driver, firmware and hardware revision.
- The serial number (if serial number access is supported by the hardware).
- The timestamp frequency (if timestamping is supported by the hardware and the driver) and the information if this is a hardware (HW) or software (SW) applied timestamp.
- The CAN module and driver related device capabilities.
- The version of the libraries which are installed together with the driver.

The dialogue box allows to configure the parameter *Base Net* and the *Smart Disconnect Feature*.

In *Base Net* a logical network number is assigned to the CAN module to distinguish between several physical ports. If a module has got more than one physical CAN port, the logical network number entered in *Base Net* is assigned to the first physical port and the following ports are numbered successively. The default value for the first instance of a hardware is always 0.

Attention: If there is more than one CAN module in the system, the user has to make sure that the logical net numbers which are assigned to the physical ports do not overlap !

The *Smart Disconnect Feature* can be enabled or disabled, if supported by the CAN hardware. The default after installation is disabled.

Attention: Parameter changes are invalid until the next start of the driver !

2.5 Windows NT 4.0 Installation

2.5.1 Installation of Software Drivers for the Host CPU

2.5.1.1 Starting Installation

When updating it is necessary to de-install the old driver version, as shown in section 2.5.1.5.

Attention:	The complete installation, configuration and starting the driver can only be done with administrator rights on the Windows NT computer!

You will find the device driver on the CD-ROM in a corresponding sub-directory.

For installation the CD-ROM containing the device driver has to be inserted into the drive and the installation program has to be executed. This can either be achieved by directly calling the program `setup.exe` or by pressing the *install* icon in dialogue box *add/remove program properties* in the folder *control panel*.

The following installation process is interactive-controlled. In the course of installation you can choose between one of three configurations:

Configuration	Description
Typical	Driver, DLL and program to update the local firmware.
Compact	Driver, DLL and configuration programs.
Custom	User-defined configuration.

Table 5: Configuration selection

After the installation and before the start of the driver, the computer has to be re-booted. This can be done directly from the setup program after the installation is finished.

Continue with the installation of the SDK, as described on page 53.

2.5.1.2 Calling Configuration Program CAN Control

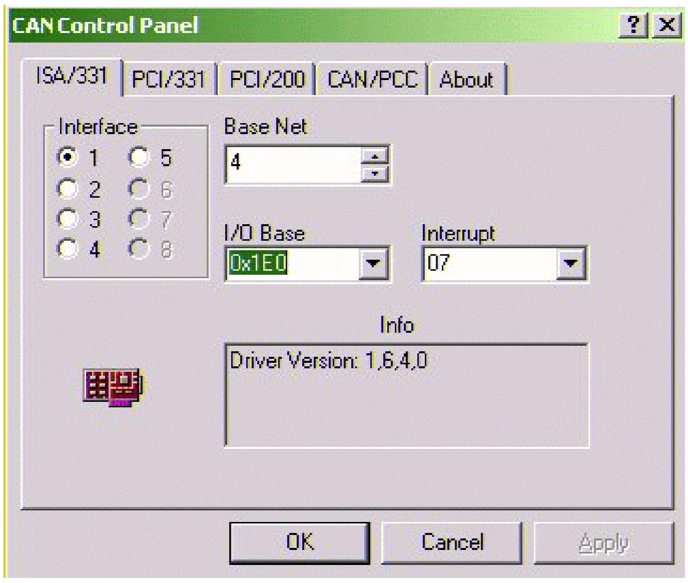
For the configuration of the driver the program CAN Control is part of the product package. It is called from the system control.

CAN

CAN Control

The following figures show a system with four esd-CAN modules (CAN-ISA/331, CAN-PCI/331, CAN-PCI/200 and CAN-PCC) exemplary. CAN Control only displays the esd-CAN modules which are installed in the system. If you have only installed one CAN module, only one file card is shown.

The following figure shows the setting options for CAN-**ISA** modules:



CAN Control Panel

ISA/331 | PCI/331 | PCI/200 | CAN/PCC | About

Interface: [✓ 1] 2 3 4 5 6 7 8

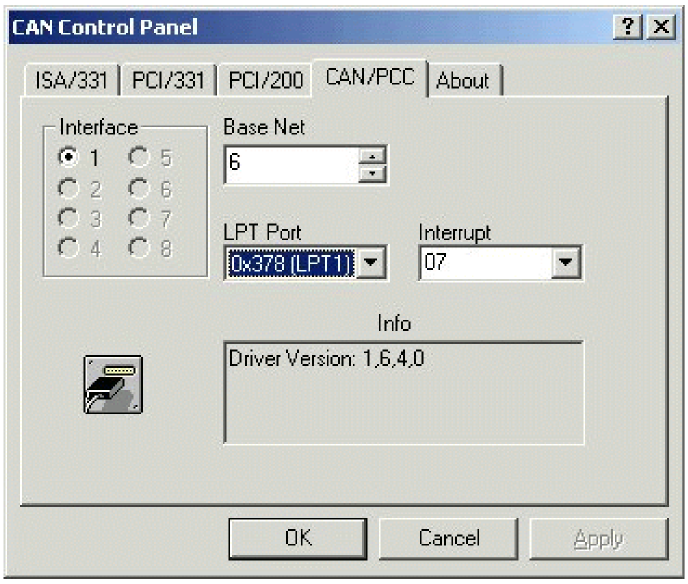
Base Net: 4

I/O Base: 0x1E0

Interrupt: 07

Info: Driver Version: 1.6.4.0

The following figure shows the setting options for the CAN-**PCC** module.



CAN Control Panel

ISA/331 | PCI/331 | PCI/200 | CAN/PCC | About

Interface: [✓ 1] 2 3 4 5 6 7 8

Base Net: 6

LPT Port: 0x378 (LPT1)

Interrupt: 07

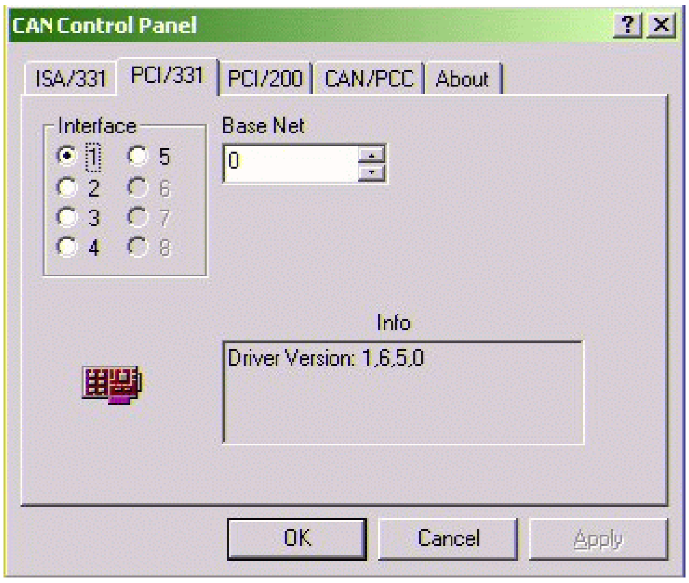
Info: Driver Version: 1.6.4.0

Under *Interface* you enter the number of the CAN module for which the settings are to be valid. In *Base Net* a base network number is assigned to the CAN module. If a module has got more than one physical CAN interface, the first gets the logical network number specified in *Base Net*, and the following are counted up from there.

Attention: The user has to make sure that no number is assigned twice when logical network numbers are assigned!

Enter your hardware configuration regarding the I/O port address for each CAN module, set the interrupt level and click OK. The driver is now installed and configured.

The following figure shows the setting options for CAN-**PCI** modules. Here, only logical CAN networks are assigned:



Base Net: 0

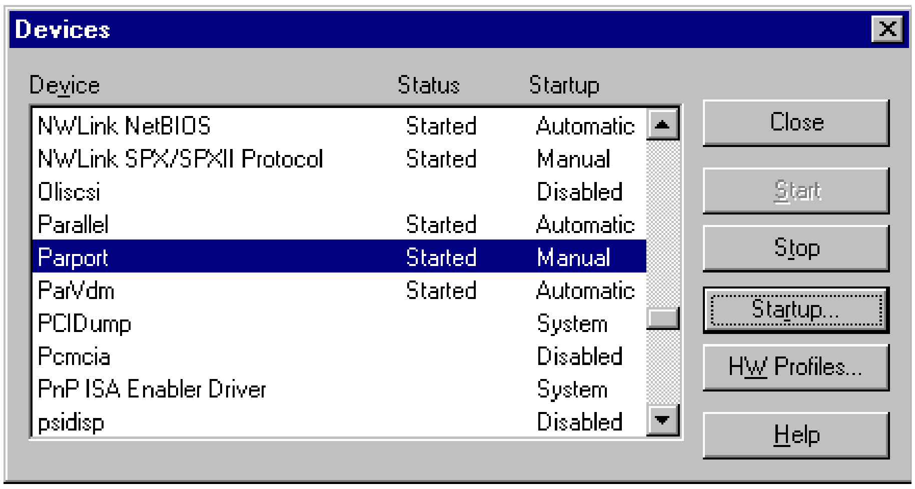
Driver Version: 1.6.5.0

2.5.1.3 CAN-PCC only: Stopping the Windows-NT Drivers for the Parallel Interface

The CAN-PCC module is operated by the parallel interface of the computer. It is standard for the Windows-NT driver to access this interface. For a free access of the CAN driver to the parallel port, the Windows-NT driver has to be stopped **before starting the CAN driver**. For this, you have to proceed as follows:

1. Select *System Control Panel* under *Settings*.

2. Here you have to double click the icon *Devices*. The following window will open:



Devices

Device	Status	Startup
NWLink NetBIOS	Started	Automatic
NWLink SPX/SPXII Protocol	Started	Manual
Oliscsi		Disabled
Parallel	Started	Automatic
Parport	Started	Manual
ParVdm	Started	Automatic
PCIDump		System
Pcmcia		Disabled
PnP ISA Enabler Driver		System
psidisp		Disabled

3. In this window you now have to cancel the device *Parport*. The system now tells you that the devices *Parallel* and *ParVdm* are also cancelled.

4. In order to prevent having to repeat this procedure with every system run-up, Startup has to be set to *Manual* for all three devices *Parallel, Parport* and *ParVdm*. With this setting the devices are not started automatically with every system run-up.

2.5.1.4 Starting the Driver

In order to start the driver, the command `net start` *`xxxx`* is to be entered in the command line first.

For the various CAN modules the drivers are called as follows:

CAN Module	Commands for Starting the Drivers
CAN-ISA/200	`net start c200i`
CAN-PC104/200 (SJA1000 version)	
CAN-ISA/331	`net start c331i`
CAN-PC104/331	
CAN-PCI/200	`net start c200`
CAN-PCI/266	
CPCI-CAN/200	
PMC-CAN/266	
CAN-PCI/331	`net start c331`
CPCI-CAN/331	
PMC-CAN/331	
CAN-PCI/360	`net start c360`
CPCI-CAN/360	
CAN-PCC	`net start cpcc`

Table 6: Commands for starting the driver

If the driver starts without complication, the start type of the driver can be set in dialogue *system control/devices* from **manual** to **automatic**, so that the driver is activated after each reboot and can also be used by users without administrator rights.

If a problem arises when starting the driver, the error cause can be taken from the *system event log file* of Windows NT 4.0.

2.5.1.5 De-Installation of Software Driver

The driver is de-installed in three steps:

Attention: Stopping the driver and the complete de-installation can only be done with administrator rights on the Windows NT computer!

1. Terminating all applications which use the services of the driver.

2. The driver has to be stopped in dialogue *control panel/devices* or by entering `net stop` *`xxxx`* in the command line.

CAN Module	Commands for Stopping the Drivers
CAN-ISA/200	`net stop c200i`
CAN-PC104/200 (SJA1000 version)	
CAN-ISA/331	`net stop c331i`
CAN-PC104/331	
CAN-PCI/200	`net stop c200`
CAN-PCI/266	
CPCI-CAN/200	
PMC-CAN/266	
CAN-PCI/331	`net stop c331`
CPCI-CAN/331	
PMC-CAN/331	
CAN-PCI/360	`net stop c360`
CPCI-CAN/360	
CAN-PCC	`net stop cpcc`

Table 7: Commands for stopping the drivers

3. Select entry for the CAN driver in the folder *control panel* in dialogue box *add/remove program properties* and press icon *add/remove*, in order to delete all files and registry entries of the driver.

2.5.2 Event Logging

All error situations at start or during operation are registered in the *system event log file* of Windows NT 4.0.

2.6 Windows 95/98/ME Installation

The Windows 95/98/ME software supports a maximum of 5 CAN modules of the same type within one system. The number of different CAN modules in one system is not limited.

Note: The CAN-USB-Mini module does not run under Windows **95**!

2.6.1 Installation of PCI-Bus Boards and CAN-USB-Mini

1. Install the board according to the hardware installation description.

2. After run-up Windows 9x/ME detects the board and opens an information window. If Windows has not found the driver (e.g. if it is the first installation), you are asked to insert the data carrier (disk or CD-ROM) with the driver.

3. At the end of the software installation you have to shut down the computer. Terminate all active applications before you shut down the computer. After restart, the driver is automatically started and in the device manager the new device type *CAN Controller* now exists, which can be used to display and edit all *esd*-CAN boards.

4. Now install the SDK as being described on page 53. The SDK contains the files which are required for developing your own applications as well as the monitor program CANreal.

Special features at installation of CAN-PCI/360:

CAN-PCI/360-modules which were manufactured before 01.01.2000 report their maximum memory upgrade of 128 Mbyte to the system during installation, even if they have got less capacity. These boards might get into conflict with graphics boards which act similarly. For these boards the actual memory space requirement can be configured via the Device Manager. More about the Device Manager can be found in chapter 'Changing the Resources Settings via the Device Manager' on page 39.

CAN-PCI/360-modules which were manufactured after 01.01.2000 do not have this problem anymore.

2.6.2 Installation of ISA-Bus Boards and CAN-PCC

2.6.2.1 Comment on the Hardware Installation of ISA-Boards

Install the board only in the computer **after** you have determined the right configuration (address area) by means of the Windows-9x/ME system control.

After the installation described below, you have also to install the SDK (see page 53).

CAN-2.0B Support of the CAN-ISA/200 and CAN-PC104/200:

Starting from driver revision 1.2.0 you can select between a driver for 11-bit identifiers (CAN 2.0A) and a driver for 29-bit-identifiers (CAN 2.0B) during the installation of CAN-ISA/200 and CAN-PC104/200. The CAN 2.0B-driver is slightly slower than the CAN 2.0A-driver, even if you only use it to transmit and receive 11-bit-identifiers, because the CAN controller has to be accessed more frequently.

Prerequisite for the CAN 2.0B-operation is that your module has not got a CAN controller of type 82C200 (delivered until 12/1999), but has got an SJA1000, for instance.

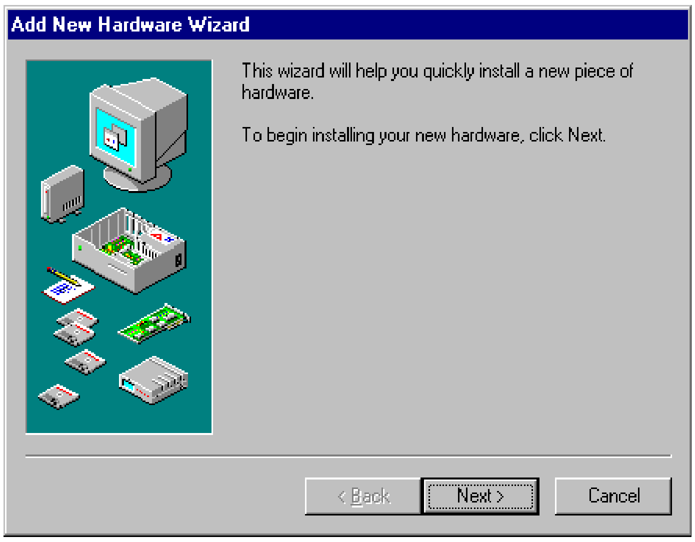
2.6.2.2 Installation of the Device Drivers

1. Call

   The Windows-9x/ME driver of the CAN-ISA boards is installed by means of the *Hardware Wizard.* It is started in the *System Control Panel* window by selecting the *Hardware* icon:

   Hardware

   After that the start window of the hardware wizard has to open:

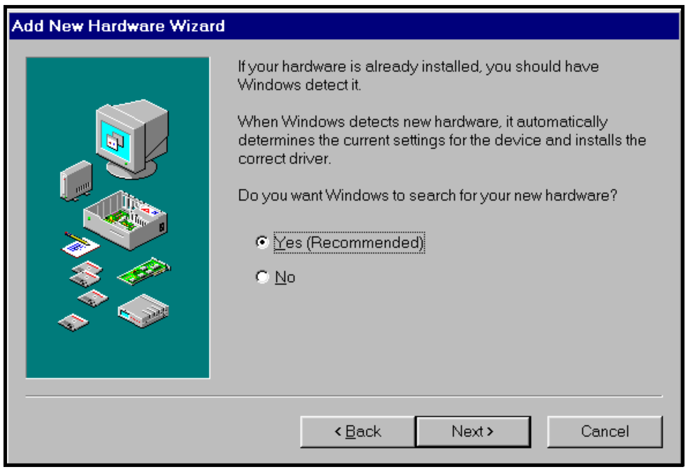


Add New Hardware Wizard

This wizard will help you quickly install a new piece of hardware.

To begin installing your new hardware, click Next.

2. The following window asks, if the hardware is to be searched for. Since the board is unknown to the system (no plug-and-play), ***No*** has to be selected. If *Yes* is selected, the computer will search for the board for a relatively long time and then report the discovery of an unknown board.



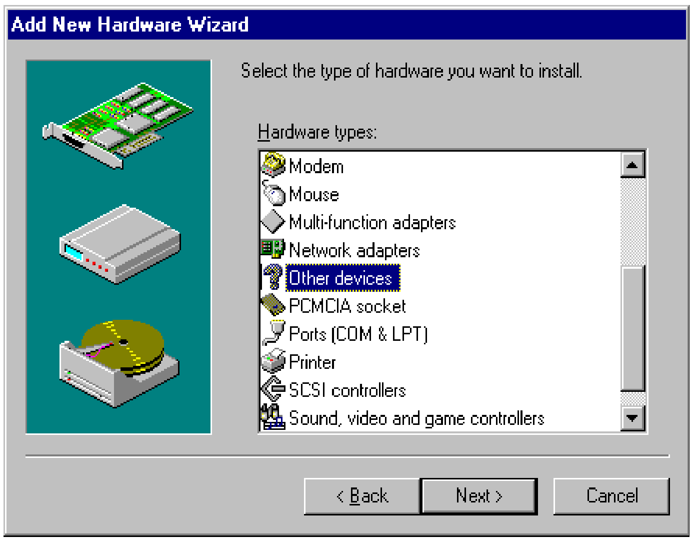
If your hardware is already installed, you should have Windows detect it.

When Windows detects new hardware, it automatically determines the current settings for the device and installs the correct driver.

Do you want Windows to search for your new hardware?

[✓ Yes (Recommended)] No

3. Then the window for the selection of the type of hardware to be installed appears.



Select the type of hardware you want to install.

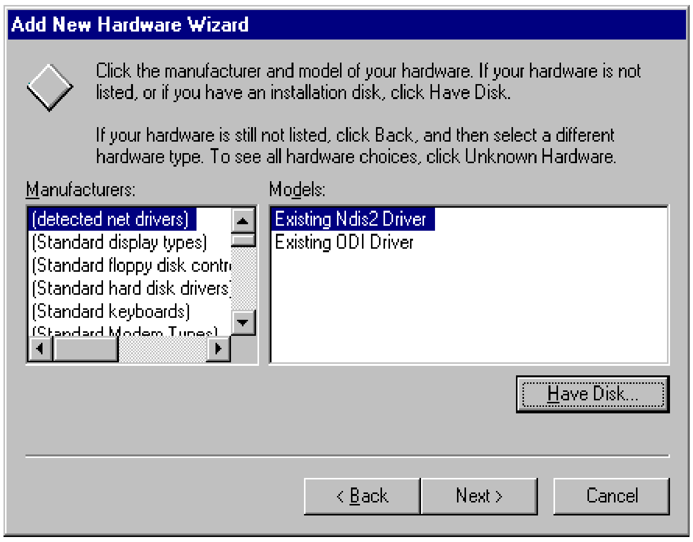
Hardware types:

- Modem
- Mouse
- Multi-function adapters
- Network adapters
- [✓ Other devices]
- PCMCIA socket
- Ports (COM & LPT)
- Printer
- SCSI controllers
- Sound, video and game controllers

Here *Other devices* has to be selected and then *Next* has to be clicked. (Only if already an esd-CAN driver had been installed, the selection *CAN Interface* would appear. It is not shown before or during the installation.)

If you haven't already done so, you should now put the data carrier (disk or CD-ROM), contained in the product package, into your drive.

4. In the window which opens after the hardware-type selection first *Have Disk* and then *Next* have to be clicked.

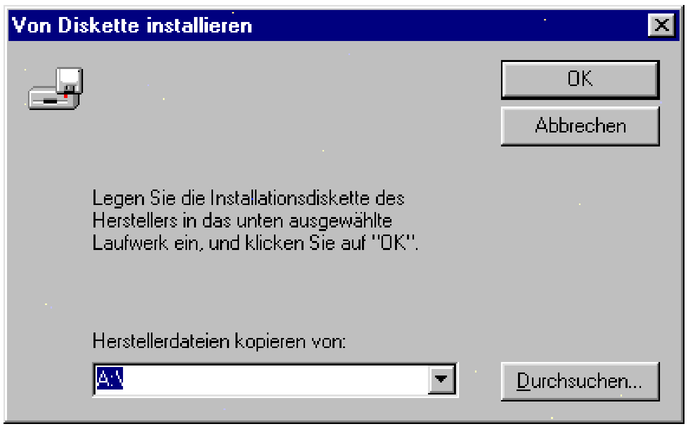


Add New Hardware Wizard

Click the manufacturer and model of your hardware. If your hardware is not listed, or if you have an installation disk, click Have Disk.

If your hardware is still not listed, click Back, and then select a different hardware type. To see all hardware choices, click Unknown Hardware.

5. The following window appears. Click the button *Find.*

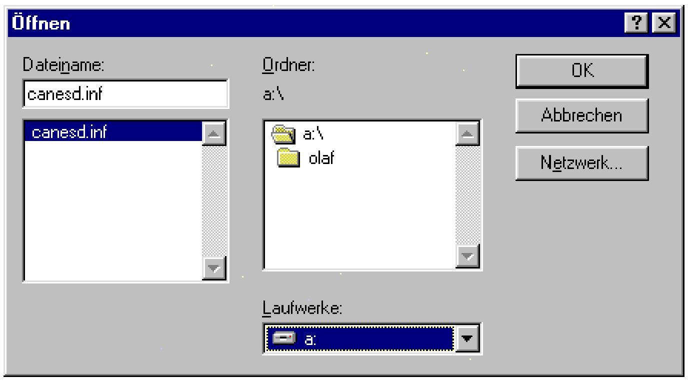


Von Diskette installieren

Legen Sie die Installationsdiskette des Herstellers in das unten ausgewählte Laufwerk ein, und klicken Sie auf "OK".

Herstellerdateien kopieren von: A:\

6. Now select the file `canesd.inf` and click OK.



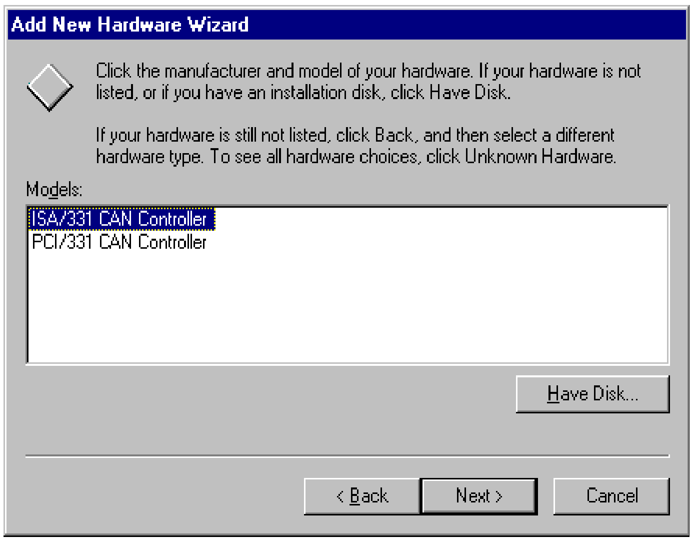
Öffnen

Dateiname: canesd.inf

Ordner: a:\

Laufwerke: a:

7. The following window, which displays all esd-CAN drivers, has to open (this example only shows the ISA/331 driver). Select the driver of your board and click *Next*.


Add New Hardware Wizard

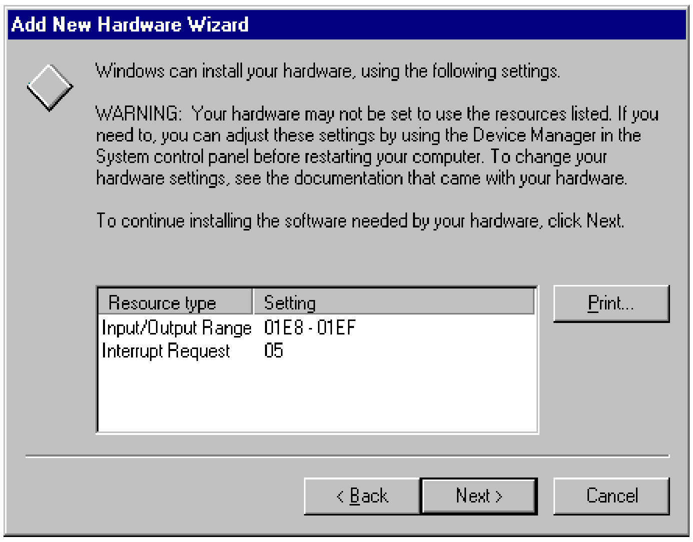
Click the manufacturer and model of your hardware. If your hardware is not listed, or if you have an installation disk, click Have Disk.

If your hardware is still not listed, click Back, and then select a different hardware type. To see all hardware choices, click Unknown Hardware.

Models:

ISA/331 CAN Controller
PCI/331 CAN Controller

8. (Only for CAN-ISA boards, otherwise continue with step 9) Windows 9x/ME installs the driver, checks the system resources, compares them to the configurations possible for the CAN-ISA board and offers a possible configuration for the board in the following window:


Add New Hardware Wizard

Windows can install your hardware, using the following settings.

WARNING: Your hardware may not be set to use the resources listed. If you need to, you can adjust these settings by using the Device Manager in the System control panel before restarting your computer. To change your hardware settings, see the documentation that came with your hardware.

To continue installing the software needed by your hardware, click Next.

Resource type	Setting
Input/Output Range	01E8 - 01EF
Interrupt Request	05

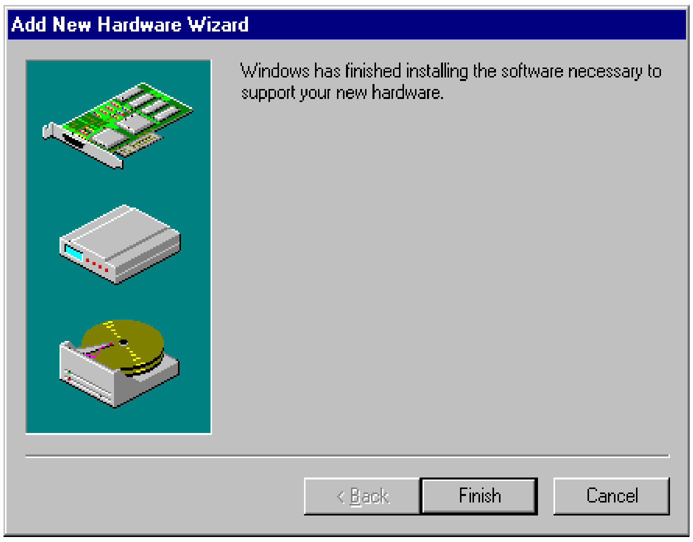
The *Input/Output range* proposed by the system control does not have to correspond to the default-I/O range which is set on board by means of jumpers or coding switches.

Attention: Therefore it is absolutely necessary to compare the jumper (or coding switch) position on the CAN-ISA board to the I/O-address space selected by the system, and change the jumpers/coding switches, if required.

Click *Next*

If the system does not propose settings:
If the system says that it does not have any resources for the CAN-ISA board, the installation has to be completed nevertheless. In this case the resources in the system have to be distributed again manually by means of the device manager after the installation.

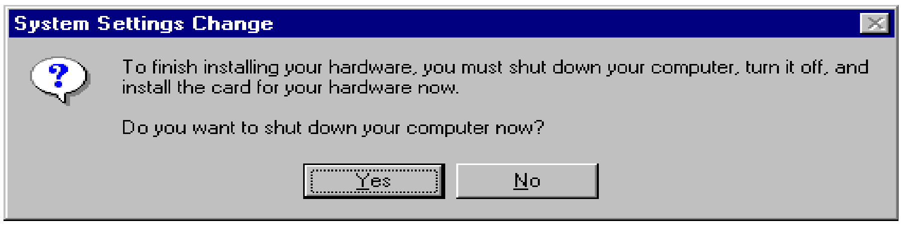
9. The successful installation of the software driver is shown by the following window:



Add New Hardware Wizard

Windows has finished installing the software necessary to support your new hardware.

10. In order to complete the software installation you have to shut down your computer after terminating all applications which are still open!



System Settings Change

To finish installing your hardware, you must shut down your computer, turn it off, and install the card for your hardware now.

Do you want to shut down your computer now?

Then switch off the computer and **install the hardware *now*** as described in the hardware manual! Compare the default setting to the setting selected under point 8 and change the jumpers (or coding switches, depending on board type), if necessary.

11. Switch on the computer again after the installation and restart Windows 95. The driver is automatically loaded by Windows 95. The device manager now has the new device class *CAN Controller* under which all *esd* CAN boards can be shown and configured.

Now install the SDK, as described on page 53. The SDK contains the files that are necessary for the development of your own applications and it contains the monitor program CANreal.

2.6.2.3 Starting the Device Drivers

2.6.2.3.1 ISA-Bus Boards

Switch on the computer again after installation and restart Windows 9x/ME. The driver is automatically loaded and started by Windows 9x/ME. In the device manager there is now the new device class *CAN Controller*, under which all *esd*-CAN boards can be shown and configured.

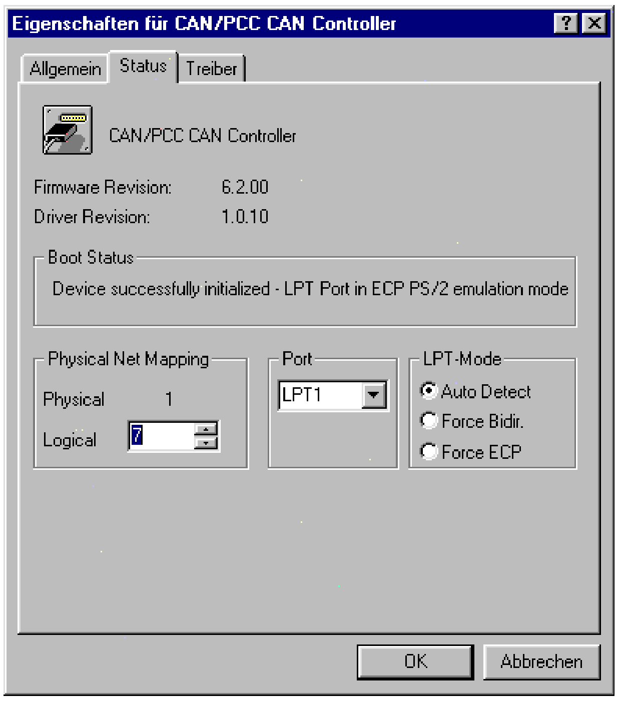
2.6.2.3.2 CAN-PCC

Switch on the computer again after installation and restart Windows 9x/ME. The driver is automatically loaded by Windows 9x/ME but not started. In the device manager, however, the new device class *CAN Controller* exists, under which all *esd*-CAN boards can be shown and configured.

For starting the driver the command `cpcc start` has to be entered explicitly in the command line. This is necessary to be able to share the parallel interface with other device drivers. After successfully starting the driver, it has an exclusive access to the parallel interface until it will be stopped again by means of `cpcc stop`.

Another reason for executing an explicit start command is that the driver is not being assigned with its own resources within Windows 9x/ME, but that it uses the available resource ‘Parallel Interface’. Therefore no smart icon *Resources* (see the following chapter) exists for this driver within the device manager. In smart icon *Status* the parallel interface to be used (LPT1, LPT2, LPT3) has to be entered below port instead and the driver automatically uses the current Windows-9x/ME configuration (I/O-address, interrupt) for this interface.

Attention:
For the operation of the CAN-PCC under Windows 9x/ME the parallel port has to be set to operating mode *Force ECP* in the properties window. If the hardware does not support this setting, the bidirectional mode has to be selected (*Force Bidir.*).


Eigenschaften für CAN/PCC CAN Controller

Allgemein | Status | Treiber

CAN/PCC CAN Controller

Firmware Revision: 6.2.00
Driver Revision: 1.0.10

Boot Status
Device successfully initialized - LPT Port in ECP PS/2 emulation mode

Physical Net Mapping
Physical 1
Logical 7

Port
LPT1

LPT-Mode
[✓ Auto Detect]
Force Bidir.
Force ECP

2.6.2.4 Changing the Resource Settings by Means of the Device Manager

2.6.2.4.1 Overview

Attention:	Changing the default settings by the device manager or the registry editor can cause conflicts so that one or more devices are not recognized by the system anymore! The device manager and the registry editor are configuration tools for advanced users who are familiar with the parameter configuration and know that changes can have various effects.

The CAN-ISA boards have fixed resource settings which are either distributed by the hardware assistant during the Windows setup or which can be configured later by the device manager.

It might occur that Windows 9x/ME is unable to configure the CAN-ISA board, because it gets into conflict with other devices. Should this be the case the board has to be configured again.

In order to change the device setting manually, the *Device Manager*, which is called via *Settings* under *System Control Panel*, can be used. By using the device manager errors can be prevented which are likely to occur when editing the registry entries directly.

If you want to solve the device conflicts manually by means of the device manager, you can use following strategies, for instance:

☐ If the conflicting device is a plug-and-play device, deactivate it to release its resources.

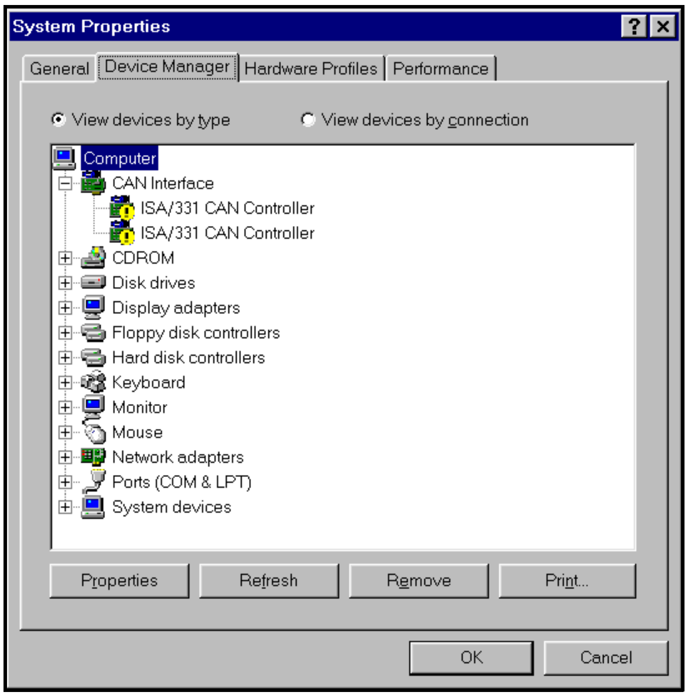
☐ If the conflicting device is a conventional device, deactivate it by removing the board from the system and unloading the drivers.

☐ Distribute the resources which are used by other devices again in order to release resources for the conflict device.

☐ Change the jumpers (or coding switches, depending on the board type) on the CAN-ISA board in order to adjust the board to the new settings.

2.6.2.4.2 Activating the Device Manager

1. Call
   Click icon of *Device Manager* in *Settings* under *System Control Panel*.
   or
   Click *My Computer* with right mouse button, click *Properties* in *Context* menu and then click the *Device Manager*.



Computer
- CAN Interface
  - ISA/331 CAN Controller
  - ISA/331 CAN Controller
- CDROM
- Disk drives
- Display adapters
- Floppy disk controllers
- Hard disk controllers
- Keyboard
- Monitor
- Mouse
- Network adapters
- Ports (COM & LPT)
- System devices

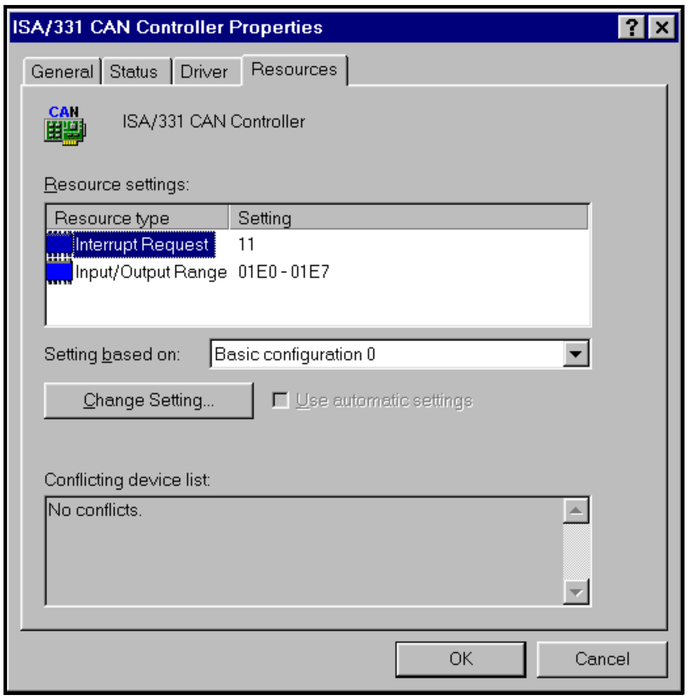
2. Double clicking the desired device type in the list with the left mouse button lists all devices of this type in the computer.

3. Select the device to be configured by double clicking. Or else mark the device and click the *Properties* icon.

2.6.2.4.3 Changing the Resource Settings by Means of the Device Manager

1. Select the device class *CAN Controller* in the device manager by double clicking.
   The tree branches and shows all devices of this type available in your computer.

2. Double clicking a device opens up its property window.
   Click the *Resources* icon under the device properties.



Resource settings:

Resource type	Setting
Interrupt Request	11
Input/Output Range	01E0 - 01E7

Setting based on: Basic configuration 0

☐ Use automatic settings

Conflicting device list:

No conflicts.

The *Conflicting Device List* shows the settings of other devices which are conflicting the current setting of the CAN-ISA board.

3. Select the setting which is to be changed, for instance the interrupt level, under *resource type*. Click the *Change setting* icon in order to keep the changed values.
   Changes can only be made, if the option *Use automatic settings* has been deactivated.
   The *Interrupt* and *Input/Output-range* (address space) settings can be changed independently.

   The dialogue box *Input/Output range* shows the various settings which are supported. An interrupt which is marked by an asterisk (*) signifies that this interrupt is already used by another device.

   After clicking the *Change setting* icon an error message might appear which states that the resource setting cannot be changed. In this case you have to select other settings until the system accepts one of the chosen settings.

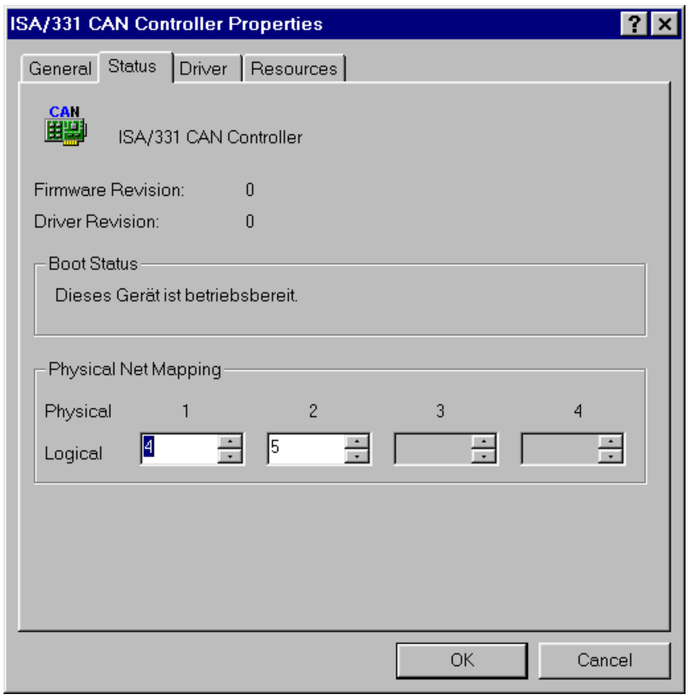
4. Select settings which are not conflicting other devices and click ‘OK’. **Remember that the I/O-range setting by jumper (or coding switches, depending on the board type) has to comply with the selected setting!**

5. Finish Windows 9x/ME, then change the hardware settings of devices which have been configured again and restart Windows 9x/ME.

2.6.2.4.4 Changing the Logical Network Number

1. Double click the device class *CAN Controller* in the device manager.
   The tree branches and all devices of this type available in your computer are shown.

2. Double clicking a device opens its property window.
   Click the *Status* icon in the property window.



ISA/331 CAN Controller

Firmware Revision: 0
Driver Revision: 0

Boot Status
Dieses Gerät ist betriebsbereit.

Physical Net Mapping

Physical	1	2	3	4
Logical	4	5		

3. In *Physical Net Mapping* logical network numbers can be assigned to the physical interfaces of the CAN board. By means of the network numbers the CAN board can be addressed by the software. When starting the driver for the first time, logical network numbers starting from 0 are assigned to boards supported by the driver. If more than one esd CAN board of various types (e.g. CAN-PCI/331 and CAN-ISA/331) are used in a computer, overlapping network numbers cannot be avoided and therefore have to be set manually.

4. A change in logical network numbers remains invalid until the computer is restarted.

2.7 RTX Installation

RTX is a real-time extension for Windows NT/2000/XP that provides high-speed and deterministic real-time capabilities. The extension is developed and distributed by the company ‘Ardence’.

2.7.1 Implementation

The support for an esd CAN module consists of a driver part and the NTCAN library part. The hardware specific driver part has to be loaded for each supported CAN module the library part has to be loaded once into the RTSS environment. Driver and library are implemented as RTSS DLL, which means that they are RTSS processes that export functions for use by other RTSS processes, share a common address space with other RTSS processes and accurately mirror the automatic resolution of references to exported functions like implicitly loaded Win32 DLLs.

The driver doesn’t export any functions and an RTSS application will never reference to the driver directly. An RTSS application will always reference to the API functions exported by the NTCAN library described in this manual. The NTCAN RTSS DLL may be loaded without a driver to create RTSS applications which refer to the NTCAN API even if the CAN hardware isn’t present in all cases.

2.7.2 Software Requirements and Differences to Win32

Software Requirements:

- Ardence RTX version 5.x or 6.x

Differences versus Win32 implementations of the NTCAN-API:

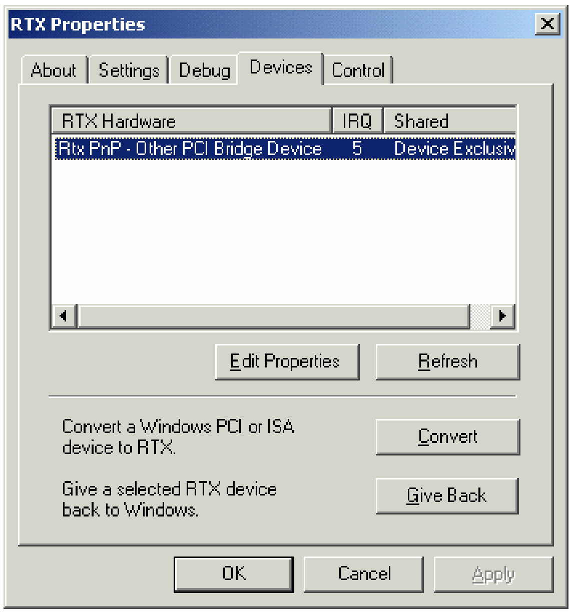
- Overlapped operations are not supported in the RTSS environment.
- Unlike to Win32 the Tx-timeout is a transaction timeout, i.e. the timeout is not valid for the transmitted CAN messages but for a complete transmission job.
- Firmware update in the RTSS environment is not supported. If a firmware update is necessary, it has to be performed in the Win32 environment with the appropriate Windows device driver.

2.7.3 Installation

In order to support the CAN modules in the RTSS environment it is necessary to make RTX manage these devices. For Windows NT the device has to be disabled through the Windows control panel because there are no Plug and Play or Power Management concerns with Windows NT (skip **Step 1** below). Managing devices on Windows 2000 or Windows XP requires converting a Windows device into an RTX device (see *RTX Runtime Documentation* for details).

Step 1: Converting to RTX device

Before the RTSS device driver can control the device the CAN module has to converted into a RTX controlled device. This is performed using the **RTX Properties** control panel applet.
Select the **Devices** Tab (RTX 5.x) or **Plug and Play** Tab (RTX 6.x). In this applet under *Devices* the CAN module has to be selected and assigned to the RTX driver according to the following figure:



RTX Hardware	IRQ	Shared
Rtx PnP - Other PCI Bridge Device	5	Device Exclusiv

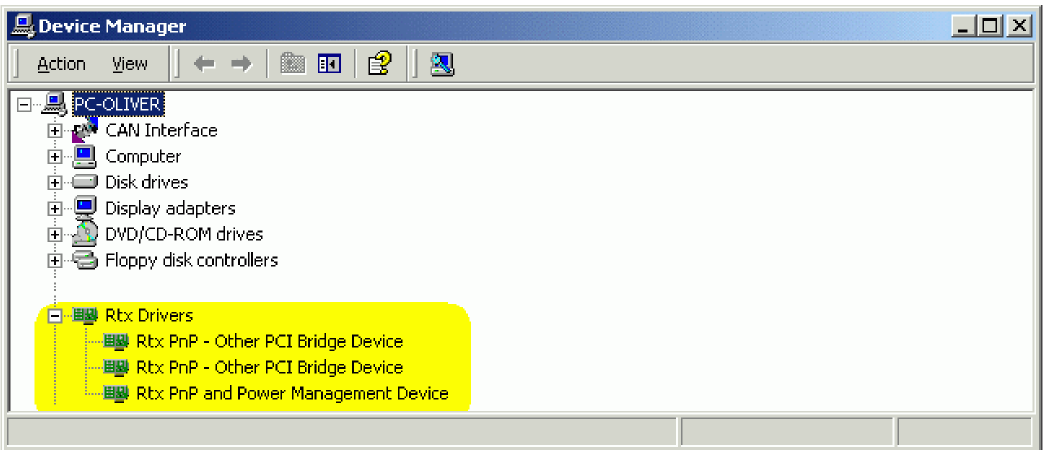
Convert a Windows PCI or ISA device to RTX.

Give a selected RTX device back to Windows.

If there has already been installed a driver under Windows 2000/XP for this CAN module it will be listed as *CAN-Interface*. If not, it will be listed as *Other PCI Bridge Device*.

After the conversion the CAN module has to be removed by means of the device manager and the computer has to be rebooted. You will find detailed information on this procedure e.g. in the release notes of RTX 5.0 in the chapter *Managing RTX Devices in Windows 2000*.

After reboot the CAN modules has to listed in the device manager as shown in the following figure:



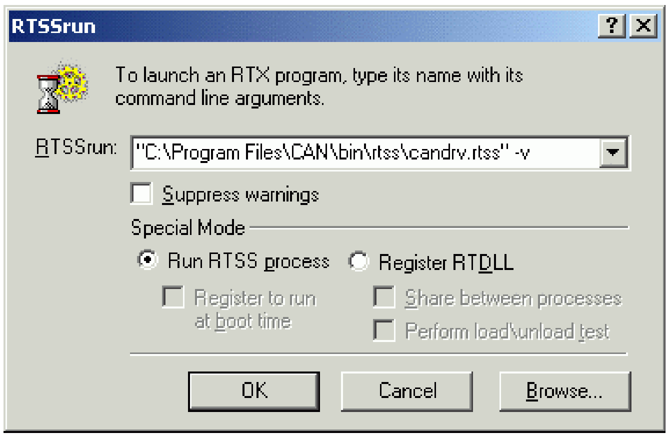
Step 2: Starting the driver

Next step is to start the driver. This can be performed with the command line version or the GUI-version of the RTX-tool *RTSSStart*. At the command line the following command has to be executed:

```
rtssstart driver.rtss [Arguments]
```

where `driver.rtss` has to be replaced with the name of the driver (e.g. `c331.rtss`).

In the GUI-version the start may be performed like this:


RTSSrun

To launch an RTX program, type its name with its command line arguments.

RTSSrun: "C:\Program Files\CAN\bin\rtss\candrv.rtss" -v

If an error occurs at start the numeric error code is displayed in the RTX-text console and the RTSS-process is terminated. In this case the driver can be started with the option ‘-v’ to submit the RTX-text console more tips to the source of the error.

If an error occurs in the start sequence it can be checked with the RTX-task manager, respectively the RTX-object viewer, if the driver is active.

Step 3: Loading the NTCAN API library

Next step is to load the NTCAN API library. This can be performed with the command line version or the GUI-version of the RTX-tool *RTSSStart*. At the command line the following command has to be executed:

```
rtssstart ntcan.rtss [Arguments]
```

If an error occurs at start the numeric error code is displayed in the RTX-text console and the RTSS-process is terminated.

If an error occurs in the start sequence it can be checked with the RTX-task manager, respectively the RTX-object viewer, if the driver is active.

2.7.4 Command Line Parameter for RTSS Device Driver

There are several command line options to configure the driver startup:

Option	Argument	Description
-V	-	After the start the driver returns the version number in the RTX-console and terminates immediately.
-v	*mask*	Additional debug output during startup in the RTX-console. The mask parameter may be given decimal or hexadecimal with leading ‘0x’
-h	-	Display a list of all possible command line options and terminate immediately.
-p	*prio*	This option configures the priority of the IST thread that processes the CAN messages. The default value for *prio* is 127.
-b	*baudrate*	This option configures the default baudrate according to the baudrate table. The default value for baudrate is to stay off-bus as long as the baudrate is set explicitly by the application.
-n	*net number*	This option configures the logical base net number which is assigned to the first physical port of the CAN module. The logical net numbers are incremented by one for each additional physical port. The default value for *net number* is 0.

Example:

The call

```
        c405.rtss -v0xFF -n2 -p118
```

- starts the driver with additional outputs and
- assigns the logical base-net number ‘2’ and an IST (interrupt service thread) priority of ‘118’ to the first CAN module and

If the driver is started without any command line options the logical net number 0 is assigned to the first physical CAN port, all IST threads get a priority of 127 and the CAN modules remain off-bus until a baudrate is explicitly assigned by the application.

2.8 Windows CE.NET Installation

Sorry no installation notes available for Windows CE.NET at the moment.

2.9 Installation and Configuration of the CAN-Bluetooth Module

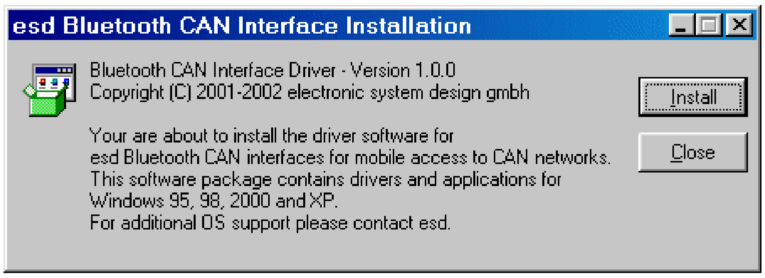
The CAN-Bluetooth module runs under the operating systems Windows 2000 and Windows XP. This chapter explains the installation and general configuration of the module. The further configuration via the integrated Web-server is described in the CAN-Bluetooth manual.

Attention: The CAN Bluetooth driver must always be installed after the CAN-SDK has been installed! If different esd-CAN modules are used in one system, the CAN Bluetooth driver must always be installed last!

2.9.1 Installation

1. Call installation program `canbtdrv.exe`.

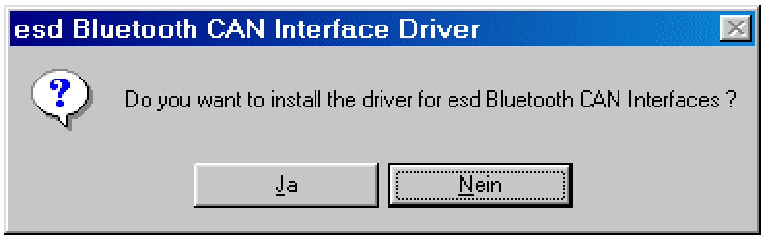
2. After the window *esd Bluetooth CAN Interface Driver* has opened, click *Install*.



Bluetooth CAN Interface Driver - Version 1.0.0
Copyright (C) 2001-2002 electronic system design gmbh

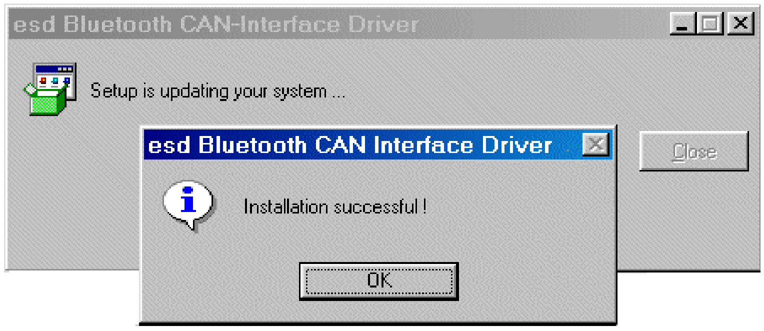
Your are about to install the driver software for
esd Bluetooth CAN interfaces for mobile access to CAN networks.
This software package contains drivers and applications for
Windows 95, 98, 2000 and XP.
For additional OS support please contact esd.

3. Start Installation



Do you want to install the driver for esd Bluetooth CAN Interfaces ?

4. The computer shows whether the installation was successful or not.

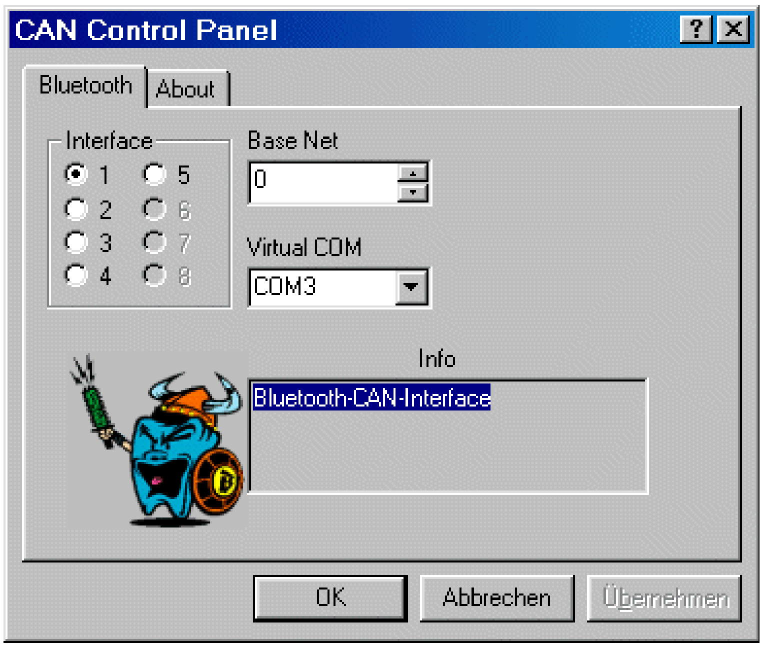

esd Bluetooth CAN-Interface Driver
Close
esd Bluetooth CAN Interface Driver
OK

Setup is updating your system ...

Installation successful !

2.9.2 Configuration

After the installation you will find in menu *Start* under *Programs* the configuration program *esd Bluetooth CAN Interface Driver.* Here you select *CANBT Config* and the window *CAN Control Panel* opens.



Interface: (● 1) ○ 2 ○ 3 ○ 4 ○ 5 ○ 6 ○ 7 ○ 8
Base Net: 0
Virtual COM: COM3
Info: Bluetooth-CAN-Interface

Here the following settings can be made:

Interface	The software supports a maximum of five different esd-CAN Bluetooth interfaces. Here you select the interface that you want to configure.
Base Net	Here the logical CAN network number is entered. Values between 0 and 255 are permissible.
Virtual COM	The Bluetooth interface is managed as virtual COM-interface. Values between 1 and 11 are permissible.

Note: Bluetooth devices usually use virtual serial interfaces for communication. The assignment between COMx and Bluetooth device is configured by means of the software which controls your Bluetooth hardware (such as PC-Card).

2.10 Installation and Configuration of the EtherCAN Module

The EtherCAN client driver supports Windows NT, Windows 2000, Windows XP, Windows Server 2003 and Windows Vista (32- and 64-bit versions). This chapter explains the installation and general configuration of the module.

Note: Before host software installation the EtherCAN module should be assigned an IP address within the subnet.
The description of the IP-address assignment and the further configuration via the integrated Web-server are described in the manual 'EtherCAN, Hardware Manual'.

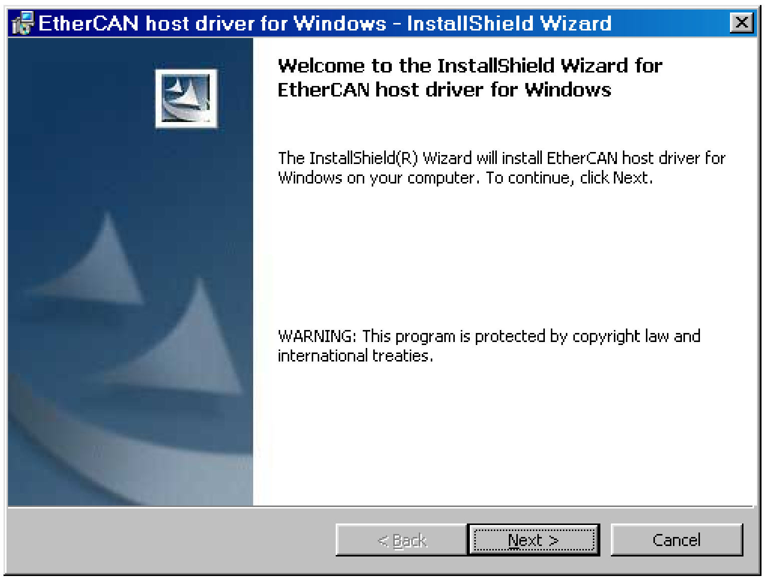
2.10.1 Installation

1. Start the installation program `setup.exe` which is located in the directory `EthetCAN/Win32` of the CAN driver CD

2. Follow the instructions of the installation program.

3. When the *Setup Type* is requested, please choose *Typical.*

4. The success or failure of the installation will be displayed at the end of installation.


EtherCAN host driver for Windows - InstallShield Wizard

Welcome to the InstallShield Wizard for EtherCAN host driver for Windows

The InstallShield(R) Wizard will install EtherCAN host driver for Windows on your computer. To continue, click Next.

WARNING: This program is protected by copyright law and international treaties.

2.10.2 Configuration

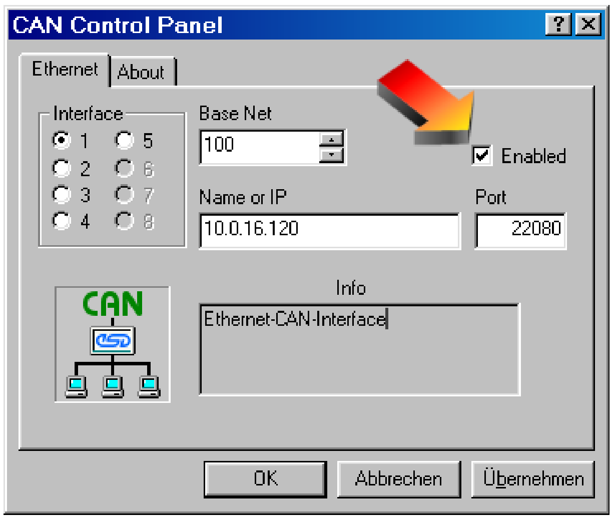
The EtherCAN driver is implemented as a user mode driver which can be integrated with a logical net number as other CAN modules for NTCAN-API based applications.

CAN Control

After the installation you will find the configuration dialogue in *Start / Settings / System /CAN Control*. You have to use this dialogue to create logical EtherCAN ports which link the remote EtherCAN module to a logical NTCAN net number.

Start CAN Control by a double click on the icon.

You have to configure the following parameter:



Interface: [✓ 1] 2 3 4 5 6 7 8
Base Net: 100
[✓ Enabled]
Name or IP: 10.0.16.120
Port: 22080
Info: Ethernet-CAN-Interface

Interface In the drop down box you can choose the EtherCAN port which is to be configured. With the *Add* button you can create up to 20 logical EtherCAN ports. With the *Delete* button you can remove the currently selected EtherCAN port. The default after installation are 5 EtherCAN ports. The first five ports can't be deleted.

Virtual CAN port
You have to assign a virtual network number between 0 and 255 and have to check the enabled box before this port is available in your software. Please make sure that this logical net number isn't already used by another EtherCAN module or a physical CAN module.

TCP/IP Configuration
Setup the IP address or hostname (if known by a DNS server) of the EtherCAN device. The description of the IP-address assignment within the system is described in the manual 'EtherCAN, Hardware Manual'. A change of the default port 22080 isn't supported at the moment.

Timeouts Currently three separate timeouts can be configured.
The *connection timeout* defines the time the EtherCAN client software waits for a response during the initial connection before the client software returns with a timeout.
The *command timeout* defines the time after which a request to the EtherCAN module has to be replied before the client software returns with a timeout.
The *keep alive* timeout defines the time after which a keep alive request has to be replied by the EtherCAN module before the host driver tries to reset and re-establish the connection.

Default The *Default* button restores all client driver defaults for the *timeout* parameter.

2.11 Installing the SDK (Software Development Kit)

After driver installation you can proceed with the installation of the CAN-SDK. It contains the necessary files, documentation and tools to develop, debug and test your own NTCAN-API based applications for all Windows versions.

Note:	If you have installed an older version (V 1.x) of the CAN SDK uninstall this version before you install this new version. If you have already installed a CAN SDK revision 2.x you can overwrite the older installation.

To install the CAN SDK start `setup.exe` in the directory CAN_SDK of the CAN driver CD. The installer is digitally signed.

2.11.1 Installation Options

At the start of the installation you are asked for the installation language. Currently an installation in English and German is supported. The installation language also defines the language of the installed documentation, if available in both languages.

Proceeding the setup you can choose between the *full installation*, the *compact installation* and a *custom installation*. The *full installation* installs everything. The *compact installation* installs only the files which are necessary for software development. The *custom installation* lets you choose which components of the SDK are to be installed. The categories currently available are Tools, Documentation, Sample Code and Software Development files.

2.11.2 IDE Integration

Many Integrated Development Environments (IDEs) allow to define paths relative to an environment variable. For this reason during the installation the environment variable `CanSdkDir` is created which is set to the installation directory of the CAN SDK. Using this environment variable in paths makes a project independent of the installation directory of the CAN SDK.

2.11.3 Supported Programming Languages and Development Environments

Please refer to the release notes which gets installed with the CAN SDK for a complete list of supported programming languages and development environments.

3. Firmware Update and CAN2.0A/B-Selection for Windows OS

3.1 Firmware Update

3.1.1 Overview

The update function of the local firmware has no effect on the boards CAN-xxx/200 , CAN-PCI/266, CPCI-CAN/200 and PMC-CAN/266, because these boards do not have a local CPU. The module CAN-PCC also does not support the update function.
The firmware of the EtherCAN modules can be updated by means of a standard FTP-client. The firmware update mode is activated by means of a web-browser (see manual 'EtherCAN, Hardware Manual').

The firmware of the local microprocessor of the intelligent CAN boards is stored in the Flash EPROM. When the board is shipped you will receive the firmware version which had been the latest at the time of the *hardware manufacturing*.

The disks or CD-ROM, included in the shipping, contain an update program by means of which you can program the flash EPROM with the latest software at the *time of shipping*.

Depending on the module and the operating system you will either find the update program '**`canupd`**' or the update program '**`upd`*`xxx`***' on your data carrier. The programs will be installed onto your hard disk during the installation of the driver.
Both programs are equal in functionality. While `canupd` is a universal update program for various modules, `upd`*`xxx`* offers specific updates for individual modules.

The modules are identified by means of the character combination when called:

CAN Module	Entry Syntax for `upd`*`xxx`*
CAN-ISA/331 CAN-PC104/331	`updc331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`updc331`
CAN-PCI/360 CPCI-CAN/360	`updc360`
CAN-PCI/405	1*)
CAN-USB-Mini	`updusb331`
CAN-USB/2	`updusb2292`

Table 8: Entry syntax for update program `upd`*`xxx`*

1*) In contrast to the other CAN modules the firmware-update of the CAN-PCI/405 is automatically executed during the driver's installation. No extra action is required.

3.1.2 Checking the Firmware Version Number

With the update program you can check the local firmware revision of the Flash EPROM and update the Flash EPROM, if necessary.

It is recommended, to compare the revision number of the delivered firmware at the floppy disk with the firmware stored in the Flash EPROM after the host driver installation!

1. The software driver of the host CPU has to be started before `canupd` or `upd`*`xxx`* is started (refer to previous chapters).

2. Open a window and switch to the installation directory.

3. Start `canupd` using the windows command shell
   Calling the program without parameters returns the output of the included firmware program with revision numbers onto the monitor. Because *canupd* is a universal tool, it includes firmware programs for several hardware applications.

   Calling the program with the following number of the logical network number of the required board returns the output of the revision numbers of the local firmware stored in the Flash EPROM onto the monitor.

   Input example: `canupd 0`

(Alternatively the firmware revision and the driver revision of the host CPU can be checked using the *event log* after the driver is started. Additionally the *event log* shows the firmware revisions which are suitable for the software driver shown.)

3.1.3 Executing the Update

1. Calling `canupd` (or `upd`*`xxx`*) with the logical net number (see above).

2. After the version control the user is asked, if the program is to be continued with the update or not.

   - **If 'y' for YES is typed, the Flash EPROM will be reprogrammed regardless of the firmware version which is more up-to date !**

   - If 'n' for NO is typed, the program is interrupted. The old firmware remains unchanged in the Flash-EPROM.

3. To activate the new firmware the local microprocessor has to be reset. This is done by stopping and restarting the software driver of the host CPU, e.g. by the command line:

   ```
   net stop xxxx
   net start xxxx
   ```

   At Windows XP systems this is executed if the device is enabled or disabled at the device manager or via a restart of the computer.

CAN Module	Entry for *xxxx* in the Command Line
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`
CAN-PCI/360 CPCI-CAN/360	`c360`
CAN-USB-Mini	`usb331`
CAN-USB/2	`usb2292`

Table 9: Driver names at the entries of the command line

3.2 Changing between CAN2.0A and CAN2.0B-Operation

All CAN modules which support the update function under Windows also offer the possibility to choose between operating the driver with 11-bit CAN identifiers (CAN 2.0A) or 29-bit CAN identifiers (CAN 2.0B).

Note:	If the 29-bit option shall be used at a module (CAN2.0B), this must be state in order (order no. C.1102.01). When shipped the default setting of the modules is '11-bit CAN identifier' (CAN 2.0A). Only at modules that have been ordered with the option C.1102.01 the default setting is '29-bit CAN identifier'.

The entry into the command line has to be the following:

CAN Module	Entry Syntax for CAN2.0A (11-bit-ID)	Entry Syntax for CAN2.0B (29-bit-ID)
CAN-ISA/331 CAN-PC104/331	`updc331i -ta` *`netnumber`*	`updc331i -tb` *`netnumber`*
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`updc331 -ta` *`netnumber`*	`updc331 -tb` *`netnumber`*
CAN-PCI/360 CPCI-CAN/360	-	-
CAN-USB-Mini	`updusb331 -ta` *`netnumber`*	`updusb331 -tb` *`netnumber`*
CAN-USB/2	`updusb2292 -ta` *`netnumber`*	`updusb2292 -tb` *`netnumber`*

Table 10: Changing to CAN2.0A or CAN2.0B

Parameter Description:

`netnumber` ... The logical net number is has to be set with this parameter (0, 1, 2, ..). The parameter has to be set mandatory! The value has to be entered separated by an blank from the preceding characters.

Attention: The operating mode selected for one net is valid for all other nets of the module, too!

In order to activate the new setting the local processor has to be reset. Resetting and rebooting are described above under step 3.

4. Unix Operating Systems

4.1 Linux Installation

Notes on the Linux Driver History:

Driver versions *from* V3.x.x (compilation is executed by customer)
In this driver package the *customer* executes the compilation of the driver. Therefore a flexible usage of different Linux kernels is possible.

Driver versions *up to* V3.x.x (compilation is executed by esd)
The Linux drivers with a version number smaller than V3.x.x are compiled for one specific Linux kernel version. The compilation is executed by *esd* in consultation with the customer. Adjustments to kernel changes are only possible by adjustment and new compilation by esd.

4.1.1 Linux Driver Installation (from V3.x.x on, except EtherCAN)

Note:

- On most Linux installations the driver installation is only possible with superuser rights.
- Please read the current `README` file that comes with the software!
- Please note the drivers delivered on the CD are most likely outdated. The increasing speed in Linux kernel development makes it almost impossible for us to provide you with drivers on CD, which work with all Linux versions and distributions. Thus we installed a download site on our website. In order to circumvent any problems before they occur, we advise you to visit this site before actually installing a driver from this CD.
  The latest driver archives can be found here: "http://www.esd-electronics.com/german/candriver"
- Please note: In order to provide the Linux CAN-drivers from our website, we needed to change the archive format to encrypted ZIP.
  The password can be received from our CAN-CD or via email from our support.

4.1.1.1 Files of the Linux Package

The software drivers for Linux are distributed on CD-ROM or delivered as archive via e-mail. The following files are contained:

File	Description
`README`	current notes and information
`Makefile`	driver compilation rules
`config.mk`	configuration file for the compile process It may be necessary to edit this file, in order to suit any peculiarities of the current system (mainly the *KERNELPATH*, see page 64)
`libntcan.a`	static CAN-API library (located in directory `./lib` *))
`libntcan.so`	dynamic CAN-API library (located in directory `./lib` *))
`ntcan.h`	header of the NTCAN-API/Library (located in directory `./lib` *)) This is the only header that has to be include in the application. Please do not use any defines located in any of the other headers, in order to keep your applications working with future version of the driver!
`cantest.c`	source code of the example-application 'cantest' (located in subdirectory `./example`) (see CAN-API manual part 1 [1])
`cantest`	binary of example-program 'cantest' (located in subdirectory `./bin` *))

File	Description
xxxx.o xxxx.c xxxx.h	source- and object-files (located in subdirectory `./src`) This driver is released as a combination of binary-objects (*.o) and source-files (`*.c` and `*.h`). This way esd can provide a CAN-driver working with many different Linux-kernels. The source files are NOT under the GPL (GNU Public Licence)! You are not allowed to modify, redistribute or sell the files! They are intellectual property of esd-electronics gmbh. **BEWARE:** Do not try to use any defines or data-structures located in these files in your own sources. This will lead to non-working applications in the future.
upd*crd*	This tool is only delivered with CAN modules that are equipped with a local processor (e.g. CAN-PCI/331). (located in subdirectory `./bin` *)) This tool can be used to switch the firmware of such a card between CAN 2.0A-firmware (used for reception of CAN-messages with 11-bit-identifier) and CAN-2.0B-firmware (used for additional reception of CAN-messages with 29-bit identifier). Syntax: upd*crd* -t*x* *net* Parameter: *crd*: CAN module ID, e.g. `pci331`, `usb331` (see table on page 62) *x*: '`a`', if CAN 2.0A firmware '`b`', if CAN 2.0B firmware *net*: Net number of the CAN interface in the system (0, 1, 2, ...)

*) **Note:** In driver archives for x86_64-Linux the path for libraries and binaries exists twice: Once for 32-bit (`./lib32` and `./bin32`) and once for 64-bit (`./lib64` and `./bin64`).

Table 11: Files of the Linux package

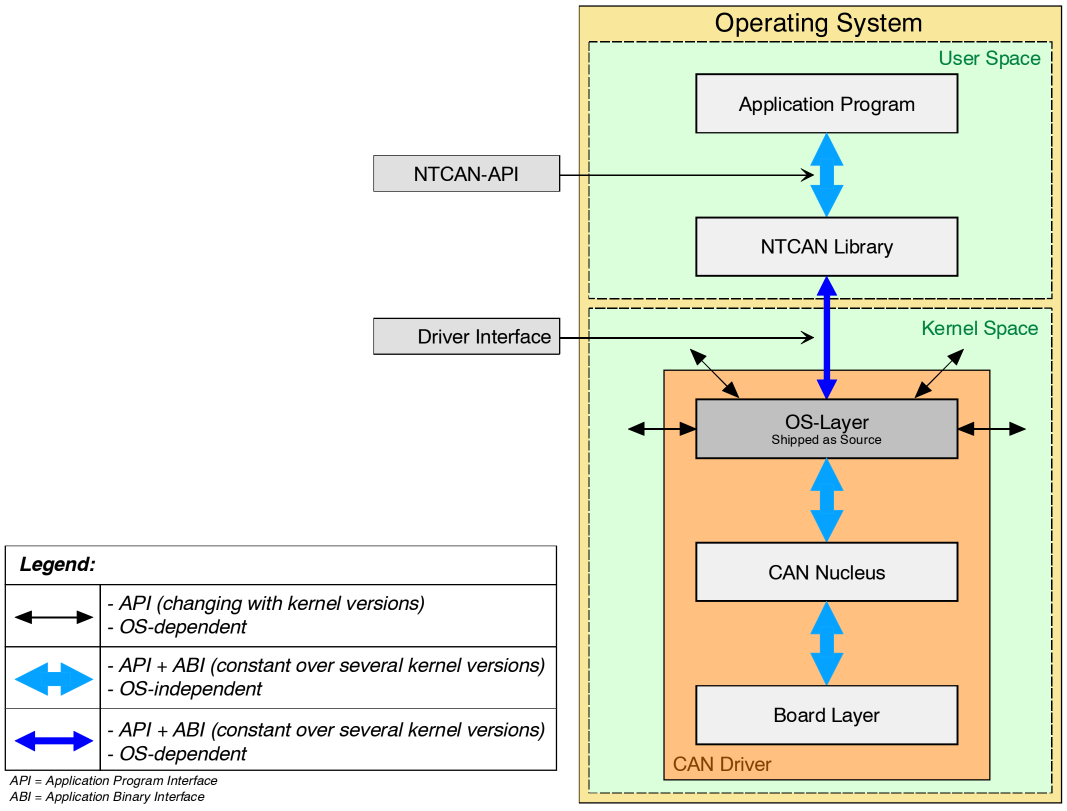

Operating System
User Space
Application Program
NTCAN-API
NTCAN Library
Driver Interface
Kernel Space
OS-Layer
Shipped as Source
CAN Nucleus
Board Layer
CAN Driver
Legend:
- API (changing with kernel versions)
- OS-dependent
- API + ABI (constant over several kernel versions)
- OS-independent
- API + ABI (constant over several kernel versions)
- OS-dependent
API = Application Program Interface
ABI = Application Binary Interface

Fig. 1: Linux driver architecture

4.1.1.2 CAN-Module-ID and Default Parameters of the Driver

CAN Module	Module ID `crd`	Default Values *)		
		`major`	Address `io`	Interrupt `irq`
CAN-ISA/200 CAN-PC104/200 (SJA1000 version)	`isa200`	`53`	`0x1E8`	`7`
CAN-ISA/331 CAN-PC104/331	`isa331`	`52`	`0x1E0`	`5`
CAN-PCI104/200 CAN-PCI/200 CAN-PCIe/200 CPCI-CAN/200 CAN-PCI/266 PMC-CAN/266	`pci200`	`54`	-	-
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`pci331`	`50`	-	-
CAN-PCI/360 CPCI-CAN/360	`pci360`	`51`	-	-
CAN-USB-Mini	`usb331`	`50`	-	-
CAN-USB/2	`usb2292`	`50`	-	-
CAN-PCI/405	`pci405`	`53`	-	-
CPCI-405 (local driver)	`cpci405`	`53`	-	-
CPCI-CPU/750 (local driver)	`cpci750`	`53`	-	-
VME-CAN2	-	-	-	-
VME-CAN4	-	-	-	-

*) The default values can be overwritten by the command `insmod` (see following chapter).

Table 12: CAN-module ID and default parameter

4.1.1.3 Installation

Note: It is absolutely necessary to install the kernel sources and configure them to comply with the running kernel, **before** installing the CAN-driver!

1. Unpacking the Archive

In order to unpack the tar-archive the file has to be unzipped and untarred:

1.1.1 Unzip the Archive

```
unzip esdcan-crd-os-arch-ver-ext.zip
```

with

crd	=	card-id (e.g.: *crd*=pci200 or *crd* =cpci405, see table on page 62)
os	=	host-operating-system (e.g.: *os* =linux_2_4_x)
arch	=	host-architecture (e.g.: *arch* =x86 or *arch* =x86_64)
ver	=	driver version (e.g.: *ver* =3.7.2)
ext	=	extension (applicable to certain cards only, e.g.: *ext*=gcc2)

You will be prompted for a password.
The password can be found in the accompanying file README_FIRST (delivered on the CAN-CD or per e-mail).

Resulting file: `esdcan-crd-os-arch-ver-ext.tar`

1.1.2 Untar the Driver Directory

```
tar -xvf esdcan-crd-os-arch-ver-ext.tar
```

with

crd	=	card-id (e.g.: *crd*=pci200 or *crd*=cpci405, see table on page 62)
os	=	host-operating-system (e.g.: *os*=linux_2_4_x)
arch	=	host-architecture (e.g.: *arch*=x86 or *arch*=x86_64)
ver	=	driver version (e.g.: *ver*=3.7.2)
ext	=	extension (applicable to certain cards only, e.g.: *ext*=gcc2)

You will end up with a directory named as the archive.

1.2 Unzip and Untar in a Single Step

Alternatively the unzipping and untaring of the driver directory can be accomplished in a single step with:

```
unzip -p esdcan-crd-os-arch-ver-ext.zip | tar -xv
```

The unpacked files will be stored in a directory that carries the same name as the archive file.

2. Compiling the Driver

```
cd ./esdcan-crd-os-arch-ver-ext
```

In some cases you need to edit a configuration file for the compilation:
In `config.mk` you need to set the variable *`KERNELPATH`* correctly. Normally the default path should be correct. If your Linux configuration differs from the standard, correct the line accordingly:

```
KERNELPATH = <your-path-to-the-kernel-source>
```

Note for Linux kernel > 2.6.0:	On some systems you might need to be "root" to compile the driver.

Compilation of the driver is simply started by typing:

```
make
```

For some cards there are warnings like `COMPILING FOR xxx`. These can be ignored and will be removed in future versions.

Now, you have a file called as described below, which is the actual driver-module in the same directory:

```
esdcan-crd-os-arch-kver
```

Dynamic loadable driver-file with

crd	=	module ID (e.g.: *crd* = `pci200` or *crd*=`cpci405`; see table on page 62)
os	=	host operating system (e.g.: *os* = `linux`; ...)
arch	=	host architecture (e.g.: *arch* = `x86`)
kver	=	target-version information ((e.g.: *kver*: `2.4.18`) For Linux the kernel version is coded here, because the compiled version is kernel specific!)

Example:
For a CAN-PCI/331 for 32-Bit-x86-Linux with 2.4.21-99-smp kernel the driver file (driver version 3.6.1) is called:

```
esdcan-pci331-linux-x86-3.6.1-2.4.21-99-smp
```

Note for Linux kernel > 2.6.0:	The driver file is called `esdcan-`*`crd`*`.ko` and is generated inside of the `src`- subdirectory.

Example (for Linux kernel > 2.6.0**):**
For the above mentioned CAN-PCI/331 the driver file is called:

`esdcan-pci331.ko` (the file is located in `./src` subdirectory)

3. File Locations

It is recommended to store the driver module in the following directory:

`/lib/modules/`*`kernel-version`*`/`

The variable *`kernelversion`* has to be replaced by the according string of the system. The string (it should be equivalent to the *`os`*-string in the driver's name (see above)) is returned, if the following command is called:

```
uname -r
```

The dynamic shared library `libntcan.so` should be placed in the directory `/usr/local/lib/` or an equivalent path, which is contained in the `LD_LIBRARY_PATH` env-variable.

Note: On 64-bit systems, there are two versions of `libntcan.so`. One in `./lib32` and one in `./lib64`. The first belongs into `/usr/local/lib` on most Linux distributions. The later should be kept together with other 64-bit libraries, e.g. in `/usr/local/lib64`.

The static version of the library `libntcan.a` can be kept wherever you want. Here at **esd** we prefer to keep it with the sources of a project, on the other hand, one might like to install it with the shared-lib at `/usr/local/lib/` .

Installation-note:

The shared library should belong to user and group “root” with the following file access permissions: `u=rwx, g=rx, o=rx`

After installation of the library, the root-user should call:

`ldconfig -n /usr/local/lib` (if installed to this directory)

Afterwards there is a link `libntcan.so.v --> libntcan.so.v.mv.r` .
For your own convenience it is advised to generate another link in your library-directory:

```
libntcan.so --> libntcan.so.v
```

The static-library, if installed in `/usr/local/lib/`, should also belong to user/group root, but it does not need (and should not have) the executable-flag.
Leading to the following file access permissions: `u=rw, g=r, o=r`

4. Make the 'inodes' (as Superuser)

```
cd /dev
mknod --mode=a+rw can0 c xx 0
mknod --mode=a+rw can1 c xx 1   ==>  as many as physical CAN nets provided
                                     by the modules
```

with
xx = major number of the driver (see table on page 62)

5. Load the Driver File (as Superuser)

Syntax:
`insmod ./esdcan-`*`crd-os-arch-kver`*` [major=`*`m`*`]`

with the following optional parameter: *m* = non-default major

The naming of the kernel module is equivalent to the naming of the driver archive as printed on page 62 (exception: kernel 2.6.x).

Note: With the module CAN-PCI/405 this call returns after approx. 5 seconds!

6. Checking the Installation

Whether the installation has been successful or not, can be checked in the following file:

`/var/log/messages`

```
kernel: esd CAN driver: pci405
kernel: esd CAN driver: baudrate not set
kernel: esd CAN driver: mode=0x00000000, major=53, 4 nodes on 1 cards
kernel: esd CAN driver: version 0.3.1 14:36:35 Feb 13 2003: successfully
loaded
```

After a successful installation, the CAN bus can be accessed by means of the NTCAN-API. The application has to be linked to the library `libntcan.a` (static) or `libntcan.so` (shared).

Note: **esd** strongly recommends usage of the shared library!

If the example application `cantest` is called, the available CAN nets are displayed.

7. Unload the Driver File (as Superuser)

Kernel 2.4.x:
`rmmod esdcan-`*`crd-os-arch-kver`*

Kernel 2.6.x:
`rmmod esdcan-`*`crd.ko`*

4.1.2 Linux Driver Installation (drivers before version V.3.x.x. only)

Note:

- The installation of drivers with version numbers ≥V3.x.x is described from page 59 on.
- The installation is only possible with superuser rights.
- The 'object mode' of the CAN-API is not supported by Linux CAN drivers before version V3.x.x!
- If the CAN-USB-Mini module is used with the Linux driver with version number < 3.x.x, it is not allowed to remove it from the USB bus while a handle is opened! It is recommended to update such a driver.

4.1.2.1 Files of the Linux Package

The software drivers for Linux are distributed on CD-ROM or delivered as archive via e-mail. The following files are contained:

File	Description	
`README.ican4`	current notes and information	The letter combination '*`ican4`*' signifies the files of the module 'VME-CAN4. For other modules these letters are changed as follows:
e.g. `ican4`	dynamically loadable driver	
`updican4`	program for firmware update	
`libntcan.a`	library with ntcan-API for static linking	
`ntcan.h`	header for ntcan-API	
`cantest.c`	source code of example program 'cantest' (see CAN-API manual part 1 [1])	
`cantest`	binary file of example program 'cantest'	

CAN Module	File Name
VME-CAN2	`ican2`
VME-CAN4	`ican4`

Table 13: Files of the Linux package

4.1.2.2 Sequence of Installation

1. **Unpacking the Archive**
In order to unpack the tar-archive file or the tar-archive disk the following command has to be called:

```
tar -xvf ican4-vx.y.z.tar
```
or
```
tar -xvf /dev/fd0
```

Note: '*ican4*' has to be entered for the VME-CAN4 module. For other modules the character combination shown in the following table has to be entered. The same applies to the following commands.

CAN Module	Entry Syntax
VME-CAN2	ican2
VME-CAN4	ican4

Table 14: Input syntax for unpack the archive

2. **Generating the 'inodes' (as Superuser)**

For VME-CAN2:
```
    cd /dev
    mknod --mode=a+rw can0 c 50 0
    mknod --mode=a+rw can1 c 50 1
```

For VME-CAN4 :
```
    cd /dev
    mknod --mode=a+rw can0 c 50 0
    mknod --mode=a+rw can1 c 50 1
    mknod --mode=a+rw can2 c 50 2
    mknod --mode=a+rw can3 c 50 3
```

3. **Loading the CAN Driver (as Superuser)**

Example for VME-CAN4 (for other boards see table above):

```
insmod ican4
```

(3A. **Unloading the CAN Driver**)
The CAN driver has to be unloaded, e.g., after an update of the local firmware in order to reset the processor of the CAN module. The CAN driver of the VME-CAN4, e.g. can be unloaded by the following command:

```
rmmod ican4
```

4. **Checking the Installation**
   Whether the installation has been successful or not, can be checked in the following file:

```
/var/log/messages
```

After a successful installation, the CAN bus can be accessed by means of the NTCAN-API (integration of '*libntcan.o*' into the application).

4.1.3 Linux Driver Installation of the EtherCAN-Module

Note:

- The installation is only possible with superuser rights (user: root).

- Please read the current `README` file that comes with the software!

4.1.3.1 Files of the Linux Package

The software drivers for Linux are distributed on a CD-ROM, which contains the following files:

File	Description
`README.x.x`	current notes and information
`cantest.c`	source code of the example-application 'cantest' (located in subdirectory `./example`) (see CAN-API manual part 1 [1])
`cantest`	binary of example-program 'cantest' (located in subdirectory `./bin`)
`makefile.ethercan`	example makefile for compiling the file `cantest.c`
`installEthercanLibs`	bash script for installation of libraries and include files
`etc/esd-plugin`	example config file for `ntcanEthPlugin` (to be stored as `/etc/esd-plugin`)

`pdf/CAN-API_Part1_ Function_e.pdf`
 	
`pdf/CAN-API_Part2_ Installation_e.pdf`
	documentation of esd-NTCAN-API (this manual)
	Part 1: Function Description,
	Part 2: Installation Guide [1]

`pdf/psys.pdf`	documentation of esd system abstraction layer API
`include/ntcan.h`	header of the ntcan-API/Library (to be stored e.g. at `/usr/local/include`) This is the only header that has to be include in the application. Please do not use any defines located in any of the other headers, in order to keep your applications working with future version of the driver!
`lib/libntcan.so.x.x.x`	shared library containing the NTCAN-API (to be stored e.g. at `/usr/local/lib`)
`lib/ntcanEthPlugin.so.x.x.x`	dynamically loadable plugin for `libntcan.a` or `libntcan.so` (to be stored at e.g. `/usr/local/lib`)

File	Description
`psys_linux/include/psys.h` `psys_linux/include/psyslinux.h`	Psys header files, esd System Abstraction Layer API (to be stored e.g. at. `/usr/local/include`)
`psys_linux/lib32/libpsys.so.x.y.z`	32-bit shared library, containing esd System Abstraction Layer (to be stored at e.g. under `/usr/local/lib`)
`psys_linux/lib64/libpsys.so.x.y.z`	64-bit shared library, containing esd System Abstraction Layer (to be stored e.g. at /usr/local/lib64)
`psys_linux/src/psysdrv.c`	source of the psys-driver
`psys_linux/src/psysdrv.h`	Psys-driver header
`psys_linux/src/Makefile`	KBuild-Makefile needed for the module generation with kernel 2.6.x
`psys_linux/Makefie`	makefile for PSYS-driver for kernel 2.4.x or 2.6.x
`psys_linux/README`	release notes and installation hints for psys / psys-driver
`psys_linux/LICENSE`	license covering PSYS driver and library

Table 15: Files of the EtherCAN Linux package

4.1.3.2 Installation

1. **Unpacking the Archive**
   In order to unpack the zipped tar-archive file the following command has to be called:

   ```
   tar xvfz ethercan-lx-2.0.10-ntcan-3.0.6- psys-1.3.0-gcc-3.3.1-glibc-2.3.2.tgz
   ```

   The unpacked files will be stored in a directory that carries the same name as the archive file.

2. **Compile and Load Psys-Driver**
   Therefore follow the installation instructions in *psys_linux/README*

3. **Installing the Libraries and Header Files**
   In order to install the libraries and header files the following command has to be called (with supervisor rights):

   ```
   ./installEthercanLibs
   ```

   All library files will be stored in `/usr/local/lib`
   and all include files will be stored in `/usr/local/include`.

4. **Adapt `/etc/esd-plugin`**
   See chapter “Configuration” on page 74

5. **Compiling the Example Program ‘cantest’**
   With the call

   ```
   make -f makefile.ethercan
   ```

   the test- and example program `cantest` will be compiled.

After successful installation you can access the CAN-Bus via the esd NTCAN API (link `libntcan` with your application). For an API documentation see the standard esd NTCAN API (`ntcan.pdf`) documentation.

4.1.3.3 Configuration

All user configurable stuff concerning EtherCAN can be found in the file `/etc/esd-plugin`.

List of available keywords in `/etc/esd-plugin` (with 0 < x < 4):

Keyword	Description	Default Value
`PeerName[x]`	host name or IP-address of EtherCAN server	-
`Net[x]`	CAN net number assigned to EtherCAN server with above `PeerName[x]`	50 + x
`ConnTimeout[x]`	Time to wait until connection to the EtherCAN server is established. If timeout exceeds, *canOpen()* returns NTCAN_SOCK_CONN_TIMEOUT.	2500 ms
`CmdTimeout[x]`	Timeout for special commands send from client to EtherCAN-server.	2500 ms
`KeepAliveTime[x]`	If there is no CAN traffic, client sends a keep-alive message to server. If sending the keep-alive message fails, the connection to the server is disconnected and the EtherCAN client will try to reconnect the server.	2500 ms
`TCPNoDelay[x]`	0: Nagle algorithm active (i.e.: Coalesce a number of TCP messages and send them all at once) 1: Nagle algorithm off (i.e.: Immediately send TCP messages without any inhibit)	1

Table 16: Configuration keywords

Example 1: EtherCAN configured as CAN net 30 and all other parameters in default setting.

```
PeerName[1]=       "10.0.16.58"
Net[1]=            30
```

Example 2: EtherCAN configured as CAN net 20 and, with increased timeouts, because it is located outside the company network.

```
PeerName[0]=       "134.66.177.1"
Net[0]=            20
KeepAliveTime[0]=  10000 # increase keep-alive timeout
ConnTimeout[0]=    25000 # increase connection timeout
CmdTimeout[0]=     5000  # increase command timeout
```

4.1.3.4 Linking Against `libntcan` (gcc-Option `-rdynamic`)

Attention! It is mandatory to use the option `-rdynamic` when linking against `libntcan`, because the dynamically loaded library `ntcanEthPlugin.so` (beside delivering some new functionality) itself needs some symbols from within `libntcan`. Without `-rdynamic` this does not work!

If your application (on runtime) complains about '`ntcanEthPlugin.so: undefined symbol: openRegistryCanIf Ether`' the option `-rdynamic` is still missing in your makefile.

4.1.3.5 Special NTCAN_XXX Return Codes, not yet Documented in `ntcan.pdf`

Currently all NTCAN_XXX return codes should be available in the NTCAN documentation. See also ntcan.h.

For return values less then `NTCAN_ERRNO_BASE` (0x100) , you have to look into `errno.h` provided by your system (typically this can be found somewhere under `/usr/src/linux/include/asm-xxx` ...)

4.2 LynxOS Installation

Note: The ‘object mode’ of the CAN-API is not supported by LynxOS.

4.2.1 Files of the LynxOS Package

The software drivers for LynxOS are contained on a 3.5”-disk with the following files:

File	Description	
`instcan`	script for loading and unloading the driver	
`README.c331`	current information and notes	The character combination ‘*`c331`*’ represents the files of module CAN-PCI/331. For other modules this combination is changed as shown below:
`c331`	dynamically loadable driver	
`c331.info`	parameter file for the driver (is read at installation and deinstallation)	
`c331dbg`	debug version of the dynamically loadable driver (should only be used, if problems arise during installation)	
`ntcan.o`	ntcan-API	
`ntcan.h`	header for the ntcan-API	
`canupd`	program for firmware update	
`cantest.c`	source code of the example program ‘cantest’ (see CAN-API manual part 1 [1])	
`cantest`	binary file of the example program ‘cantest’	

CAN Module	File name
VME-CAN2	`ican2`
VME-CAN4	`ican4`
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`

Table 17: Files of the LynxOS package

4.2.2 Sequence of Installation Under LynxOS

1. **Unpacking the Archive**
   In order to unpack the tar-archive the following command has to be called:

```
tar -xvf c331-lynx-v1.0.0.tar
```

Note: '`c331`' has to be entered for the CAN-PCI/331 module. For other modules the character combination shown in the following table has to be entered:

CAN Module	Entry
VME-CAN2	`ican2`
VME-CAN4	`ican4`
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`

Table 18: Input syntax for unpack the archive

2. **Loading the CAN Driver:**

```
instcan c331
```

(2.B Unloading the CAN Driver:)
The CAN driver has to be unloaded, e.g., after an update of the local firmware in order to reset the processor. In the CAN-PCI/331 module, e.g., the driver can be unloaded by the following command:

```
instcan -u c331
```

4.3 PowerMAX OS Installation

Note: The ‘object mode’ of the CAN-API is not supported by PowerMAX OS.

4.3.1 Files of the PowerMAX OS Package

The software drivers for PowerMAX OS are on a 3.5”-disk, which contains the following files:

File	Description	
`README.ican4`	current notes and information	The letter combination ‘*`ican4`*’ signifies the files of the module VME-CAN4. For other modules these letters are changed as follows:
`e.g.` `ican4`	dynamically loadable driver	
`libntcan.o`	ntcan-API	
`ntcan.h`	header for ntcan-API	
`cantest.c`	source code of example program ‘cantest’ (see CAN-API manual part 1 [1])	
`cantest`	binary file of example program ‘cantest’	

CAN Module	File Name
VME-CAN2	`ican2`
VME-CAN4	`ican4`

Table 19: Files of the PowerMAX OS package

4.3.2 Sequence of Installation Under PowerMAX OS

1. **Copy the tar-Archive `vmecan4_v1.2.tar` in the Home Directory**

2. **Unpacking the Archive**
   In order to unpack the tar-archive the following command has to be called:

   ```
   tar xvf vmecan4_v1.2/ican4
   ```

   Note: '*`ican4`*' has to be entered for the VME-CAN4 module. For other modules the character combination shown in the following table has to be entered. The same applies to the following commands.

CAN Module	Entry Syntax
VME-CAN2	`ican2`
VME-CAN4	`ican4`

Table 20: Input syntax for unpack the archive

3. **Generating the 'inodes' (as Superuser)**

   ```
   cd 4_2/vmecan4_v1.2/ican4/
   ```
   or
   ```
   cd 4_3/vmecan4_v1.2/ican4/
   ```

4. **Only at the First Installation: Add Adapter Definition**
   At the first installation of the CAN modules the following lines hat to be added to the file `/usr/include/sys/adapt3er_vme.h/` :

   ```
    #define ADAPTER_ICAN2   (AVB+0x12)   /* 0x212 - esd ican2 */
    #define ADAPTER_ICAN4   (AVB+0x13)   /* 0x213 - esd ican4 */
   ```

5. **Only if an Existing Driver is Updated:**
   If an existing driver should be updated, the following line is necessary:
   ```
   modadmin -U ican4
   ```

6. **Installation**
   - in case of first installation:
   ```
   ./install -f
   ```

   with following reboot

   - in case of an update:
   ```
   ./install -u
   ```

7. **Change Directory:**
   change to
   `~/4_2/vmecan4_v1.2` for PowerMAX OS 4.2
   or
   `~/4_3/vmecan4_v1.2` for PowerMAX OS 4.3

8. **Start Driver:**
   After calling
   ```
   modadmin -l ican4
   ```
   the driver displays his start message.

9. **Starting `cantest`:**
   After starting `cantest` with
   ```
   ./cantest
   ```
   the program shows four accessible CAN nets (net 0...3)

After a successful installation, the CAN bus can be accessed by means of the NTCAN-API (integration of ‘*libntcan.o*’ into the application).

4.4 Solaris Installation

Note: The ‘object mode’ of the CAN-API is not supported by Solaris.

4.4.1 Files of the Solaris Package

The software drivers for Solaris are contained on a 3.5”-disk. This disk contains the following files:

File	Description	
`install`	script for installation of driver	
`README.c331`	current notes and information	The character combination ‘`c331`’ signifies the files of the CAN-PCI/331 module. For other modules the characters change as follows:
`c331`	dynamically loadable driver	
`c331.conf`	parameter file for driver (is read at installation and deinstallation)	
`ntcan.o`	ntcan-API	
`ntcan.h`	header for ntcan-API	
`canupd`	program for firmware update	
`cantest.c`	source code of example program ‘cantest’ (see CAN-API manual part 1 [1])	
`cantest`	binary file of example program ‘cantest’	

CAN Module	File Name
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`
VME-CAN4	`ican4`

Table 21: Files of the Solaris package

4.4.2 Sequence of Installation Under Solaris

1. **Unpacking the Archive**

   In order to unpack the tar-archive you have to call the following command:

```
tar -xvf c331-solaris-vx.x.x.tar
```

   (x.x.x = driver version number)

   Note: The entry '*c331*' has to be made for the CAN-PCI/331 module. Please specify the corresponding letter combination shown in the table below for other modules:

CAN Module	Entry Syntax
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`
VME-CAN4	`ican4`

   Table 22: Entry syntax when unpacking the archive

2. **Preparing the System to Dynamically Create 'inodes':**

   To make sure that during the driver installation 'inodes' are automatically created under the `/dev` directory, it is necessary to conform the file `/etc/devlink.tab`. This has to be made with superuser rights. It is recommendable to make a backup of this file before starting. The following line has to be added to the file:

```
type=can;   \M0
```

   When installing the driver for the first time, the files `/dev/can`*`x`* should be automatically created now, *x* indicating the (hexadecimal) network number.

3. Conforming the Configuration File

3.1 Conforming the Configuration File for CAN-ISA Modules:

Before the driver is installed for the first time, its driver configuration file has to be conformed to the hardware configuration. The following properties have to be available in `driver.conf`, where `driver` has to be substituted by the according driver name. Only the properties printed in italics have to be conformed. For a summarizing overview of the (bus-specific) properties) please consult the according manual pages (`driver.conf, sysbus, pci`).

Name	Parameter	Meaning
`name`	String	Driver name
`class`	String	Bus type
`interrupts`	Numeric	Interrupt vector
`interrupt-priorities`	Numeric	Interrupt priority level (IPL)
`reg`	Numeric	Three values, separated by commas. The second corresponds to the CAN interface basis address and the third describes the size of the I/O area in bytes.

Table 23: Properties of the driver configuration file

The value given in *`reg`* has to correspond to the basis address configured in the hardware by means of jumpers or coding switches. The IPL level has to be assigned to a high-level interrupt and the interrupt vector must not have been assigned by another hardware component.

Below an example configuration file for a CAN-ISA/331 is shown. Its basis address has been configured at 0x1E0 and it is to use the interrupt vector 7 with an IPL of 11:

```
# Copyright (c) 1998, by esd gmbh.
#
name="isa331" class="sysbus" interrupts=7 interrupt-priorities=11 reg=1,0x1e0,8;
```

3.2 Conforming the Configuration File for CAN-PCI Modules:

Since PCI devices report themselves, the configuration and assignment of resource is automatically when the system is started (Plug & Play). The device driver automatically adopts the assigned resource so that it does not have to be manually assigned in the driver configuration file. The only parameter which can be set is the interrupt priority to be used by the IPL driver (see also example `c331.conf`).

```
# Copyright (c) 1999-2000 electronic system design gmbh
#

# do not remove the next line
interrupt-priorities=9;
```

3.1 Conforming the Configuration File for the VME-CAN4 Module:

Before the driver is installed for the first time, its driver configuration file has to be conformed to the hardware configuration. The following properties have to be available in `driver.conf`, where `driver` has to be substituted by the according driver name. Only the properties printed in italics have to be conformed. For a summarizing overview of the (bus-specific) properties) please consult the according manual pages (`driver.conf, sysbus, pci`).

Name	Parameter	Meaning
`name`	String	Driver name, here always: `ican4`
`class`	String	Bus type here always: `vme`
`interrupts`	Numeric	Interrupt level and vector Four interrupts at the same interrupt level with interrupt vectors with the offset of 1, 4 and 5 have to be defined (see following example). **Note:** The hardware of the VME-CAN4 supports up to 8 interrupt vectors. Because of the compatibility to the VME-CAN2 the vectors 1, 2, 5 and 6 are used (according entries: 0x80, 0x81, 0x84, 0x85). **Attention:** The interrupt level has to carefully selected to obviate conflicts with existing interrupts of the system!
`reg`	Numeric	Contains six values separated by commas, that define the address range of the VME-CAN4: - The first value defines the 1. address range: `0xad` => A16 - The second value defines the board address within the 1. address range that is used to initialize the address registers of the VME-CAN4: `0xe000 + (`*`coding_switch_setting`*` · 0x100)` (only, if the geographical addressing is inactive, please refer hardware manual of VME-CAN4) - The third value defines the size of the 1. address range in bytes and thus the offset to the next board within the A16-address range of the system: always `0x100` - The fourth value defines the 2. address range: `0x4d` => A32 - The fifth value defines the A32-address of the VME-CAN4 board: e.g. `0x10000000` - The sixth value defines the size of the 2. address range in bytes: always `0x00100000` **Note:** Due to reasons of the address coding the offset to the next board within the A32-address range in the system is `0x00200000`!

Table 24: Properties of the driver configuration file

Below an example configuration file for the VME-CAN4:

```
name="ican4"
class="vme"
interrupts=6,0x80,6,0x81,6,0x84,6,0x85
reg=0xad,0xe100,0x100,0x4d,0x10000000,0x00100000;
```

4. Installing the CAN Driver

In order to install the driver the script `install` has to be executed as superuser. It copies the driver files into the target directory, installs and starts the driver. From now on the driver will be automatically loaded and started with every system start.

After it has been successfully installed, you can access the CAN bus via the NTCAN-API (including '*ntcan.o*' into the application).

5. Checking the Installation

5.1 For CAN-PCI Modules Only:

If the installation script has been executed correctly, the driver is started and automatically loaded with every system start.
By entering "`modinfo | grep d3x`" into the root you can check whether the driver is loaded or not. An output similar to the following has to appear:

```
205 f5d11000   34d9 132   1  c331 (CAN-PCI/331 driver v1.3.3)
```

5.2 For All Modules:

You can check whether the installation was successful by reading the driver boot message in the following file:

```
/var/adm/messages
```

6. Unloading the CAN driver

Unloading the CAN driver is. For example, necessary after an update of the local firmware for resetting the processor. For the CAN-ISA/331 module this can be achieved by the following command, which has to be executed as a superuser:

```
rem_drv c331i
```

4.5 SGI-IRIX6.5 Installation

Note: The ‘object mode’ of the CAN-API is not supported by SGI-IRIX6.5 .

4.5.1 Files of the SGI-IRIX6.5-Package

The software drivers for SGI-IRIX6.5 are shipped on a 3.5” disk. The disk contains the following files:

File	Description	
`c331`	CAN driver (object code)	The letter combination ‘*c331*’ indicates the files of module CAN-PCI/331. For other modules these letters will be substituted as follows:
`c331.master`	driver configuration file	
`c331.sm`	driver configuration file	
`makefile`	makefile for installing and loading the driver	
`ntcan.o`	ntcan-API	
`ntcan64.o`	ntcan-API / 64-bit version	
`ntcan.h`	Header for the ntcan-API	
`cantest.c`	source code of the example program ‘cantest’ (see CAN-API manual part 1 [1])	
`cantest`	binary file of the example program ‘cantest’	

CAN Module	File Name
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`
CAN-PCI/405	`pci405`

Table 25: Files of the SGI-IRIX6.5 package

4.5.2 Sequence of Installation Under SGI-IRIX6.5

1. **Login as Root**

2. **Unpacking the Archive**
   In order to unpack the tar-archive of the CAN modules CAN-PCI/331, CPCI-CAN/331 and PMC-CAN/331 you have to call the following command:

```
tar -xvf c331-IPyy-vx.x.x.tar
```

with

x.x.x	= software driver version (e.g. 2.1.0)
yy	= processor identification (e.g. '32' for SGI-O2)

In order to unpack the tar-archive of the CAN module CAN-PCI/405 you have to call the following command:

```
tar -xvf esdcan-pci405-irix-mips-x.x.x-IPyy-z.z
```

with

x.x.x	= software driver version (e.g. 0.3.2)
yy	= processor identification (e.g. '27')
z.z	= operating system version (e.g. '6.5')

3. **Change Directory (e.g. CAN-PCI/331, CPCI-CAN/331 and PMC-CAN/331)**

```
cd c331-IPXX-v2.1.0
```

4. **Install Driver Data**

```
smake install
```

5. **Load Driver**

`smake load`	If the driver has been installed and loaded correctly, a message of the driver has to appear on the screen now.

4.6 AIX Installation

4.6.1 Special Features of the AIX Implementation

- the CAN driver has been designed for operating system version AIX 4.2.1
- hardware platform: PowerPC
- implemented esd-CAN modules: CAN-PCI/331, CPCI-CAN/331
- 29-bit identifiers are not yet being supported
- the ‘object mode’ of the CAN-API is not supported by AIX

4.6.2 Files of the AIX Package

The software drivers for AIX are contained on a 3.5” disk. The disk contains a tar-archive.

File	Description	
`README.c331`	current notes and information	The character combination ‘`c331`’ signifies the files of the following modules:
`c331`	dynamically loadable driver	
`ntcan.o`	ntcan-API	
`ntcan.h`	header for ntcan-API	
`nttest.c`	source code of example program ‘cantest’	

CAN Module	File Name
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`

Table 26: Files of AIX package

4.6.3 Installation Sequence under AIX

1. **Login as Root**

2. **Unpacking the Archive**
   In order to unpack the tar-archive the following command has to be called:

   ```
   tar -xvf c331-ppc-vx.y.z.tar
   ```

   (x.y.z = driver version number)

3. **Copying the Files**
   At unpacking the directory `/c331-ppc-vx.y.z` is created.
   - Copy the files `cfg_pci331` and `ucfg_pci331` from this directory into the directory `/usr/lib/methods/`.
   - Copy from this directory the file `c331` into the directory `/usr/lib/drivers/pci/`.

4. **Create Entries in the Data Base**

   ```
   run odmadd pci331
   ```

5. **Call the Configuration Manager**

   ```
   cfgmgr -v
   ```

6. **Check the Files and Create Symbolical Links**

   Check whether the devices `c33100` and `c33101` are entered in directory `/dev/`. Create a symbolical link:

   ```
   ln -s /dev/c33100 /dev/can0
   ln -s /dev/c33101 /dev/can1
   ```

   The driver is now installed.

7. Check Whether the Driver Runs With CAN tool 'CANreal'

- Make sure that the wiring and terminations are correct and make sure that at least one other working CAN participant has been connected!

- Start the monitor program CANreal:
  ```
  ./canreal &
  ```

- Set the baud rate of your CAN network for network 0 in CANreal and click *Init*.
  'Init done' appears in the window.

- Click *Add Id* to select the identifier area of 0 to 2047.

- Click *Start* to display the messages of the CAN bus.

8. Test Program `cantest`

In addition to CANreal, which has got a UNIX-typical user interface, you can also use the test program `cantest` under AIX. It is operated by means of command line specification (for details see chapter 'Test Program cantest' in CAN API manual part 1 [1]).

- compile cantest:
  ```
  gcc -o cantest nttest.c ntcan.o
  ```

- call cantest:
  ```
  ./cantest
  ```

5. Installation of Real Time OS (Non Windows)

5.1VxWorks Installation

Each VxWorks CAN driver package contains a file `relnotes.htm` in HTML format which contains the revision history of the drivers and late-breaking information which did not make into one of the manuals. Please read this file before installing the driver.

Note: The local CAN driver of the boards CPCI-405, PMC-CPU/405 and CPCI-CPU/750 is part of the BSP and is not deployed in a separate software package.

5.1.1 Notes on VxWorks

5.1.1.1 Supported Host-CPU Architecture and Operating System Versions

The CAN driver is available by esd for various CPU architectures and VxWorks operating system versions. All binaries are compiled with the GCC compiler which corresponds to the VxWorks release. Please state the host CPU architecture and the VxWorks version ordering a driver license for a specific board. The following combinations are currently supported:

VxWorks Version ->	≤ 5.4		5.5.x		6.x	
Host CPU CAN-Module	386/486	PowerPC	486/ Pentium	PowerPC	Pentium (I -IV)	PowerPC
CAN-ISA/200	☻	-	☻	-	-	-
CAN-PC104/200 (SJA1000 version)	☻	-	☻	-	-	-
CAN-ISA/331 CAN-PC104/331	☻	-	☻	-	-	-
CAN-PCI104/200 CAN-PCI/200 CAN-PCI/266 CPCI-CAN/200 PMC-CAN/266	☻	☻	☻	☻	☻	☻
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	☻	☻	☻	☻	☻	☻
CAN-PCI/360 CPCI-CAN/360	☻	-	☻	-	-	
VME-CAN4	-	☻	-	☻	-	☻

☻...	CAN-API driver available for the listed host CPU
-...	no driver available at the moment (please ask our support for the current development status)

Table 27: Supported host CPUs and VxWorks versions

5.1.1.2 Local CAN Drivers for CAN Modules that Act as Hosts Themselfs

Compared to the modules described in the previous chapter these modules come with a complete VxWorks BSP containing a local CAN driver. Using these modules as target is not supported at the moment.

CAN Module	Local Driver for CAN-API	Operation of the Module as Target in the System with external CAN Driver
PMC-CPU/405	included in BSP	in preparation for module version PMC-CPU/405-A
CPCI-405	included in BSP	in preparation for module version CPCI-405-A
CPCI-CPU/750	included in BSP	not scheduled
EPPC-405	included in BSP	n.a.
EPPC-405-HR	included in BSP	n.a.
EPPC-405-UC	included in BSP	n.a.

Table 28: CAN modules with local VxWorks operating system and local CAN driver

5.1.2 Integration of the Driver in a Project

The integration and installation of the VxWorks 6.x CAN drivers and libraries is different to VxWorks 5.x. For VxWorks 6.x the driver installation and configuration is integrated into the WindRiver Workbench. For earlier versions of VxWorks the driver and libraries are deployed as binaries which have to be linked to your application or BSP together with a configuration file which has to be adapted to your hardware configuration.

5.1.2.1 Content of the driver package for VxWorks 6.x

The driver software for VxWorks 6.x comes as a package for all supported host CPU architectures and CAN hardware with the following directory layout structure:

Directory	Description
`doc\`	Documentation of driver installation and NTCAN-API
`src\`	Example code (`cantest.c`) also included in binary format which can be optionally added to your project.
`target\`	This directory contains all necessary files to integrate and configure the driver. **For driver installation and integration with the WindRiver Workbench the complete folder has to be copied into the target directory of your VxWorks 6.x installation keeping the directory hierarchy.** The CAN driver support will be available with the next start of the Workbench.

Table 29: Directory hierarchy of the VxWorks 6.x package

5.1.2.2 Content of the driver package for VxWorks 5.x

The driver software for VxWorks 5.x comes as a CAN hardware specific package with the directory structure `/VxWorks-Version/CPU-Architecture/`. Supported VxWorks versions are 5.4 and 5.5.1. Each directory contains the following files:

File	Description	
`ldc331i`	script for loading the driver	The character combination '*c331i*' signifies files of the CAN-ISA/331 module. For other modules the characters have to be changed as follows:
`c331i.sys`	CAN driver (binary code)	
`c331iini`	example startup code for the driver (binary code)	
`c331iini.c`	example configuration and startup code for the driver (source code)	
`ntcan.o`	ntcan-API	
`ntcan.h`	header for ntcan-API	
`cantest.c`	source code of example program 'cantest' (see CAN-API manual part 1 [1])	
`cantest`	binary file of example program 'cantest'	

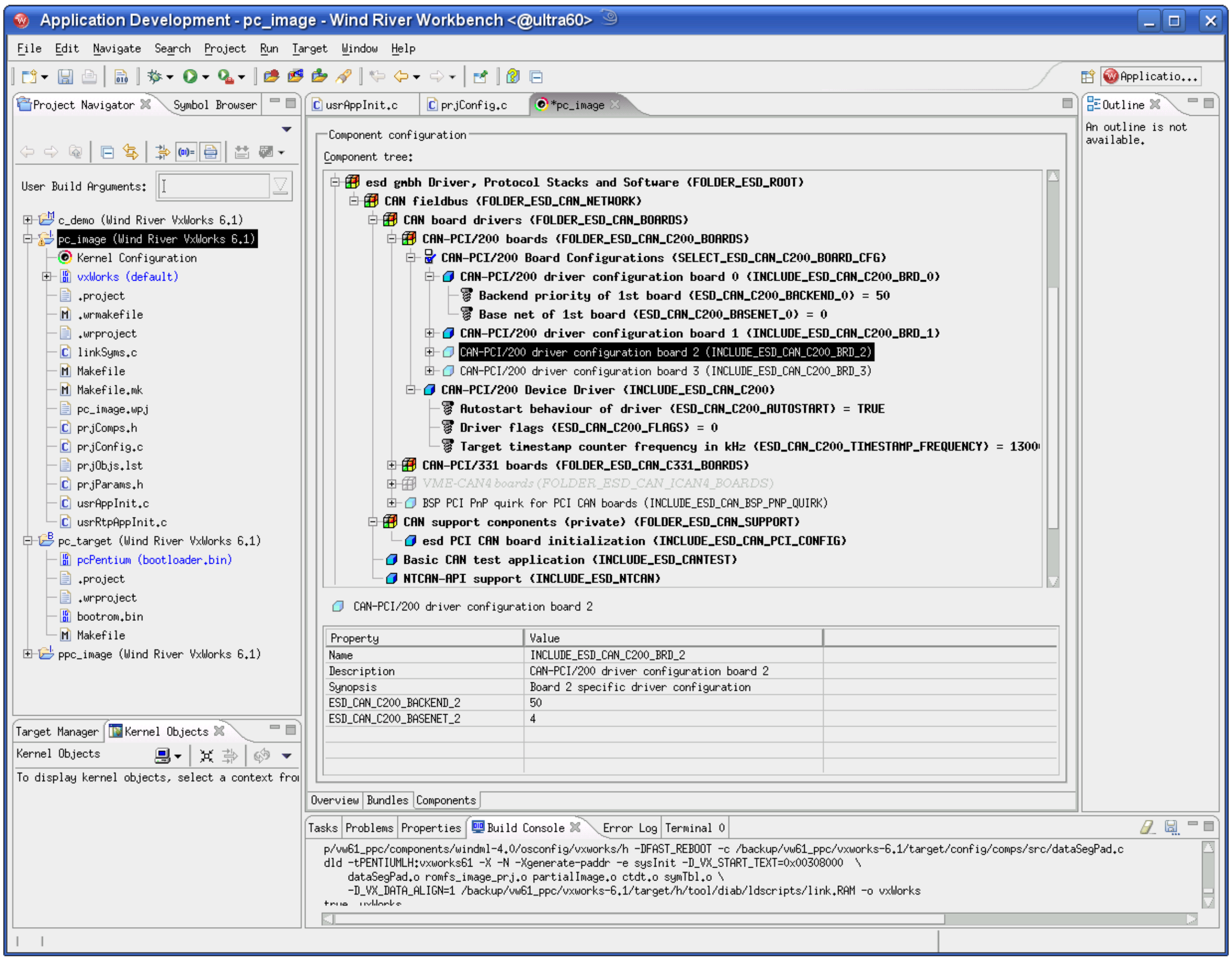
CAN Module	File Name
CAN-ISA/200 CAN-PC104/200	`c200i`
CAN-ISA/331 CAN-PC104/331 PMC-CAN/331	`c331i`
CAN-PCI/200 CPCI-CAN/200	`c200`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`
CAN-PCI/360 CPCI-CAN/360	`c360`
VME-CAN4	`can4`

Table 30: Files in the VxWorks 5.x package

5.1.3 Driver Configuration

5.1.3.1 Configuration of the VxWorks 6.x Driver

The integration of the CAN driver in your application and the configuration of the driver parameter is done with the WindRiver Workbench. After the driver installation you will have a new node in your Workbench VxWorks kernel parameter tree with the name *esd gmbh Driver, Protocol Stacks and Software*.


Application Development - pc_image - Wind River Workbench <@ultra60>
File Edit Navigate Search Project Run Target Window Help
Project Navigator
Symbol Browser
User Build Arguments:
usrAppInit.c
prjConfig.c
*pc_image
Outline
An outline is not available.
Target Manager
Kernel Objects
To display kernel objects, select a context fro
Overview
Bundles
Components
Tasks
Problems
Properties
Build Console
Error Log
Terminal 0

```
c_demo (Wind River VxWorks 6.1)
pc_image (Wind River VxWorks 6.1)
    Kernel Configuration
    vxWorks (default)
    .project
    .wrmakefile
    .wrproject
    linkSyms.c
    Makefile
    Makefile.mk
    pc_image.wpj
    prjComps.h
    prjConfig.c
    prjObjs.lst
    prjParams.h
    usrAppInit.c
    usrRtpAppInit.c
pc_target (Wind River VxWorks 6.1)
    pcPentium (bootloader.bin)
    .project
    .wrproject
    bootrom.bin
    Makefile
ppc_image (Wind River VxWorks 6.1)
```

Component configuration

Component tree:

```
esd gmbh Driver, Protocol Stacks and Software (FOLDER_ESD_ROOT)
  CAN fieldbus (FOLDER_ESD_CAN_NETWORK)
    CAN board drivers (FOLDER_ESD_CAN_BOARDS)
      CAN-PCI/200 boards (FOLDER_ESD_CAN_C200_BOARDS)
        CAN-PCI/200 Board Configurations (SELECT_ESD_CAN_C200_BOARD_CFG)
          CAN-PCI/200 driver configuration board 0 (INCLUDE_ESD_CAN_C200_BRD_0)
            Backend priority of 1st board (ESD_CAN_C200_BACKEND_0) = 50
            Base net of 1st board (ESD_CAN_C200_BASENET_0) = 0
          CAN-PCI/200 driver configuration board 1 (INCLUDE_ESD_CAN_C200_BRD_1)
          CAN-PCI/200 driver configuration board 2 (INCLUDE_ESD_CAN_C200_BRD_2)
          CAN-PCI/200 driver configuration board 3 (INCLUDE_ESD_CAN_C200_BRD_3)
        CAN-PCI/200 Device Driver (INCLUDE_ESD_CAN_C200)
          Autostart behaviour of driver (ESD_CAN_C200_AUTOSTART) = TRUE
          Driver flags (ESD_CAN_C200_FLAGS) = 0
          Target timestamp counter frequency in kHz (ESD_CAN_C200_TIMESTAMP_FREQUENCY) = 1300
      CAN-PCI/331 boards (FOLDER_ESD_CAN_C331_BOARDS)
      VME-CAN4 boards (FOLDER_ESD_CAN_ICAN4_BOARDS)
      BSP PCI PnP quirk for PCI CAN boards (INCLUDE_ESD_CAN_BSP_PNP_QUIRK)
    CAN support components (private) (FOLDER_ESD_CAN_SUPPORT)
      esd PCI CAN board initialization (INCLUDE_ESD_CAN_PCI_CONFIG)
    Basic CAN test application (INCLUDE_ESD_CANTEST)
    NTCAN-API support (INCLUDE_ESD_NTCAN)
```

CAN-PCI/200 driver configuration board 2

Property	Value	
Name	INCLUDE_ESD_CAN_C200_BRD_2	
Description	CAN-PCI/200 driver configuration board 2	
Synopsis	Board 2 specific driver configuration	
ESD_CAN_C200_BACKEND_2	50	
ESD_CAN_C200_BASENET_2	4	

```
p/vw61_ppc/components/windml-4.0/osconfig/vxworks/h -DFAST_REBOOT -c /backup/vw61_ppc/vxworks-6.1/target/config/comps/src/dataSegPad.c
dld -tPENTIUMLH:vxworks61 -X -N -Xgenerate-paddr -e sysInit -D_VX_START_TEXT=0x00308000  \
    dataSegPad.o romfs_image_prj.o partialImage.o ctdt.o symTbl.o \
    -D_VX_DATA_ALIGN=1 /backup/vw61_ppc/vxworks-6.1/target/h/tool/diab/ldscripts/link.RAM -o vxWorks
```

Below this entry you will find the node *CAN Fieldbus* which contains the configuration of esd CAN drivers.

Below *CAN Fieldbus* there is a node *CAN board drivers* where you can add the driver for the CAN modules you want to support in your project. Every driver support consists of a node *XXX Board Configuration* to include and configure board specific parameter and *XXX Device Driver* to configure driver specific parameter.

Below *CAN Fieldbus* there is a node *CAN support components* to include additional components of the CAN support.
For configuration option and details please refer to property description of the configuration parameter in the Workbench.

5.1.3.2 Configuration of the VxWorks 5.x Driver

Driver for VxWorks 5.x are configured calling the driver start with a pointer to an initialized structure of the type CAN_INFO. This can be either performed modifying the code of the example startup file `caninit.c` (recommended) or by calling the driver start routine from within your own application. In either case please take a look into the `caninit.c` to see configuration examples for several target architectures.

The following text describes the member of the CAN_INFO structure:

```
struct CAN_INFO
{
    unsigned long base;
    unsigned char net[4];
    unsigned char prio;
    unsigned char irq;
    unsigned char reserved[6];
};
```

`base` **CAN-ISA modules**:

I/O-address of the CAN-ISA module.
`base` has to be set to the value configured on the hardware via jumpers or coding switches. If `base` is set to ‘0’, the driver terminates the search for further CAN interfaces.

CAN-PCI modules:

For **x86 host** CPU architectures the base address of the CAN hardware is configured usually by a BIOS. In order to tell the device driver to use this address you have to set the parameter `base` to -1. You have to make that your BSP got enough unused entries in the MMU-Memory Descriptor table to enter the PCI-address area of the CAN module there (this is usually the case).
For **PPC host** CPU architectures where the BSP or a bootloader performs the PCI bus configuration you have to set the parameter `base` to -1. In this case the PCI-address space is accessed with the common VxWorks offset of 0xC0000000. The offset value of the address area is set in the driver.
For **PPC host** CPU architectures without PCI bus configuration by the BSP or bootloader you have to find a 3 MB *free* page size aligned physical address area which address is used for the parameter `base` .

`net[0]` Logical network number which is to be assigned to the first physical CAN port in this module. If a module has got more than one CAN port, the ports get consecutive numbers starting with this number.
The values of `net[1]` to `net[3]` are ignored by the driver.
The user has to make sure that all logical CAN network numbers are only assigned once in a system, otherwise the driver might not be initialized correctly.

`prio` Priority of back-end task, which is, depending on the hardware, started for each physical interface or module.

`irq` **CAN-ISA modules**:

Interrupt vector for this CAN-ISA module

CAN-PCI modules:

For **x86 host** CPU architectures the interrupt of the CAN hardware is configured usually by a BIOS. In order to tell the device driver to use this address you have to set the parameter `irq` to -1.

For **PPC host** CPU architectures where the BSP or a bootloader performs the PCI bus configuration you have to set the parameter `irq` to -1.

For **PPC host** CPU architectures without PCI bus configuration by the BSP or bootloader you have to find out the interrupt used by your target for the PCI slot of the corresponding CAN module in your host hardware manual and set the parameter `irq` to this value regarding an BSP specific interrupt vector offset.

5.1.4 Driver Start

5.1.4.1 Start of the VxWorks 6.x Driver

The driver can either be set to auto start via the Workbench (recommended) or can be started by your application by calling `xxxStart`() where xxx is the driver specific prefix as described below:

CAN Module	Entry Syntax
CAN-ISA/200 CAN-PC104/200 (SJA1000 version)	`c200i`
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/200 CPCI-CAN/200	`c200`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`
CAN-PCI/360 CPCI-CAN/360	`c360`
VME-CAN4	`can4`
CPCI-405 CPCI-CPU/750 PMC-CPU/405 EPPC-405 EPPC-405-HR EPPC-405-UC	This CAN drivers will be started by calling `canStart()`.

Table 31: Entry syntax for starting the driver

5.1.4.2 Start of the VxWorks 5.x Driver

If you use a modified `caninit.c` to start your driver you should call `xxxStart`() otherwise you have to call the `xxx_install()` routine which is exported by the driver with an initialized CAN_INFO structure. In both cases xxx is the driver specific prefix as described above.

5.1.5 Test Application

The driver package comes with the test program `canTest` which can be either optionally configured to your application via the Workbench (VxWorks 6.x) or can be linked to your application or loaded via a target shell (VxWorks 5.x). With the help of this program you can check the function of the CAN interface. `canTest` is described in the chapter 'Test program cantest' in CAN-API manual part 1 [1]. Since VxWorks allows less parameters in the command line as required by `canTest`, the parameters have to be entered as string - in apostrophes. The string is evaluated by the test program. Default values are used for all parameters which are not included in the string.

5.1.6 CAN 2.0B (29-bit Identifier) Support

All CAN modules support 29 bit identifier according to CAN 2.0B. Starting with firmware revision 0.C.09 the following active CAN modules have a firmware which can be switched between 11 bit active and 29 bit passive support and 29/11 bit active support.

CAN-ISA/331,
CAN-PC104/331
CAN-PCI/331,
CPCI-CAN/331,
PMC-CAN/331
CAN-PCI/360,
CPCI-CAN/360
VME-CAN4,

When the driver is running the current firmware mode is put out to *stdout* by calling

```
xxx_switch (op, net)
```

you can switch between the two operating modes while the driver is running where xxx is the driver specific prefix.

`op`	operating mode If '0' is entered for `op`, the current operating mode is shown. If '1' is entered, the firmware recognizes the new operating mode, but only switches to it after the following boot-up.
`net`	logical network number

After a switch of the operating mode you have to reset your hardware.

5.2 Installation Under QNX4 and QNX6

The installation can only be made with superuser rights.
The installation of QNX6 drivers is described from page 103 on.

5.2.1 QNX4

5.2.1.1 Files of the QNX4 Packages

The software drivers for the QNX package are shipped on a 3.5” disk. The disk contains the following files:

File	Description	
`readme`	current notes and installation information	The character combination ‘*c331*’ signifies files of the CAN-PCI/331 module. For other modules the characters have to be changed as follows:
`c331`	CAN resource manager for CAN-PCI/331	
`updc331`	program for firmware update	
`libntcan3s.lib`	CAN-API library (stack parameter)	
`libntcan3r.lib`	CAN-API library (register parameter)	
`ntcan.h`	header for CAN-API	
`cantest.c`	source code of example program ‘cantest’ (see CAN-API manual part 1 [1])	
`cantest`	binary file of example program ‘cantest’	

CAN Module	File Name
CAN-ISA/200	`c200i`
CAN-PC104/200 (SJA1000 version)	`c200i`
CAN-ISA/331 CAN-PC104/331	`c331i`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`

Table 32: Files of the QNX4 packages

5.2.1.2 Sequence of Installation Under QNX4

5.2.1.2.1 Installation of ISA-Boards

First install the hardware of the module. When installing ISA boards, please make sure to set the board address correctly. The address corresponding to the standard configuration (e.g. `0x1E8` for CAN-ISA/200) should be set before shipping already.

Calling the Resource Manager
Log in as 'root'.
Start the resource manager (e.g. for QNX4, CAN-ISA/200): '`c200i &`'

After this call an output similar to the following is to appear on the screen:

```
C200i[0x1e8]: Using I/O-Base 0x1E8
C200i[0x1e8]: Using Interrupt 5
C200i[0x1e8]: "CAN_ISA200" with 1 Nets identified
C200i[0x1e8]: Hardware-Version=1.0.00
C200i[0x1e8]: Firmware-Version=0.0.00
C200i[0x1e8]: Driver-Version  =1.0.00
C200i[0x1e8]: Net 0 successfully created
```

Note: Depending on the module the resource manager is called by means of different commands. The character combination '`c200i`' for QNX4 shown in the example above, has to be substituted for other modules as follows:

CAN Module	Entry Syntax QNX4
CAN-ISA/200	`c200i`
CAN-PC104/200 (SJA1000 version)	
CAN-ISA/331 CAN-PC104/331	`c331i`

Table 33: Input syntax when calling the resource manager

Command Line of the Resource Manager for ISA Boards

The resource manager can be configured by means of the parameters explained below.

Parameter	Function
-h	Showing a help text
-n *net*	Assigning the logical network number. *net* = 0...255 default: *net* = 0
-p *port*	Selecting the ISA board in the system by means of the port address set via the hardware (see also hardware manual of the module).
-i *irq*	Setting the interrupt to be used by the board. You have to determine an available interrupt which has to be set here. If this parameter is not specified, interrupt 5 will be used.

Table 34: Parameters of the resource manager on ISA boards

Example for the installation of two CAN-ISA/200 boards in one system:

Call	Function
`c200i -p0x1e8 -i5 -n0 &`	Interrupt 5 and the CAN network number 0 are assigned to the CAN-ISA/200 module with the address 0x1E8 .
`c200i -p0x1e0 -i7 -n1 &`	Interrupt 7 and the CAN network number 1 are assigned to the CAN-ISA/200 module with the address 0x1E0 .

Table 35: Example for the installation of two CAN-ISA/200 boards

Now you can test the communication on the CAN bus by means of the example program `cantest`.

Attention!	Make sure that your module is correctly wired! A second CAN participant is required to make the module work correctly, because otherwise the CAN controller creates error messages. Make also sure that the bus is correctly terminated.

5.2.1.2.2 Installation of PCI-Boards

First install the module hardware.

Log in as ‘root’.
Start the resource manager (example CAN-PCI/331): ‘`c331 &`’

Now an output similar to the one below has to appear on the screen:

```
C331[0]: Using Interrupt 12
C331[0]: "CAN_PCI331" with 2 Nets identified
C331[0]: Hardware-Version=1.1.00
C331[0]: Firmware-Version=0.c.00
C331[0]: Driver-Version  =1.0.00
C331[0]: Net 0: Successfully created
C331[0]: Net 1: Successfully created
```

Note: Depending on the module the resource manager might be called by means of different commands. The character combination ‘*`c331`*’ , as shown in the example above, has to be substituted for other modules as shown below:

CAN Module	Entry Syntax QNX4
CAN-PCI/200 CPCI-CAN/200 CAN-PCI/266	-
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`c331`

-... has not been implemented yet

Table 36: Input syntax when calling the resource manager on PCI boards

Command Line of the Resource Manager for PCI Boards

The resource manager can be configured by means of the parameters described below.

Parameter	Function
-h	Showing a help text.
-n *net*	Assigning the logical network number. *net* = 0...255 default: *net* = 0
-p *index*	Selecting the esd-CAN-PCI boards in the system. The boards are numbered starting with ‘0’. The assignment of numbers and boards is determined by the plug-and-play controller. In case of doubt, the assignment should be checked by means of a test.

Table 37: Parameters of the resource manager on PCI boards

Example for the installation of two CAN-PCI/331 boards in one system:

Call	Function
`c331 -p0 -n0 &`	The CAN network numbers 0 and 1 are assigned to the CAN-PCI/331 module with the PCI number 0.
`c331 -p1 -n2 &`	The CAN network numbers 2 and 3 are assigned to the CAN-PCI/331 module with the PCI number 1.

Table 38: Example for the installation of two CAN-PCI/331 boards

Now you can test the communication on the CAN bus by means of the example program `cantest`.

Attention!	Make sure that your module is correctly wired! A second CAN participant is required to make the module work correctly, because otherwise the CAN controller creates error messages. Make also sure that the bus is correctly terminated.

5.2.2 QNX6

5.2.2.1 Files of the QNX6 Package

The software drivers for the QNX package are shipped on a 3.5” disk. The disk contains the following files:

File	Description	
`readme`	current notes and installation information	The character combination ‘*`pci331-i20`*’ signifies files of the CAN-PCI/331 module. For other modules the characters have to be changed as follows
`CHANGELOG`	change list / driver history	
`devcan-pci331-i20`	CAN resource manager for CAN-PCI/331 under QNX6	

`upd-isa331`
`upd-pci331`
`upd-pci360`
	programs for firmware update (shipping on request)	

`libntcan.so.`*`x`*	dynamic CAN-API library (*x*... library version)	
ntcan.h	header of CAN-API	
cantest.c	source code of example program ‘cantest’ (see CAN-API manual part 1 [1])	
cantest	binary file of example program ‘cantest’	

CAN Module	Code
CAN-ISA/200 CAN-PC104/200 (SJA1000 version)	`...isa200-sja1000`
CAN-PCI/200 CAN-PCI/104/200 CAN-PCIe/200 CPCI-CAN/200 CAN-PCI/266 PMC-CAN/266	`...pci200-sja1000`
CAN-ISA/331 CAN-PC104/331	`...isa331-i20`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`...pci331-i20`
CAN-PCI/360 CPCI-CAN/360	`...pci360-i20`
CPCI-405	`...cpci405-pcimsgx`

Table 39: Files of the QNX6 package

5.2.2.2 Sequence of Installation Under QNX6

5.2.2.2.1 Installation of ISA-Boards

First install the hardware of the module. When installing ISA boards, please make sure to set the board address correctly. The address corresponding to the standard configuration (e.g. 0x1E8 for CAN-ISA/200) should be set before shipping already.

Calling the Resource Manager
Log in as 'root'.
Start the resource manager (e.g. for CAN-ISA/200):

```
devcan-isa200-sja1000 -v &
```

After this call an output similar to the following is to appear on the screen:

```
    CAN_ISA_200 version 3.3.0 build 17:03:04 Dec 15 2004
    Card[0]: Baud=0x7fffffff Mode=0x00000000 Nodes=can0
    Card[0]: Hardware-version 1.0.0
```

Depending on the module the resource manager is called by means of different commands. The code *`isa200-sja1000`* in the character combination '`devcan-`*`isa200-sja1000`*' shown in the example above, has to be substituted for other modules as described in table 39 on page 103.

Command Line of the Resource Manager for ISA Boards:

With the command

```
devcan-isa200-sja1000 -h
```

the input syntax of all available parameters (here CAN-ISA/200) is listed:

```
devcan-isa200-sja1000 [-b baudrate][-m mode][-n net][-v[v..]][-c card-options] &
```

The resource manager can be configured by means of the parameters explained below.

Parameter	Function
`-b` *baudrate*	Input of baudrate (default value: module is ‘OFF Bus’)
`-m` *mode*	Not supported at the moment (default value = 0)
`-n` *net*	Assigning the logical network number. *net* = 0...255 (default value = 0)
`-v[`*v...*`]`	Verbose level
`-p` *prio*	Back-end handling priority (default value = 19)
`-h`	Showing a help text
`-c` *card_options*	Setting of the I/O-address and the interrupts of a CAN module: `io =` *address* (I/O-address) irq = *interrupt* (An unused interrupt of the system has to be determined and set with this parameter.)

Table 40: Parameters of the resource manager on ISA boards

Example for the installation of two CAN-ISA/200 boards in one system:

```
devcan-isa200-sja1000 -n0 -c io=0x1e8,irq=5 -n1 -c io=0x1e0,irq=7 &
```

Now you can test the communication on the CAN bus by means of the example program `cantest`.

Attention!	Make sure that your module is correctly wired! A second CAN participant is required to make the module work correctly, because otherwise the CAN controller creates error messages. Make also sure that the bus is correctly terminated.

5.2.2.2.2 Installation of PCI-Boards

First install the module hardware.

Log in as ‘root’.
Start the resource manager (example CAN-PCI/331):

```
devcan-pci331-i20 -v &
```

Now an output similar to the one below has to appear on the screen:

```
    pci331 version 3.3.1 build 17:01:52 Dec 15 2004
    Card[0]: Baud=0x7fffffff Mode=0x00000000 Nodes=can0 can1
    Card[0]: Hardware-version 1.1.4 Firmware-version 0.12.30
```

Note: Depending on the module the resource manager might be called by means of different commands. The code *`pci331-i20`* in the character combination ‘`devcan-`*`pci331-i20`*’, as shown in the example above, has to be substituted for other modules as described in table 39 on page 103.

Command Line of the Resource Manager for PCI Boards:

With the command

```
devcan-pci331-i20 -h
```

the input syntax of all available parameters (here CAN-PCI/331) is listed:

```
devcan-pci331-i20 [-b baudrate][-m mode][-n net][-v[v..]][-c card-options] &
```

The resource manager can be configured by means of the parameters described below.

Parameter	Function
`-b` *`baudrate`*	Input of baudrate (default value: module is ‘OFF Bus’)
`-m` *`mode`*	Not supported at the moment (default value = 0)
`-n` *`net`*	Assigning the logical network number. *net* = 0...255 (default value = 0)
`-v[`*`v...`*`]`	Verbose level
`-p` *`prio`*	Back-end handling priority (default value = 19)
`-h`	Showing a help text
`-c` *`card_options`*	`pci =` *`index`* (= “PCI Index”as indicated by the QNX Shell command “`pci -v`” `sdid =` *`id`* (= “Subsystem ID”as indicated by the QNX Shell command “`pci -v`” If no parameter value is set, access to all boards is enabled).

Table 41: Parameters of the resource manager on PCI boards

Example for the installation of two CAN-PCI/331 boards in one system:

```
devcan-pci331-i20 -n0 -c pci=0 -n2 -c pci=1 &
```

Now you can test the communication on the CAN bus by means of the example program `cantest`.

Attention!	Make sure that your module is correctly wired! A second CAN participant is required to make the module work correctly, because otherwise the CAN controller creates error messages. Make also sure that the bus is correctly terminated.

6. Update of Local Firmware for Non-Windows Operating Systems

6.1 Overview

The update program for the local firmware is implemented **only for Linux, LynxOS, VxWorks and QNX operating systems** at the moment.
The update function is insignificant for the CAN-xxx/200 board and CPCI-xxx/200, because this boards do not have a local processor. The CAN-PCC module also does not support the update function.

The firmware for the local processor of the intelligent CAN boards is stored into Flash-EPROM memory components. When the board is shipped you will receive the firmware version which had been the latest at the time of the *hardware manufacturing*.

The disks or CD-ROM, included in the shipping, contain an update program by means of which you can program the flash EPROM with the latest software at the *time of shipping*.

Depending on the module and the operating system you will either find the update program '**`canupd`**' or the update program '**`upd`*`xxx`***' on your data carrier. The programs will be installed onto your hard disk during the installation of the driver.
Both programs are equal in functionality. While `canupd` is a universal update program for various modules, `upd`*`xxx`* offers specific updates for individual modules.

The modules are identified by means of the character combination when called:

CAN Module	Entry Syntax for `upd`*`xxx`*			
	LynxOS	VxWorks, QNX4	Linux	QNX6
CAN-ISA/331 CAN-PC104/331	`updc331i`		`updisa331`	`upd-isa331`
CAN-PCI/331 CPCI-CAN/331 PMC-CAN/331	`updc331`		`updpci331`	`upd-pci331`
CAN-PCI/360	n.a.	`updc360`	`updpci360`	`upd-pci360`
CAN-PCI/405	n.a.	1*)		
CPCI-CAN/360	n.a.	⚠ 2*)		
USB-Mini	n.a.		`updusb331`	n.a.

Table 42: Entry syntax for update program `upd`*`xxx`*

1*) In contrast to the other CAN modules the firmware of the CAN-PCI/405 is part of the driver binary and automatically downloaded during driver startup. No extra action is required.

2*) Please contact our support team for more information about CPCI-CAN/360 firmware update.

⚠ **Attention!** Do not execute `updc360` or `updpci360` for the **CPCI**-CAN/360 module! This will produce a faulty firmware programming that is complexly to repair.

6.2 Checking the Firmware Version Number

With the update program you can check the local firmware revision of the Flash EPROM and update the Flash EPROM, if necessary.

It is recommended, to compare the revision number of the delivered firmware at the floppy disk with the firmware stored in the Flash EPROM after the host driver installation!

Calling the program `canupd` or `upd`*`xxx`* without parameters returns the output of the included firmware program with revision numbers onto the monitor. Because `canupd` is a universal tool, it includes firmware programs for several hardware applications.

Under the operating system Linux it is also possible to check the firmware version in the directory `/var/log/messages` after loading the driver.

Calling the program with the following number of the logical network number of the required board returns the output of the revision numbers of the local firmware stored in the Flash EPROM onto the monitor.

Input example: `canupd 0`

The program automatically compares the firmware versions and shows in clear text on the monitor, whether the stored version is still up-to-date, or if an update is required.

6.3 Executing the Update

1. Calling `canupd` or `updxxx` with desired logical network number (see above).

2. After the version has been checked, you will be asked, whether the program is to carry on with the update or not.

   - If you enter 'y' for YES, the Flash-EPROM is then being programmed again, **regardless of the firmware version which is more up-to-date**.

   - 'n' for NO cancels the procedure. The firmware in the Flash-EPROM remains unchanged.

3. In order to activate the new firmware, the local processor has to be reset. This can be achieved by stopping and restarting the software driver of the host-CPU.

The following table shows the command entries which are required for this procedure:

CAN Module	Stopping and Restarting the CAN Driver with Operating System ...			
	LynxOS	VxWorks	QNX	Linux

CAN-ISA/331	stop:
	`instcan -u c331i`
	
	restart:
	`instcan c331i`
		stop:
		terminate system
		
		restart:
		`c331iStart()`
			stop:
			terminate shell by
			`Ctrl-C`
			or
			terminate process by
			`kill` *`process`*
			
			restart: `c331i &`
				stop syntax:
				see Linux installation description on page 66 *)
				
				restart syntax:
				see Linux installation description on page 66 *)

CAN-PC104/331				

CAN-PCI/331	stop:
	`instcan -u c331`
	
	restart:
	`instcan c331`
		stop:
		terminate system
		
		restart:
		`c331Start()`
			stop:
			terminate shell by
			`Ctrl-C`
			or
			terminate process by
			`kill` *`process`*
			
			restart: `c331 &`	

CPCI-CAN/331				
PMC-CAN/331				
CAN-PCI/360	-	stop: terminate system restart: `c360Start()`	stop: terminate shell by `Ctrl-C` or terminate process by `kill` *`process`* restart: `c360 &`	
CAN-USB-Mini	-	-	-	

*) Supported Linux kernel and standard configuration prerequisited.

Table 43: Commands for stopping and restarting the CAN drivers

7. References

[1] CAN-API Manual, Part 1: Function Description
esd electronic system design gmbh

Index