# ICD Quick Installation

# ICD Quick Installation

**Version 18-Nov-2020**

## History

16-Dec-19 Details on power supplies added to chapter “Tool Configuration”.

15-Aug-18 Chapter “Tool Configuration” was updated.

# Warning

**WARNING:**

To prevent debugger and target from damage it is recommended to connect or disconnect the debug cable only while the target power is OFF.

Recommendation for the software start:

1. Disconnect the debug cable from the target while the target power is off.
2. Connect the host system, the TRACE32 hardware and the debug cable.
3. Power ON the TRACE32 hardware.
4. Start the TRACE32 software to load the debugger firmware.
5. Connect the debug cable to the target.
6. Switch the target power ON.
7. Configure your debugger e.g. via a start-up script.

Power down:

1. Switch off the target power.
2. Disconnect the debug cable from the target.
3. Close the TRACE32 software.
4. Power OFF the TRACE32 hardware.

**Important Information Concerning the Use of the TRACE32 Development System**

Due to the special nature of the TRACE32 development system, the user is advised that it can generate higher than normal levels of electromagnetic radiation which can interfere with the operation of all kinds of radio and other equipment.

To comply with the European Approval Regulations therefore, the following restrictions must be observed:

1. The development system must be used only in an industrial (or comparable) area.
2. The system must not be operated within 20 metres of any equipment which may be affected by such emissions (radio receivers, TVs etc).

# Introduction

This manual introduces the typical configurations for the TRACE32 hardware-based debug and trace tools and provides guidance on installing the TRACE32 software for this product group.

## How This Manual is Organized

- **Brief Overview of Documents for New Users:** Informs new users about important architecture-independent and architecture-specific documents.
- **Tool Configuration:** Provides information about and illustrations of the TRACE32 debug and trace tools.
- **Software Installation:** Describes the default installation of TRACE32 under MS Windows and PC Linux, Mac OS and SunOS.

## Contacting Support

LAUTERBACH GmbH
Altlaufstrasse 40
85635 Hoehenkirchen-Siegertsbrunn
Germany

**Phone** (+49) 8102-9876-555

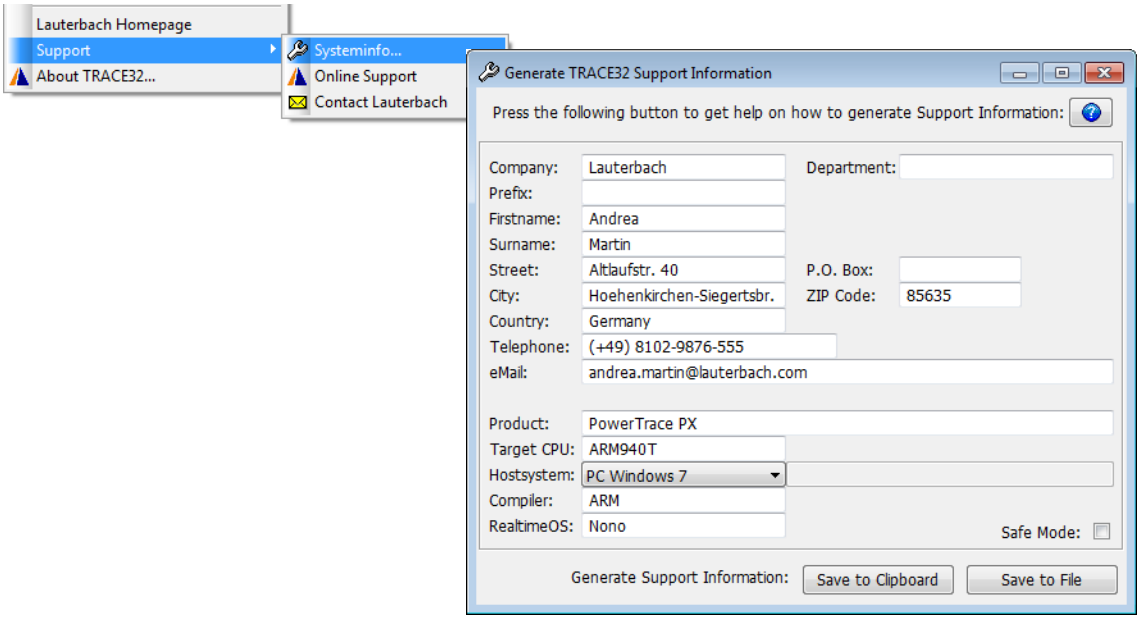
**Fax** (+49) 8102-9876-187

**Internet** https://www.lauterbach.com/tsupport.html or https://www.lauterbach.com/report.html
Here you'll find local and special support addresses.

**E-mail** support@lauterbach.com
General support address where your request will be answered within a short time if it is a basic support request or redirected to the appropriate address.

Be sure to include detailed system information about your TRACE32 configuration.

1. To generate a system information report, choose **TRACE32** > **Help** > **Support** > **Systeminfo**.




| | |
|---|---|
| **NOTE:** | Please help to speed up processing of your support request. By filling out the system information form completely and with correct data, you minimize the number of additional questions and clarification request e-mails we need to resolve your problem. |

2. Preferred: click **Save to File**, and send the system information as an attachment to your e-mail.

3. Click **Save to Clipboard**, and then paste the system information into your e-mail.

# Brief Overview of Documents for New Users

**Architecture-independent information:**

- **"Debugger Basics - Training"** (training_debugger.pdf): Get familiar with the basic features of a TRACE32 debugger.
- **"T32Start"** (app_t32start.pdf): T32Start assists you in starting TRACE32 PowerView instances for different configurations of the debugger. T32Start is only available for Windows.
- **"General Commands"** (general_ref_*<x>*.pdf): Alphabetic list of debug commands.

**Architecture-specific information:**

- **"Processor Architecture Manuals"**: These manuals describe commands that are specific for the processor architecture supported by your debug cable. To access the manual for your processor architecture, proceed as follows:

  - Choose **Help** menu > **Processor Architecture Manual**.

- **"OS Awareness Manuals"** (rtos_*<os>*.pdf): TRACE32 PowerView can be extended for operating system-aware debugging. The appropriate OS Awareness manual informs you how to enable the OS-aware debugging.

# Tool Configuration

This chapter gives a brief overview of typical TRACE32 tool configurations. Your final tool configuration can, of course, be more complex, especially if adapters or converters are needed.

## Power Supply

When your TRACE32 hardware is delivered, you will receive one of the following power supplies:

- **Wall Mount Power Supply**

  The Wall Mount Power Supply is delivered with the PowerDebug Module USB 3.0 and the uTrace.

- **Desktop Power Supply**

Please ONLY use the delivered power supplies. The following schematic drawings for the basic configuration always show the power supply to be used.

If you want to cascade several modules, please contact our support (https://www.lauterbach.com/tsupport.html) to find out which power supply must be used.

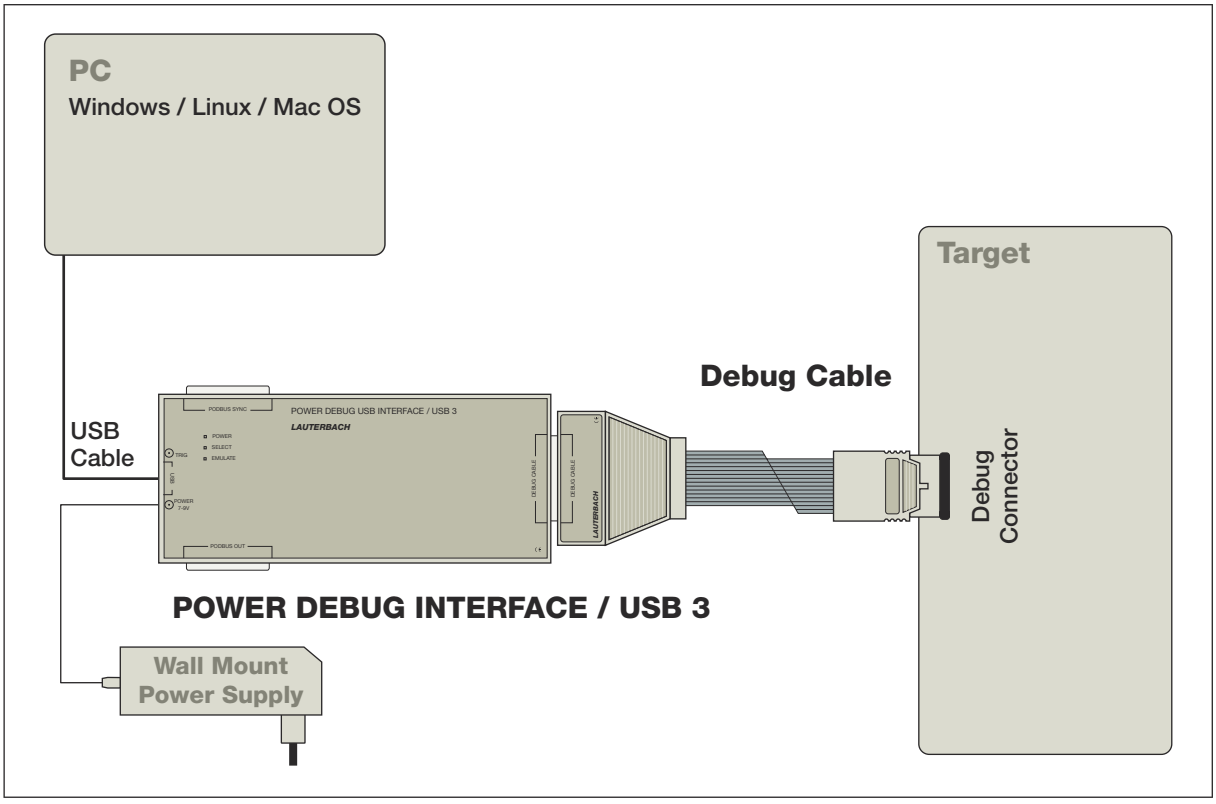
## TRACE32 Debug Tools

A TRACE32 hardware-based debugger consists of:

- A universal debugger hardware
- A debug cable specific to the (main) processor architecture under debug;

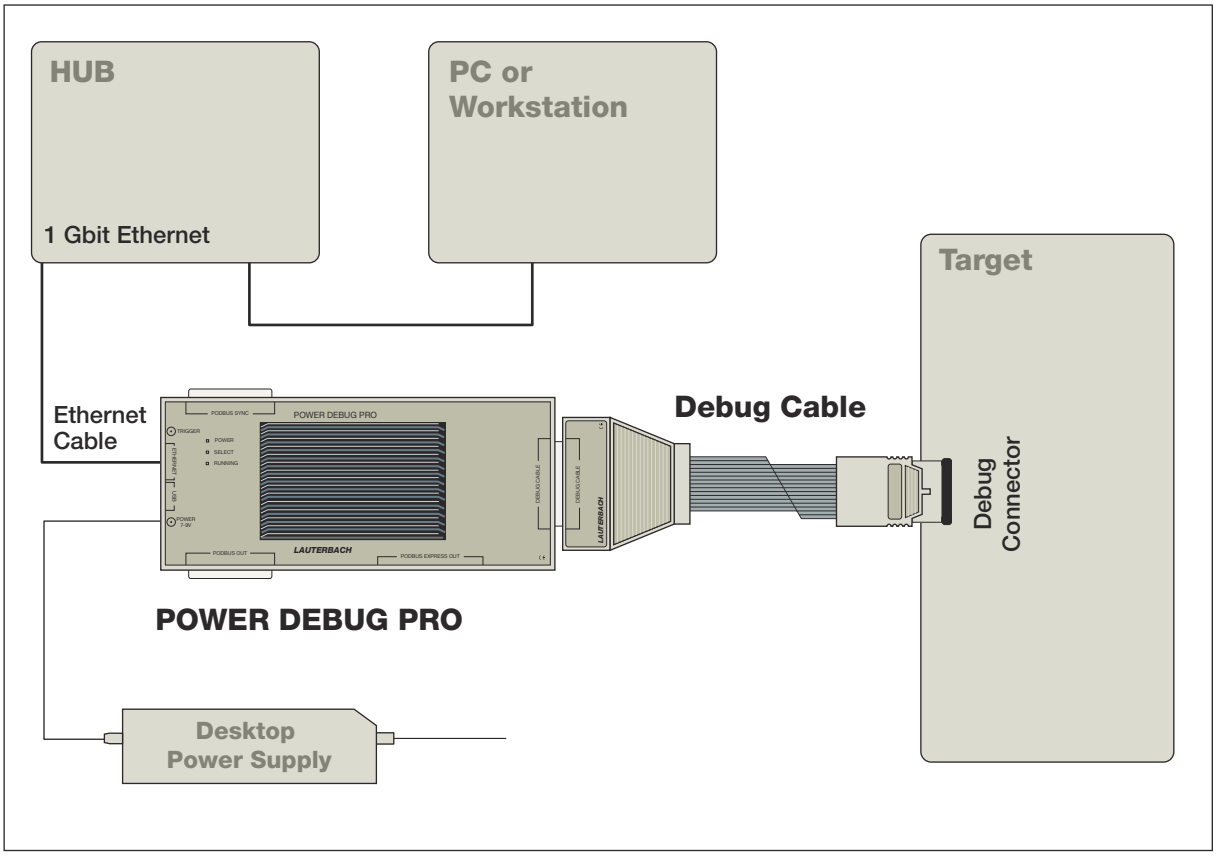
  The debug cable can contain a multicore license or debug licenses for further processor architectures if a multicore chip should be debugged. It can also contain traces licenses mainly used to decode core trace information stored in an onchip trace RAM.

The POWER DEBUG INTERFACE provides a USB3 interface to the host computer.



POWERDEBUG PRO provides:

- USB3 and Gigabit Ethernet interface to the host computer
- PODBUS EXPRESS interface to connect a **TRACE32 POWERTRACE**.




Discontinued products:

- POWER DEBUG II
- **POWER DEBUG ETHERNET**

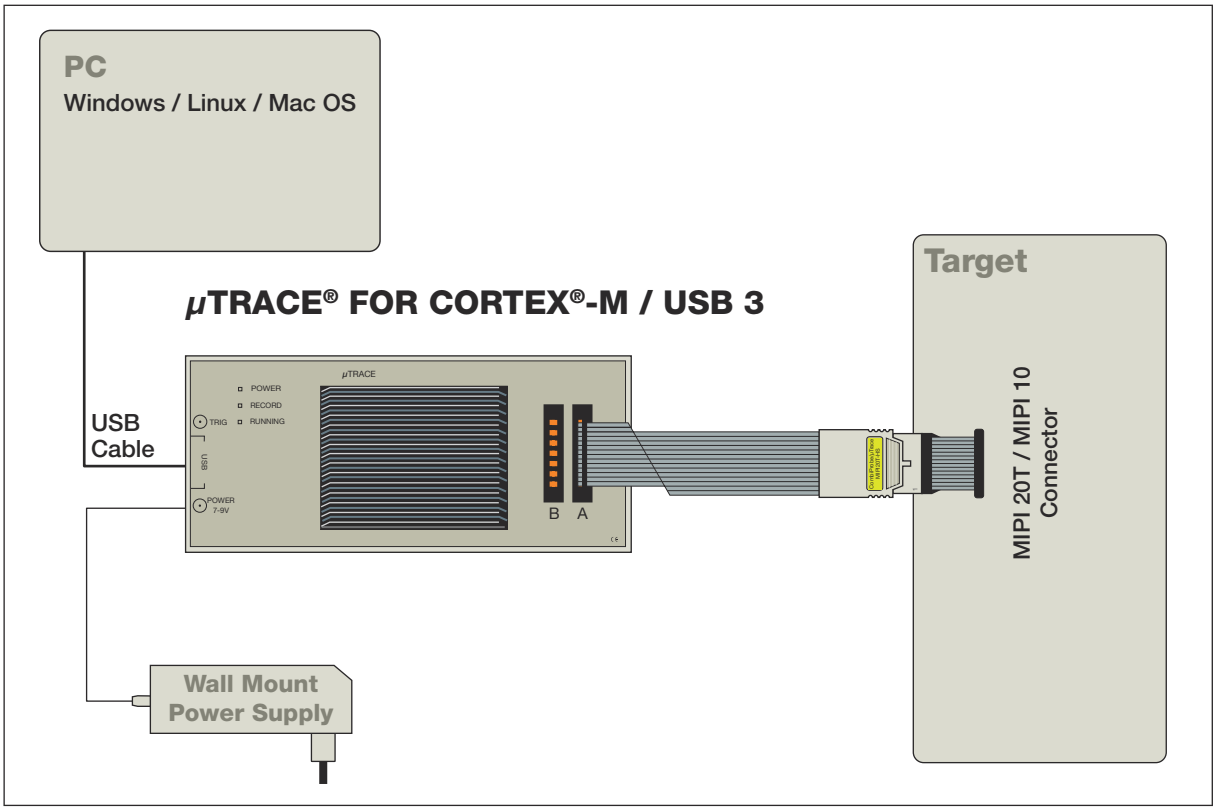
# µTrace for Cortex-M

uTrace is an all-in-one debug and trace tool especially designed for Cortex-M processors. It consists of:

- A uTrace module, that also provides 256 MByte of trace memory
- A Whisker MIPI20T-HS

The uTrace module can contain a multicore license, if a chip containing more than one Cortex-M core should be debugged.

Not-Cortex-M cores can not be debugged by the uTrace.




Discontinued products:

- **uTrace for Cortex-M MIPI34**

## Arm/Cortex

The CombiProbe for Arm/Cortex has two main applications:

- It allows debugging via two debug connectors if the (main) core is an Arm or Cortex core
- It allows multicore debugging of a Cortex-M core and a not-Cortex-M core as well as Cortex-M tracing

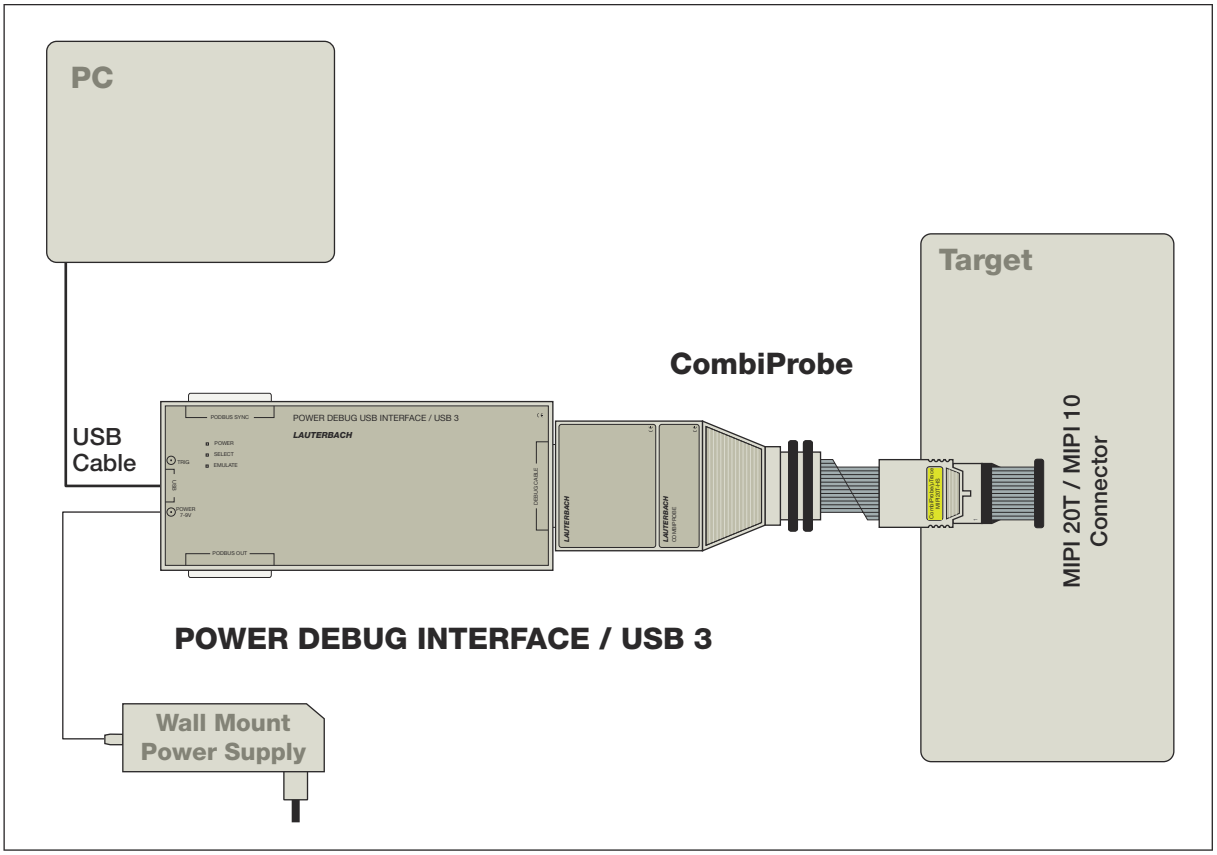
The TRACE32 hardware-based debug and trace tool can consist of:

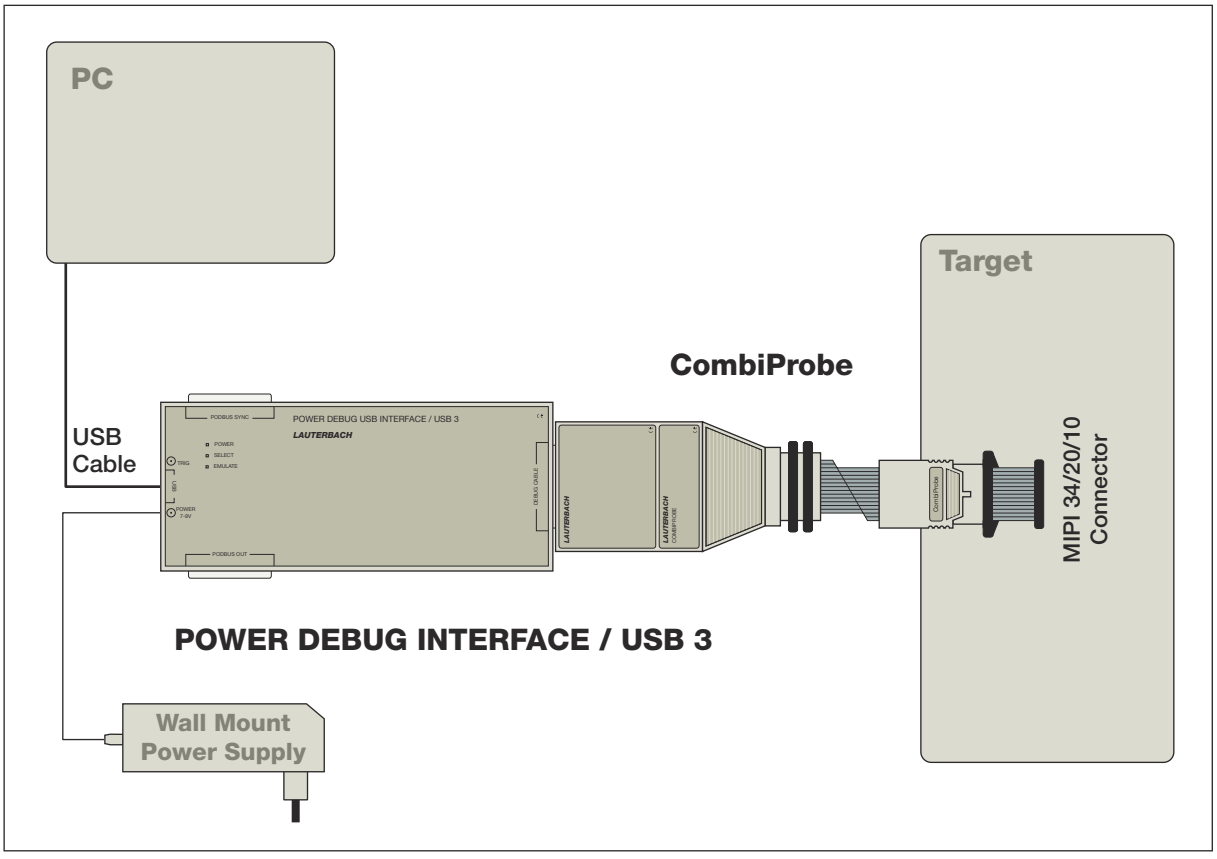
- A universal debugger hardware
- A CombiProbe hardware licensed for debugging of an Arm or Cortex core.

  The CombiProbe is an all-in-one debug and trace tool that provides 128 MB of trace memory

  The CombiProbe can contain a multicore license or debug licenses for further processor architectures if a multicore chip should be debugged
- One/two Whisker MIPI20T-HS or one/two Whisker MIPI34

### POWER DEBUG INTERFACE / USB 3 plus COMBIPROBE MIPI20T






Discontinued products:

- **POWER DEBUG INTERFACE / USB2 and COMBIPROBE**
- **POWER DEBUG II and POWER TRACE II and COMBIPROBE**

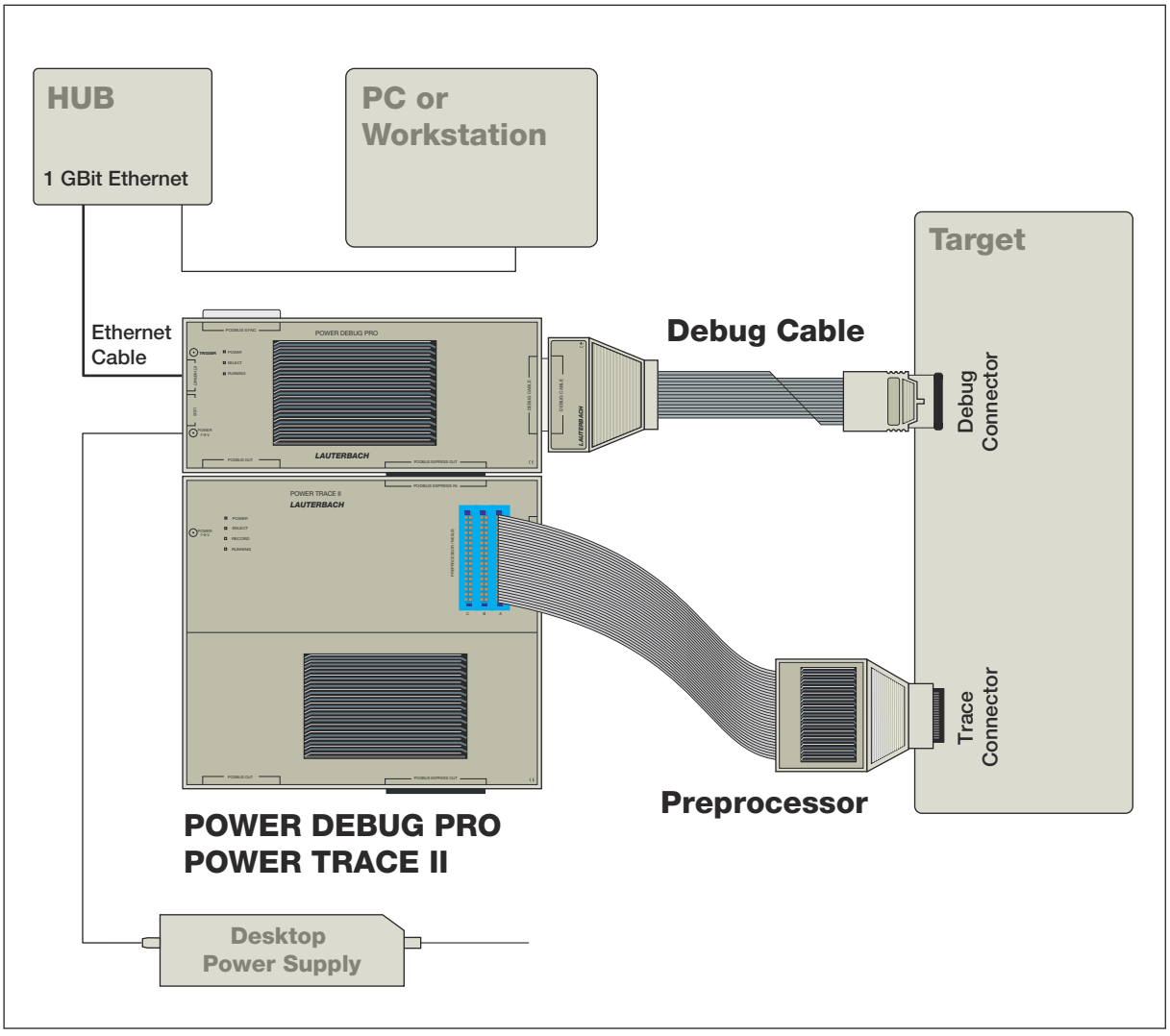
## Tools with Parallel or Serial Preprocessors

A TRACE32 hardware-based debug and trace tool can consist of:

- The universal debugger hardware **POWER DEBUG PRO**
- A debug cable specific to the (main) processor architecture under debug;

  The debug cable can contain a multicore license or debug licenses for further processor architectures if a multicore chip should be debugged.
- A universal trace module POWER TRACE II with 1 GByte, 2 GByte or 4 GByte of trace memory or POWER TRACE II LITE with 512 MByte of trace memory
- A parallel or serial preprocessor specific to the processor architecture and its trace protocol

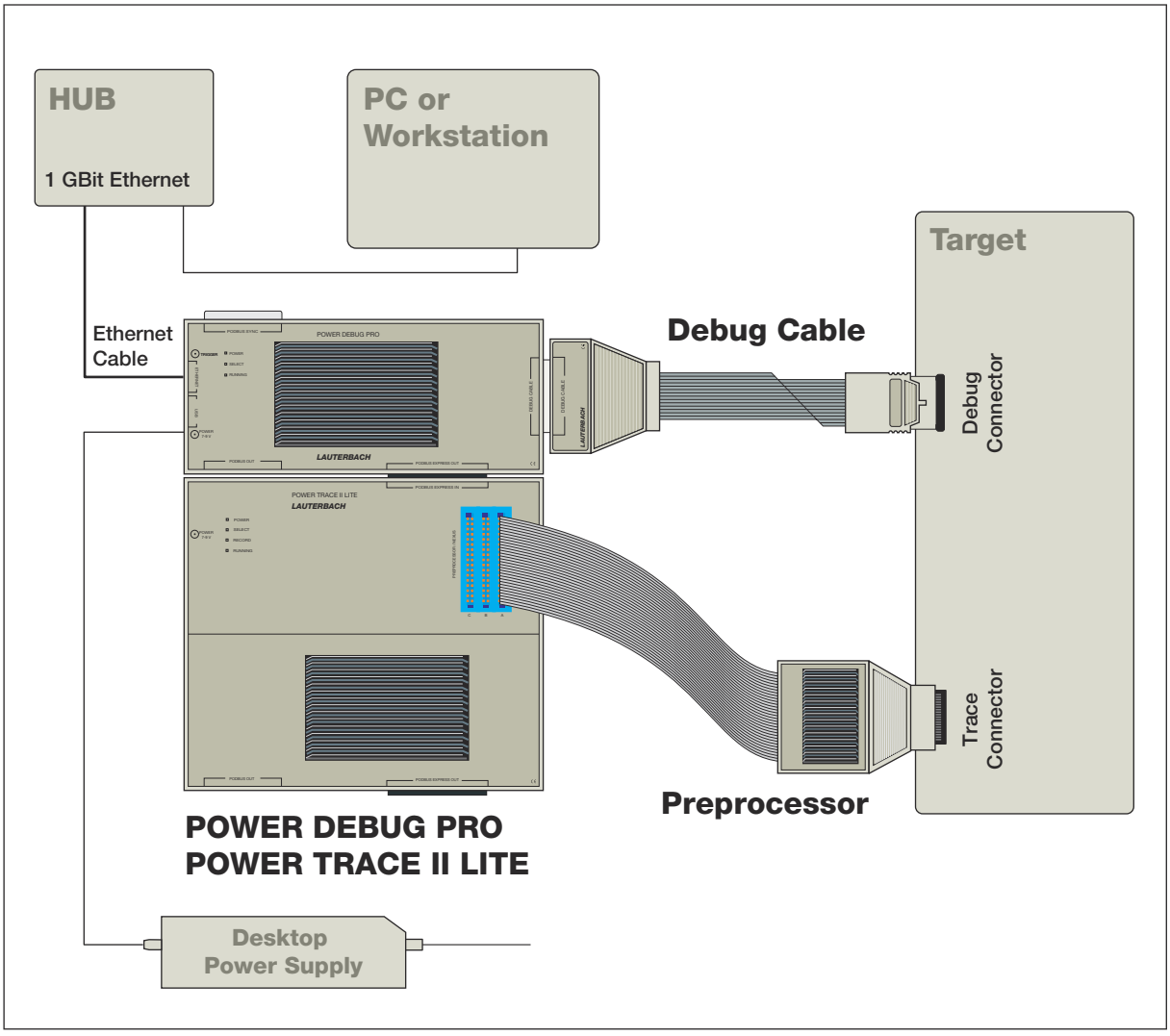
  The preprocessor can contain trace licenses for further processor architectures if a multicore chip exports trace information in various trace protocols.




Discontinued products:

- **POWER DEBUG II and POWER TRACE II**

POWER TRACE II LITE is not supported for all processor architectures and is not suitable for all targets due to its bandwidth limitation of 9.6 GBit/s.



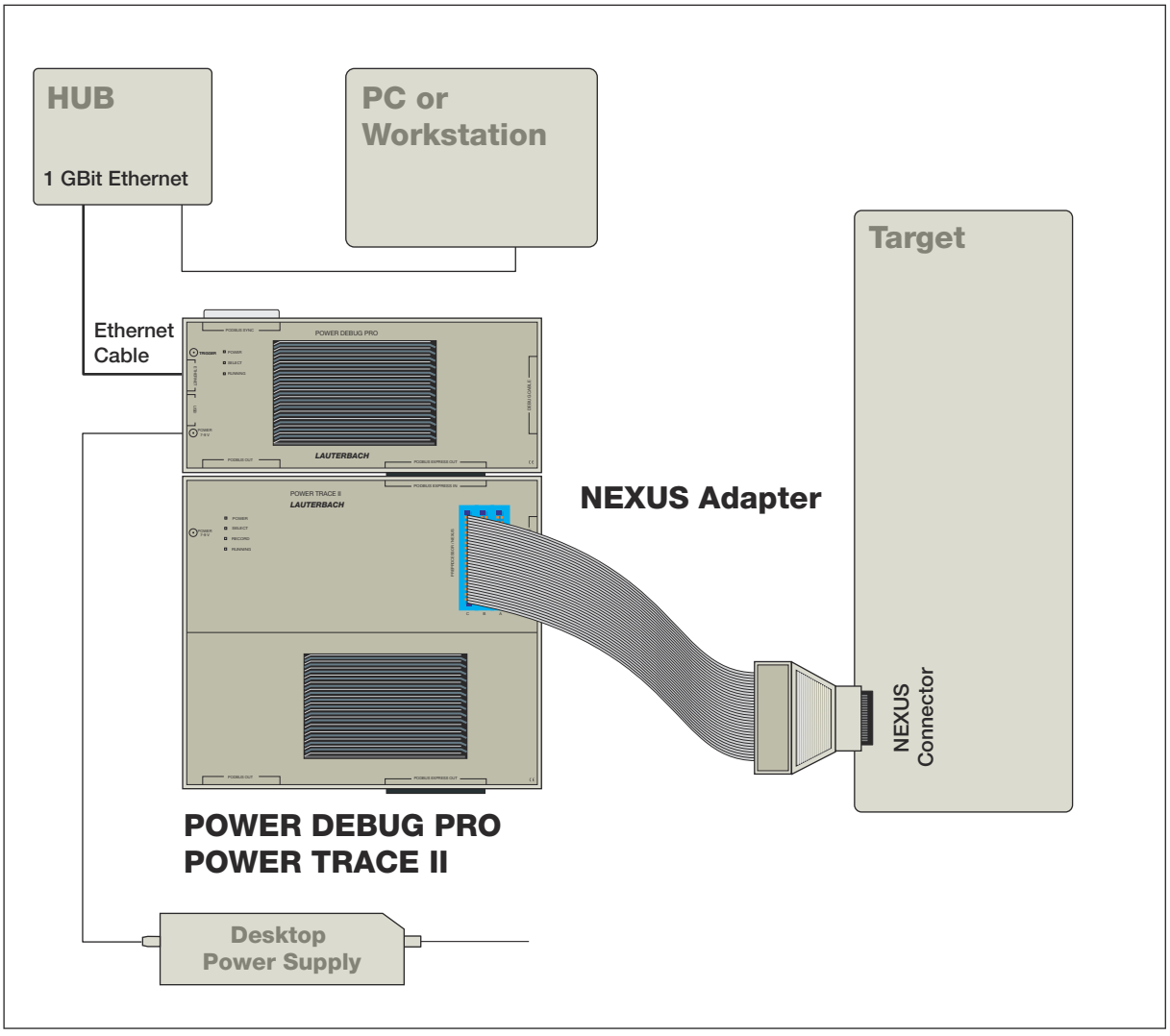

Discontinued products:

- **POWER TRACE / ETHERNET**

A TRACE32 hardware-based debug and trace tool can consist of:

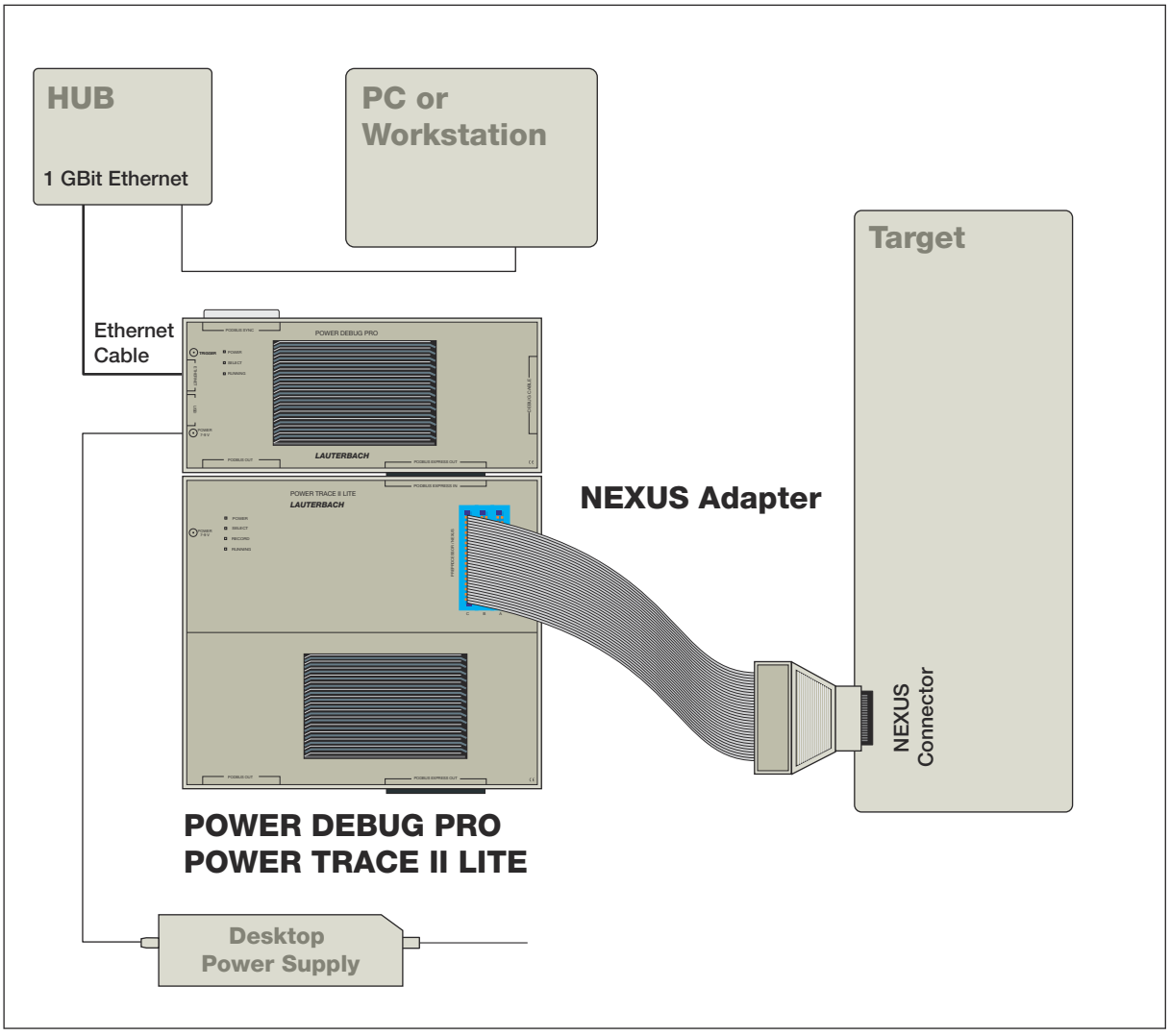
- The universal debugger hardware **POWER DEBUG PRO**
- A universal trace module POWER TRACE II with 1 GByte, 2 GByte or 4 GByte trace memory or POWER TRACE II LITE with 512 MByte of trace memory
- A parallel NEXUS adapter specific for the processor architecture under debug and trace

  The NEXUS adapter can contain debug/trace licenses for further processor architectures if a multicore chip is under debug




Discontinued products:

- **POWER DEBUG II and POWER TRACE II for NEXUS**




Discontinued products:

- **POWER TRACE / ETHERNET for NEXUS**

A TRACE32 hardware-based debug and trace tool can consist of:

- The universal debugger hardware **POWER DEBUG PRO**
- A debug cable specific to the (main) processor architecture under debug;

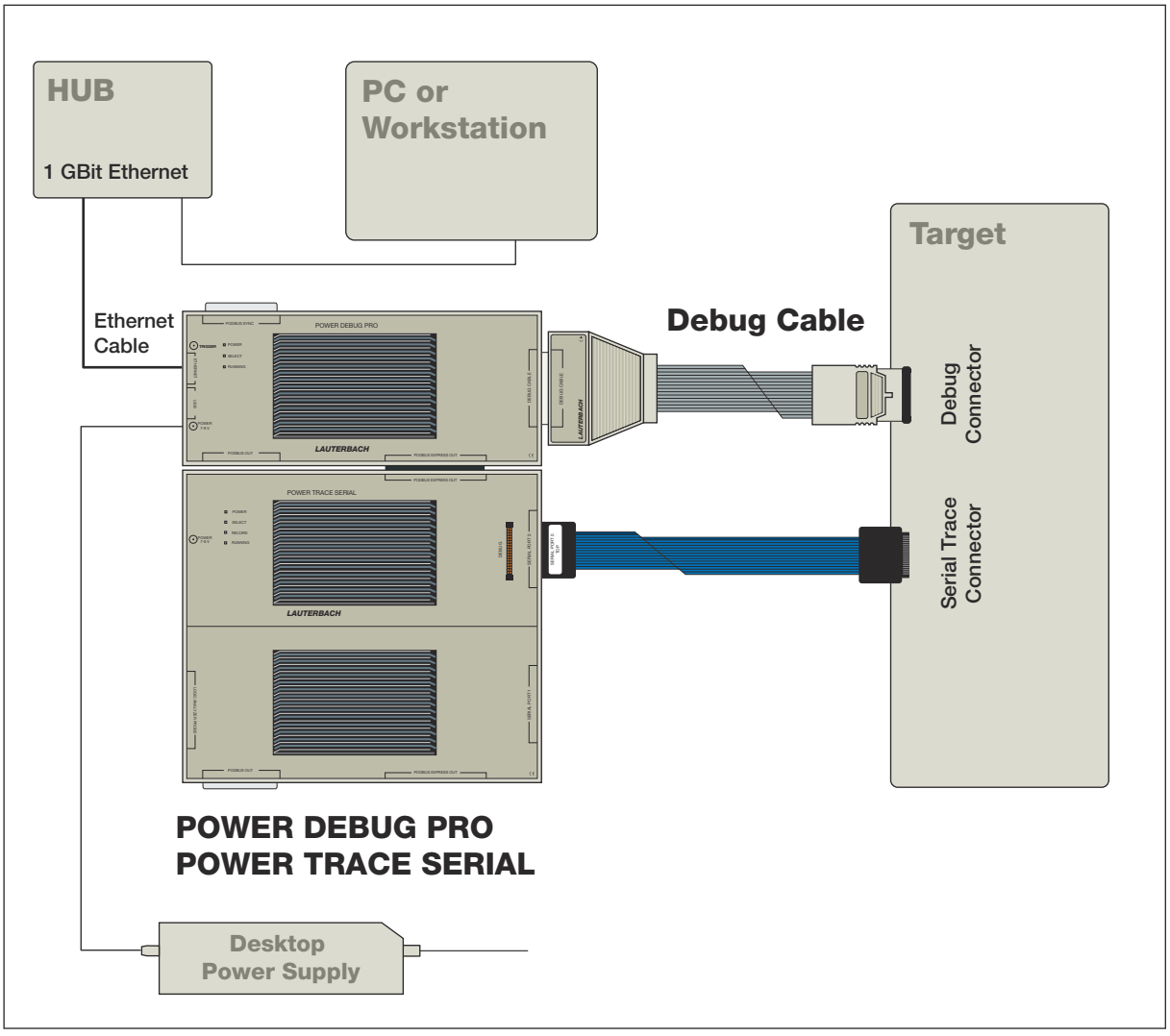
  The debug cable can contain a multicore license or debug licenses for further processor architectures if a multicore chip should be debugged.
- A universal trace hardware POWER TRACE SERIAL with 4 GByte of trace memory licensed for a processor architecture and its trace protocol

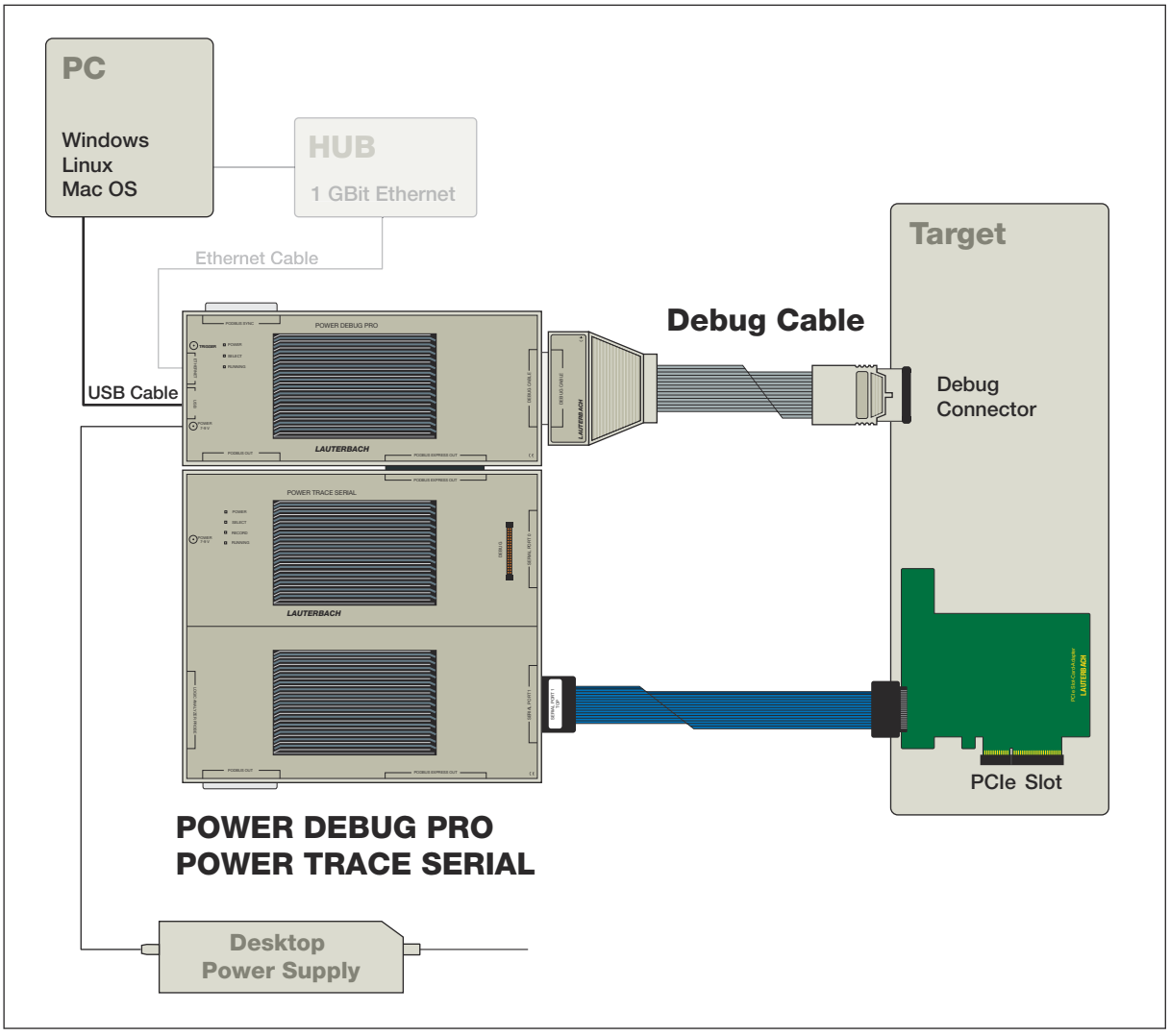
  POWER TRACE SERIAL can contain trace licenses for further processor architectures if a multicore chip exports trace information in various trace protocols

POWER TRACE SERIAL was designed for two use cases:

- Recording trace information from high-speed Aurora-based trace ports
- Recording trace information from PCIe-based trace ports

  Tracing from a PCIe-based trace port requires a *License for PCI Express* programmed to POWER TRACE SERIAL





# Software Installation

This chapter describes the installation of TRACE32 under:

- MS Windows
  - Quick Installation
  - Ethernet
  - USB
- PC_Linux
  - Ethernet
  - USB
- Mac OS
- SunOS, Solaris (SUN), HP-UX

## Quick Installation

1. Insert the installation DVD into the DVD drive.
2. Install TRACE32 by double-clicking “setup.bat” or “files\bin\setup64\setup.exe” (WIN2000/XP/WIN Server 2003/WIN Vista/WIN7/WIN8/WIN10).
3. Follow the on-screen instructions.
4. Upon completion of the installation, start TRACE32 via the Windows **Start** button as described in **“ICD Tutorial”** (icd_tutorial.pdf).

In multicore/multiprocessor debug environments, it is recommended that Windows users start TRACE32 via the **T32Start** application.

1. Start **T32Start** via the Windows **Start** button.
2. Configure **T32Start** according to your requirements. See chapter **“Quick Start”** (app_t32start.pdf).
3. Configure the TRACE32 help system with a few mouse-clicks to display the PDF help files in your favorite PDF viewer; see **“Configure the Help System”** (ide_user.pdf).

## Ethernet

First a new node must be created for TRACE32. The Ethernet address of the emulator is on a sticker located on the reverse side of the system. The administrator must add an entry containing the IP address and node name to the name server, or the following line must be added to the file HOSTS:

```
192.9.200.5    t32
```

**Note, the above used INTERNET address is an example only**. Contact your network administrator for a new INTERNET address for TRACE32.

The INTERNET address is requested by a DHCP/RARP protocol by TRACE32. If no DHCP/RARP server is running, the address for the first connect must be set in the host table. After the first successful connect the INTERNET address is stored in the non-volatile memory within TRACE32. The following command sets the host translation table:

```
arp -s t32 0-c0-8a-0-0-0
```

| NOTE: | **On Windows the ARP command is only available if you are logged in as an administrator.** |
|---|---|

If the ARP command is not available, the internet address must be set by connecting the system via fiber optic link or parallel interface or USB.

To use the network access, the net driver must be activated. The node name can be changed, when not identical to 't32'.

**Configuration Command:**

| | |
|---|---|
| PBI=<br>NET | Used for controller-based debugging |
| NODE=<*node_name*> | (default: t32) |
| PACKLEN=1024 | Limits the size of the UDP packages to 1024 |

## USB Interface

The USB driver must be selected. Windows 2000 / XP or Vista or Windows 7 or Windows 8 or Windows 10 is required.

When the device is first connected to the system, the hardware assistant detects a new USB device and asks for a driver directory.

If the TRACE32 software is already installed, the required file (t32usb.inf) can be found in the TRACE32 installation directory (e.g. c:\t32\). Otherwise please insert the TRACE32 installation DVD and navigate to the directory ~~/bin/windows/drivers or let the system search for it.

**Configuration Command:**

| | |
|---|---|
| PBI=<br>USB | Select PODBUS interface via USB connection |
| PBI=<br>USB<br>NODE=T32-ARM | Select PODBUS interface via named USB connection<br><br>The PODBUS interface is identified by a name (**IFCONFIG**). A name is required if several debug modules are connected via USB and used simultaneously.<br>The manufacturing default device name is the serial number of the debug module. e.g. `NODE=E18110012345` |

## Quick Installation

Since November 2012, the TRACE32 PowerView GUI for Linux is available in two versions:

- **Qt** GUI (MWI and MDI) (executables **with** the suffix "-qt" : e.g. t32marm-**qt**)
- **Motif** GUI (MWI) (executables **without** the suffix "-qt" : e.g. t32marm)

### Common steps

In the following **example** the **directory** **`/opt/t32`** is used as the system directory.

The system directory is created by the following commands:

```
mkdir /opt/t32                          # or similar
```

The files are extracted from the CD to the system directory with the following commands:

```
mount /mnt/cdrom                    # or similar
cd /opt/t32
cp -r /mnt/cdrom/files/*         ./
chmod -R u+w *
cp ./demo/practice/autostart.cmm ./
mv bin/pc_linux64/config.t32     ./  # not necessary if the TRACE32
                                     # executable is called with
                                     # configuration filename parameter
               # e.g. t32marm-qt -c /opt/t32/bin/pc_linux64/config.t32

/mnt/cdrom/files/bin/pc_linux64/filecvt ./
                                    # converts all filenames to lower
                                    # case and files into UNIX format and
                                    # uncompresses all files if necessary
```

The following environment variables must be set (e.g. in .bashrc for the BASH-shell):

```
export T32SYS=/opt/t32
export T32TMP=/tmp
export T32ID=T32
```

The **TRACE32 online help system** uses an external PDF viewer for displaying the information in PDF format.

Please execute the TRACE32 command **SETUP.PDFViewer.state** inside the TRACE32 PowerView GUI once.
If the autodetection fails, a manual setting will be necessary.

**Legacy information for Acrobat Reader usage:**

Download Acrobat Reader from http://www.adobe.com and install it if not already installed on the system. Usually, you have to be root for the installation!

```
tar -xvzf linux-508.tar.gz              # or similar filename
./INSTALL                               # run the install script
```

Set the environment variable “ACROBAT_PATH” to the Acrobat installation path::

```
export ADOBE_PATH=/opt/Adobe/Reader8    # added in ~/.bashrc for BASH
or
export ACROBAT_PATH=/opt/Acrobat5       # added in ~/.bashrc for BASH
```

**Copy the TRACE32 plug-in into the Acrobat plug_ins folder (without new line)**:

```
cp /mnt/cdrom/files/bin/pc_linux/trace32.api
                                  $ADOBE_PATH/Reader/intellinux/plug_ins
or
cp /mnt/cdrom/files/bin/pc_linux/trace32.api
                                   $ACROBAT_PATH/Reader/intellinux/plug_ins
```

Verify that you have write permission to the system directory and prepare the configuration file **`config.t32`**:

```
cd /opt/t32/bin/pc_linux64  # depends on the location of the actual used
#  or                       # configuration file
cd /opt/t32                 # default file location is /opt/t32 ($T32SYS)

vi config.t32               # define interface type, …

…

                           # e.g. when using ethernet interface
PBI=                       #
NET                        # please replace t32 with the actual assigned
NODE=t32                   # network node name for the ICD module

                           # e.g. when using USB interface
PBI=
USB
```

Uncompress the executable files before usage (not necessary when filecvt was used before):

```
cd /opt/t32/bin/pc_linux64
gzip -d t32m*.gz               # or   gunzip t32m*.gz
```

Include the executable file in the PATH variable:

```
export PATH=$PATH:/opt/t32/bin/pc_linux64  # added in ~/.bashrc for BASH
                                           # preferred solution
```

Starting the TRACE32 executable file could be done in several ways:

```
                                    # preferred solutions
export PATH=$PATH:/opt/t32/bin/pc_linux64  # added in ~/.bashrc for BASH
./t32marm-qt -c /opt/t32/bin/pc_linux64/config.t32
                                    # TRACE32 executable is called with
                                    # configuration filename parameter


                                    # starting executable with a
                                    # PRACTICE startup script file
./t32marm-qt -c /opt/t32/bin/pc_linux64/config.t32
```

Configure the TRACE32 help system with a few mouse-clicks to display the PDF help files in your favorite PDF viewer; see **"Configure the Help System"** (ide_user.pdf).

### Qt GUI specific steps

The minimum requirements for the Qt GUI are:

- Kernel: 2.6.32
- libc: 2.11.1
- Qt libs: 4.6.2

Minimum versions of some popular Linux distributions:

| Distribution | minimum release | required packages |
|---|---|---|
| Ubuntu | 10.04 | libqtcore4, libqtgui4 |
| Debian | 6.0 | libqtcore4, libqtgui4 |
| Mint | 9.0 | libqtcore4, libqtgui4 |
| RedHat | RHEL 6.1 | qt, qt-x11 |
| CentOS | 6.0 | qt, qt-x11 |

| | | |
|---|---|---|
| Fedora | 13 | qt, qt-x11 |
| SUSE | 11-SP1 | libqt4 |
| openSUSE | 11.3 | libqt4 |

**Font settings in the configuration file config.t32:**

No special font settings are required. Each installed fixed width font can be used. The default font is Courier..

```
SCREEN=
FONT=Liberation Mono     ; Selects font Liberation Mono for data output
FONT=NOANTIALIAS         ; disables font aliasing (default: ANTIALIAS)
```

**GUI Configuration in the configuration file config.t32:**

The GUI can be configured with STYLE options in the **SCREEN=** section of the configuration file **config.t32**. The following STYLE options can be set:

| | |
|---|---|
| **STYLE=STATUSBAR ON** | Enables the status bar of the main window (default) |
| **STYLE=STATUSBAR OFF** | Disables the status bar of the main window |
| **STYLE=TOOLBAR ON** | Enables the tool bar of the main window (default) |
| **STYLE=TOOLBAR OFF** | Disables the tool bar of the main window |
| **STYLE=TOOLBAR TOP** | Places the tool bar on the top edge of the main window (default) |
| **STYLE=TOOLBAR RIGHT** | Places the tool bar on the right edge of the main window |
| **STYLE=TOOLBAR BOTTOM** | Places the tool bar on the bottom edge of the main window |
| **STYLE=TOOLBAR LEFT** | Places the tool bar on the left edge of the main window |
| **STYLE=COMMANDLINE TOP** | Places the command line and the soft keys to the top edge of the main window (default) |
| **STYLE=COMMANDLINE BOTTOM** | Places the command line and the soft keys to the bottom edge of the main window |
| **STYLE=MDISCROLL OFF** | disables MDI area scroll bars (default) |
| **STYLE=MDISCROLL ON** | enables MDI area scroll bars |
| **STYLE=SMALLSIZE** | Reduces the size of dialog elements (e.g. buttons, check boxes, ...) on some systems (Ubuntu Unity, ...) |

| | |
|---|---|
| **STYLE=NORMALSIZE** | Selects normal size of dialog elements (e.g. buttons, check boxes, ...) on some systems (Ubuntu Unity, ...) (default) |
| **STYLE=PLASTIQUE**[1] | Selects Qt predefined theme Plastique |
| **STYLE=CLEANLOOKS**[1] | Selects Qt predefined theme Cleanlooks |
| **STYLE=WINDOWS**[1] | Selects Qt predefined classic Windows theme |
| **STYLE=CDE**[1] | Selects Qt predefined CDE theme |
| **STYLE=MOTIF**[1] | Selects Qt predefined Motif theme |
| **DPI=AUTO (**default)<br>Linux only | TRACE32 uses the OS DPI rate for font scaling. |
| **DPI=***<value>*<br>Linux only | User-defined DPI rate for font scaling.<br>Range: `48` to `448` |
| **DPI=NONE**<br>Linux only | No DPI scaling. |
| **PALETTE** ***<n>*** **=**<br>***<red><green><blue>***[2] | Change color value, the intensities will vary from **0 to 255 for Qt**.<br>**For Motif** are the valid color values **0 to 65535**.<br><n> is the object type displayed in a **SETUP.COLOR** window. |

1: If no predefined theme is set, the theme of the current desktop is used.

2: If no special color is set, the TRACE32 default is used.

Prepare and install the fonts:

**Since TRACE32 software release April 2010 the font installation is simplified.**
It's necessary to place a subdirectory named fonts (e.g. /opt/t32/fonts) under the TRACE32 system directory (e.g. /opt/t32). The TRACE32 PowerView software automatically searches for the required TRACE32 fonts in this directory if the fonts are not provided by the host operating system.

When bitmap fonts are blocked/locked from the host operating system, a usage overwrite can be activated by adding the following lines inside the actual used TRACE configuration file e.g. config.t32.

```
SCREEN=                     ; bitcoded values (0..3 allowed)
FONTMODE=3                  ; bit0: bitmap system fonts activated
                            ; bit1: bitmap TRACE32 client fonts activated
```

Font installation for TRACE32 software releases older than April 2010:

```
cd /opt/t32/fonts
mkfontdir ./
                                  # must be done under the original
xset +fp /opt/t32/fonts           # Xserver user (normally not as root)
xset fp rehash                    # only temporary adding of TRACE32
                                  # font directory or

chkfontpath -a /opt/t32/fonts     # permanent adding of the fontdirectory
                                  # not available under SUSE distribution

ln -s /opt/t32/fonts              # permanent adding of the fontdirectory
/etc/X11/fontpath.d/t32-fonts     # available under FEDORA distribution
```

The xset commands add the TRACE32 fonts only temporary. After the next booting or logout the setting will be lost. A solution could be adding the xset commands into the login script of the actual shell from the actual user e.g. ~/.bashrc for BASH shell or just in the batch script for starting TRACE32.

The TRACE32 fonts can be added alternatively to an existing font server configuration.
e.g. add path /opt/t32/fonts to the catalog entry inside the font server configuration file /etc/X11/fs/config.

Or add the TRACE32 fonts permanent with an administration tool.
e.g. under **SUSE**: N -> Control Center -> System Administration -> Font Installer

When the 'TRACE32 menu or softkey text are displayed as graphic characters under **Fedora Core Linux** versions, some fonts are missing. Install them with:

```
yum install xorg-x11-fonts-ISO8859-1-75dpi
```

Before the installation a new node must be created. The Ethernet address of the system is placed on the bottom side of the system. The following line must be added to the file
/etc/hosts:

```
192.168.0.5    t32
```

Note that the INTERNET address given here is an example only. Contact your network administrator for a new INTERNET address for TRACE32.

The Ethernet address of the system must be entered in the file /etc/ethers (not common - only when using a RARP server):

```
0:c0:8a:0:0:0   t32
```

The INTERNET address is requested by a RARP protocol by TRACE32. If no RARP server is running, the address for the first connect must be set in the host table. After the first successful connect the INTERNET address is stored in nonvolatile memory within TRACE32. The following command sets the host translation table:

```
arp -s t32 0:c0:8a:0:0:0
```

This command must be executed **immediately before** the first startup of the emulator. It is not required for future startups because the INTERNET address is stored in the emulator. The arp cache table should be checked just before the first startup with the command
'arp -a'.

**NOTE:** A ping will only work after the TRACE32 software was booted once and the new IP address was stored automatically during boot phase into the flash of the TRACE32 modul.

The net driver must be activated. The node name can be changed, when not identical to 't32'.

Configuration Command:

PBI=
NET

| | |
|---|---|
| NODE=<*node_name*> | Node name of TRACE32 (default: t32) |
| POOL=<*node_name*>, … | Define a set of nodes, which are scanned for connection. |

In addition to the generic requirements, USB needs:

kernel >= 2.4 for FullSpeed USB support (12 MBit/s)

kernel >= 2.4.22 for HighSpeed USB support (480 MBit/s)

udev filesystem requires kernel >= 2.6
or
usbdevfs mounted on /proc/bus/usb and hotplug package

**UDEV method (kernel >= 2.6):**

The newer udev file system support needs a special rule file for TRACE32 USB devices inside the directory /etc/udev/rules.d/.

```
su
cp bin/pc_linux64/udev.conf/kernel_starting_2.6.32/10-lauterbach.rules /etc/udev/rules.d
```

**Legacy support of hotplug method (devfs):**

The hotplug package is no strict requirement, but highly recommended, if you want to avoid running the TRACE32 executables as root all the time.

To enable proper TRACE32 hotplugging, change to the directory on the CD (or with an extracted update) with the Linux executables and issue the following commands in a shell:

```
su
grep -iq trace32 /etc/hotplug/usb.usermap || cat usb.usermap.trace32
                                               >>/etc/hotplug/usb.usermap
install -m 0755 trace32 /etc/hotplug/usb/
exit
```

You can verify proper operation with the t32usbchecker tool coming with the CD or update.

The USB driver must be activated. The minimum settings in the configuration file **config.t32** are:

```
;Configuration Command:

PBI=
USB
```

**NOTE:** USB can only be used with the host-based executables (name matches t32m*), NOT with t32cde*.

## Prerequisites

The TRACE32 debug software for the Mac requires OS X 10.7 or newer.

## Installation of the TRACE32 Software

In the following example the directory **/opt/t32** is used as the system directory.

### 1. Create the system directories

Open a terminal and create the system directories with the following commands:

```
mkdir ~/t32                          # or similar
mkdir ~/t32/bin
```

### 2. Copy the files

The files are copied from the CD to the system directory with the following commands:

```
# The CD will be mounted in /Volumes/TRACE_<Rel_Tag> e.g.
# /Volumes/TRACE_201302 for the Release R.2013.02
cd ~/t32
cp -r /Volumes/TRACE_201302/files/*            .
cp -r /Volumes/TRACE_201302/files/bin/macosx64 ./bin
chmod -R u+w *
cp ./demo/practice/autostart.cmm  .
mv bin/macosx64/config.t32        . # not necessary if the TRACE32
                                    # executable is called with
                                    # configuration filename parameter
                        # e.g. t32marm-qt -c ~/t32/bin/macosx64/config.t32
```

## 3. Set up environment

The following environment variables must be set (e.g. in .bashrc for the BASH-shell):

```
export T32SYS=~/t32
export T32TMP=/tmp              # or similar
export T32ID=T32
```

Include the executable file in the PATH variable:

```
export PATH=$PATH:~/t32/bin/macosx64  # added in ~/.bashrc for BASH
```

## 4. Configure TRACE32

Verify that you have write permission to the system directory (set with environment variable T32SYS) and edit the configuration file `config.t32`.

**Interface setting:**

Ethernet interface

For the adaptation to ethernet a new node must be created. The following line must be added to the file /etc/hosts:

```
192.168.0.5    t32
```

Note that the IP address given here is an example only. Contact the network administrator for a new IP address for TRACE32. Add the following lines to your `config.t32` file:

```
PBI=
NET                          # please replace t32 with the actual
NODE=t32                     # assigned nodename for the ICD modul
```

USB interface

Add the following lines to your `config.t32` file

```
PBI=
USB
```

**Fontsettings :**

No special font settings are required. Each installed fixed width font can be used. The default font is Courier..

```
SCREEN=
FONT=Liberation Mono     ; Selects font Liberation Mono for data output
FONT=NOANTIALIAS         ; disables font aliasing (default: ANTIALIAS)
```

**GUI Configuration:**

The GUI can be configured with STYLE options in the SCREEN section of the configuration file. The following STYLE options can be set:

| | |
|---|---|
| **STYLE=STATUSBAR ON** | Enables the status bar of the main window (default) |
| **STYLE=STATUSBAR OFF** | Disables the status bar of the main window |
| **STYLE=TOOLBAR ON** | Enables the tool bar of the main window (default) |
| **STYLE=TOOLBAR OFF** | Disables the tool bar of the main window |
| **STYLE=TOOLBAR TOP** | Places the tool bar on the top edge of the main window (default) |
| **STYLE=TOOLBAR RIGHT** | Places the tool bar on the right edge of the main window |
| **STYLE=TOOLBAR BOTTOM** | Places the tool bar on the bottom edge of the main window |
| **STYLE=TOOLBAR LEFT** | Places the tool bar on the left edge of the main window |
| **STYLE=COMMANDLINE TOP** | Places the command line and the soft keys to the top edge of the main window (default) |
| **STYLE=COMMANDLINE BOTTOM** | Places the command line and the soft keys to the bottom edge of the main window |
| **STYLE=MDISCROLL OFF** | disables MDI area scroll bars (default) |
| **STYLE=MDISCROLL ON** | enables MDI area scroll bars |

Please be aware that only network interfaces are supported (not USB).

## Installation of the TRACE32 Debugger Software

In the following example the directory **/home/t32** is used as the system directory.

The system directory is created with the following command:

```
mkdir /home/t32                          # or similar
mkdir /home/t32/bin
```

The files are extracted from the CD to the system directory with the following commands:

```
mount -F hsfs -o ro /dev/dsk/c0t6d0s2 /cdrom/trace32

                                         # or similar

cd /home/t32
cp -r /cdrom/trace32/files/*      .
chmod -R u+w *
cp ./demo/practice/autostart.cmm  .
mv bin/suns/config.t32            .      # not necessary if the TRACE32
                                         # executable is called with
                                         # configuration filename
                                         # parameter
                                         # e.g.
                                         # t32marm -c/home/t32/bin/
                                         #           suns/config.t32

                                         # converts all filenames to
/cdrom/trace32/files/bin/suns/filecvt .  # lower case and files into
                                         # UNIX
                                         # format and uncompresses all
                                         # files if necessary
```

The following environment variables must be set (e.g. in .cshrc for the C-shell):

```
setenv T32SYS /home/t32
setenv T32TMP /tmp
setenv T32ID  T32
```

Prepare and install the fonts:

```
cd /home/t32/fonts
mkfontdir .
xset +fp /home/t32/fonts
xset fp rehash
```

The xset commands add the TRACE32 fonts only temporary. After the next booting or logout the setting will be lost. A solution could be adding the xset commands into the login script of the actual shell from the actual user e.g. ~/.bashrc for BASH shell or just in the batch script for starting TRACE32.

The TRACE32 fonts can be added alternatively to an existing fontserver configuration.

The TRACE32 online help uses the Adobe Acrobat Reader for displaying the information in PDF format. Download Acrobat Reader from http://www.adobe.com and install it if not already installed on the system. Usually, you have to be root for the installation!

```
gzip -d sol-508.tar.gz       # or similar filename
tar -xvf sol-508.tar         # run the install script
./INSTALL
```

Set the environment variable “ACROBAT_PATH” to the Acrobat installation path::

```
setenv ACROBAT_PATH /opt/Acrobat5     # added in ~/.cshrc for C-shell
```

Copy the TRACE32 plug-in in the Acrobat plug_ins folder (without newline):

```
cp /cdrom/files/bin/suns/trace32.api
                                $ACROBAT_PATH/Reader/sparcsolaris/plug_ins
```

Verify that you have write permission to the system directory and prepare the configuration file **config.t32**:

```
cd /home/t32/files/bin/suns      # depends on the location of the
                                 # actual used configuration file
#  or                            # the default file location
cd /home/t32                     # is /home/t32 (==$T32SYS)
vi config.t32

…

PBI=
NET                              # please replace t32 with the actual
NODE=t32                         # assigned node name for the ICD modul
```

Uncompress the executable files before usage (not necessary when filecvt was used before):

```
cd /home/t32/bin/suns
gzip -d t32m*.gz            # or   gunzip t32m*.gz
```

Include the executable file in the PATH variable:

```
setenv PATH $PATH:/home/t32/bin/suns     # added in ~/.cshrc for C-shell
                                         # preferred solution
```

## Preparations for the Ethernet Interface

Before the installation a new node must be created. The Ethernet address of the system is placed on the bottom side of the system. The following line must be added to the file
/etc/hosts:

```
192.168.0.5    t32
```

Note that the INTERNET address given here is an example only. Contact your network administrator for a new INTERNET address for TRACE32. The Ethernet address of the system must be entered in the file /etc/ethers:

```
0:c0:8a:0:0:0   t32
```

The INTERNET address is requested by a RARP protocol by TRACE32. If no RARP server is running, the address for the first connect must be set in the host table. After the first successful connect the INTERNET address is stored in nonvolatile memory within TRACE32. The following command sets the host translation table:

```
arp -s t32 0:c0:8a:0:0:0
```

This command must be executed **immediately before** the first startup of the emulator. It is not required for future startups because the INTERNET address is stored in the emulator. The arp cache table should be checked just before the first startup with the command
'arp -a'.

**NOTE:** A ping will only work after the TRACE32 software was booted once and the new IP address was stored automatically during boot phase into the flash of the TRACE32 modul.

The net driver must be activated. The node name can be changed, when not identical to 't32'.

```
;Configuration commands in the configuration file config.t32:

PBI=
NET

NODE=<node_name>              ;Node name of TRACE32 (default: t32)

POOL=<node_name>, …           ;Define a set of nodes, which are scanned
                              for connection.
```

**If you can not solve your problem with the following hints contact our support line:**

telephone: ++49 8102/9876-555
facsimile: ++49 8102/9876-999
e-mail: support@lauterbach.com

**System doesn't response to ping on Ethernet**

Internet address already setup in system, or arp used?
When arp is used, it must be used on the same workstation short before.
Ethernet address correct?
System on the correct subnet?
Cables and transceiver o.k.?
Ethernet software in host (PC) configured correctly?

**xset +fp fontpath gives error 'bad value …'**

Does the font directory exist?
Does the fonts.dir file exist (created by mkfontdir)?
Is the directory seen under the same name by the X-server?
Have all directories that lead to the font directory read and execute permissions for everybody?

**Executable program does not start or gives fatal error**

When transferring between different OS-systems, files copied in binary mode?
Access rights to file in directory o.k.?
Configuration file contents o.k.?

**Executable program displays 'FATAL ERROR selecting device-driver …**

Using configuration file for MS-DOS for the WINDOWS-Driver?
WINDOWS and workstation drivers cannot load new drivers.
Environment variable 'T32CONFIG' and/or 'T32SYS' correctly set?

**Executable program displays 'error reading config.t32:'**

Configuration file contents o.k.?
Commands in file in uppercase?
Blanks inserted/not inserted?
Device specific commands placed after device header?
Device configuration blocks separated by empty lines?
Environment variable 'T32CONFIG' and/or 'T32SYS' correctly set?

**Executable program stops without message, but with window opened**

Access rights to directory o.k.?
On UNIX host, try with 'NOLOCK' feature.
When using the RS232 interface: Is a login process active on the tty?

**Program stops with message 'font xxxx not found'**

Do fonts appear in the 'xlsfonts' command?

Can one font (e.g. t32-lsys-16) be displayed by 'xfd -fn t32-lsys-16'?

Fonts added to X-Windows FONTPATH?

Fonts converted, when required, and .bdf files removed?

Command to generate font directory executed with correct parameters?

Fonts installed on the X-Windows server, not client?

If using an X-Terminal, use the conversion programs for the X-Terminal?

**Executable program displays 'boot.t32 not found'**

Access rights to directory o.k.?

Read and write access to boot.t32 (write required on UNIX without NOLOCK)?

Configuration file contents o.k.?

Environment variable 'T32SYS' correctly set?

**Executable program stops after displaying 'error reading boot.t32'**

When transferring between different OS-systems, files copied in binary mode?
Access rights granted?
Try again after switching off the TRACE32 system?

**Executable program stops after displaying 'booting …' or 'finished.'**

When transferring between different OS-systems, files copied in binary mode?
Packet size set correctly on Ethernet, handshake set when required?

**Bootloader stops with message “fatal error …'”**

When transferring between different OS-systems, files copied in binary mode?
Mixing different versions of the software, e.g. MCC.T32 and MCCxxx.t32?

**Bootloader displays “cannot save image …”**

Write access right on system directory?
Disk full?
Existing read-only file?

**Software crashes or stops after booting is finished**

Boot image file maybe destroyed, remove all boot0x.t32 files?
Connection of modules o.k., connector bend?

**Software doesn't work stable**

Boot image file maybe destroyed, remove all boot0x.t32 files?
Connection of modules o.k., connector bend?
Check connection of Fibre Optic, Ethernet or Parallel interface.
On Ethernet try with smaller packet size and/or handshake.

**Emulation system doesn't work correctly**

Check Emulation Probe Manual in “Targets” part of the manual.

**Parallel Port not working stable**

Check that the port is on the correct mode. Choose either EPP 1.9 or compatible mode. The mode selection can usually be done in the BIOS setup (can be activated during booting).

**USB debugger not detected at all by LINUX**

There are a few reasons why this can happen:

- the running kernel does not support USB yet
- USB not enabled during kernel configuration
- USB enabled as modules in the kernel configuration, but module autoload did fail or isn't configured
- usbdevfs is not mounted
- usbdevfs is mounted, but not at /proc/bus/usb
- bad USB cable, use the original one or make sure it is at max. 3 meters long
  cable type lettering: 28AWG/1PR 24AWG/2C
- old debugger firmware - version V6.5 or later needed

**Menu or softkey text is displayed wrong under LINUX**

On systems which use a mixed set of 8bit and 16bit menu fonts, and have only *-iso10646-* system fonts installed, no meaningful glyphs are rendered in the menu or softkeys of TRACE32 main window.

If this happens, please install the additional iso8859 system font package(s).

```
e.g.  yum install xorg-x11-fonts-ISO8859-1-75dpi
```

**Fixed width font t32sys not found under WINDOWS**

When you start the TRACE32 executable the fonts are loaded. If a SW update will be done, which replaces the TRACE32 font file named t32font.fon, the new fonts will not be activated as long as the old fonts are loaded.

This happens even if both font files are identical.

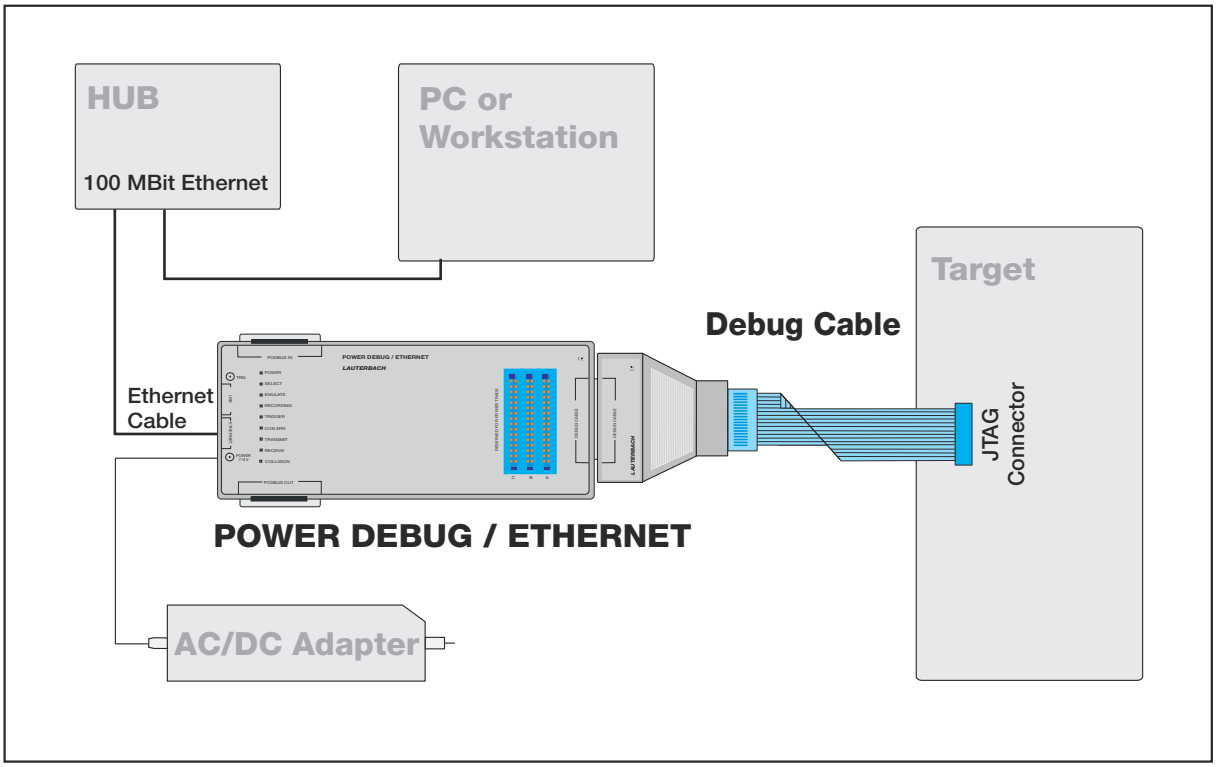
Please reboot your Windows PC to solve this issue.

# FAQ

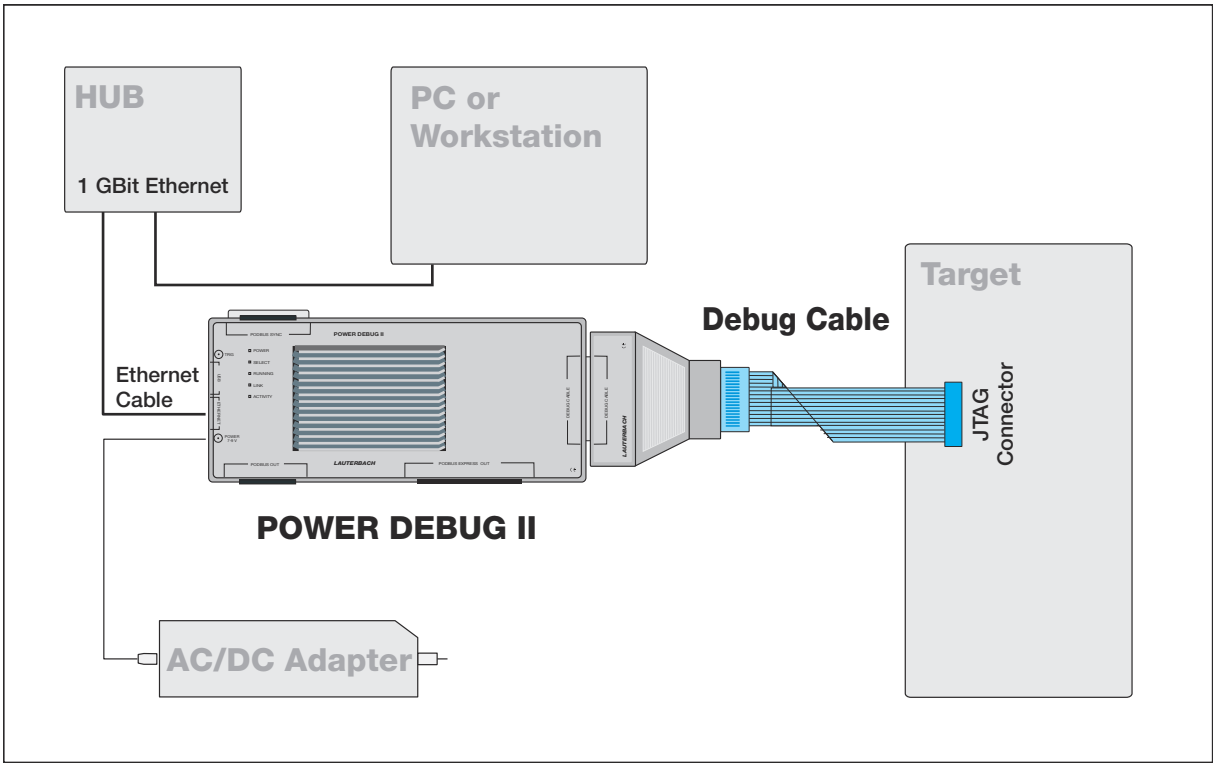
Please refer to our Frequently Asked Questions page on the Lauterbach website.

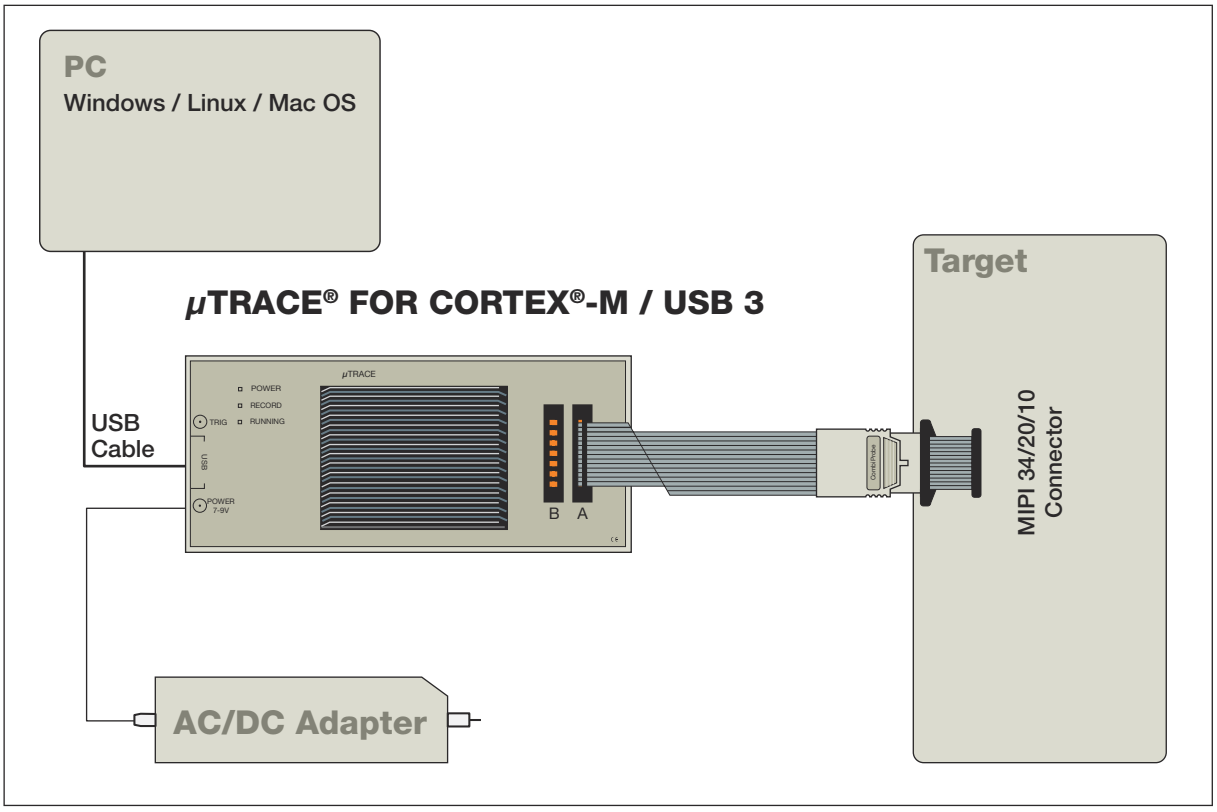
## POWER DEBUG / ETHERNET with Debug Cable

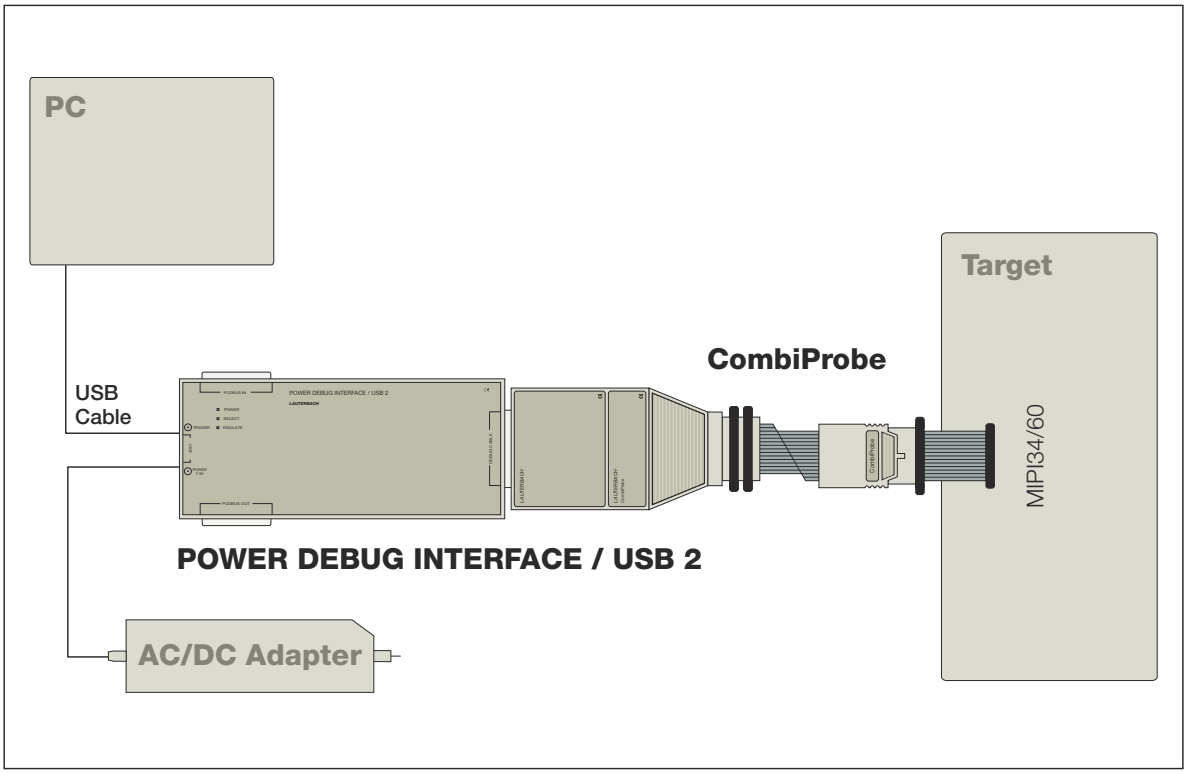
100 MBit ethernet or USB 2.x interface to host computer



GBit ethernet or USB 2.x interface to host computer

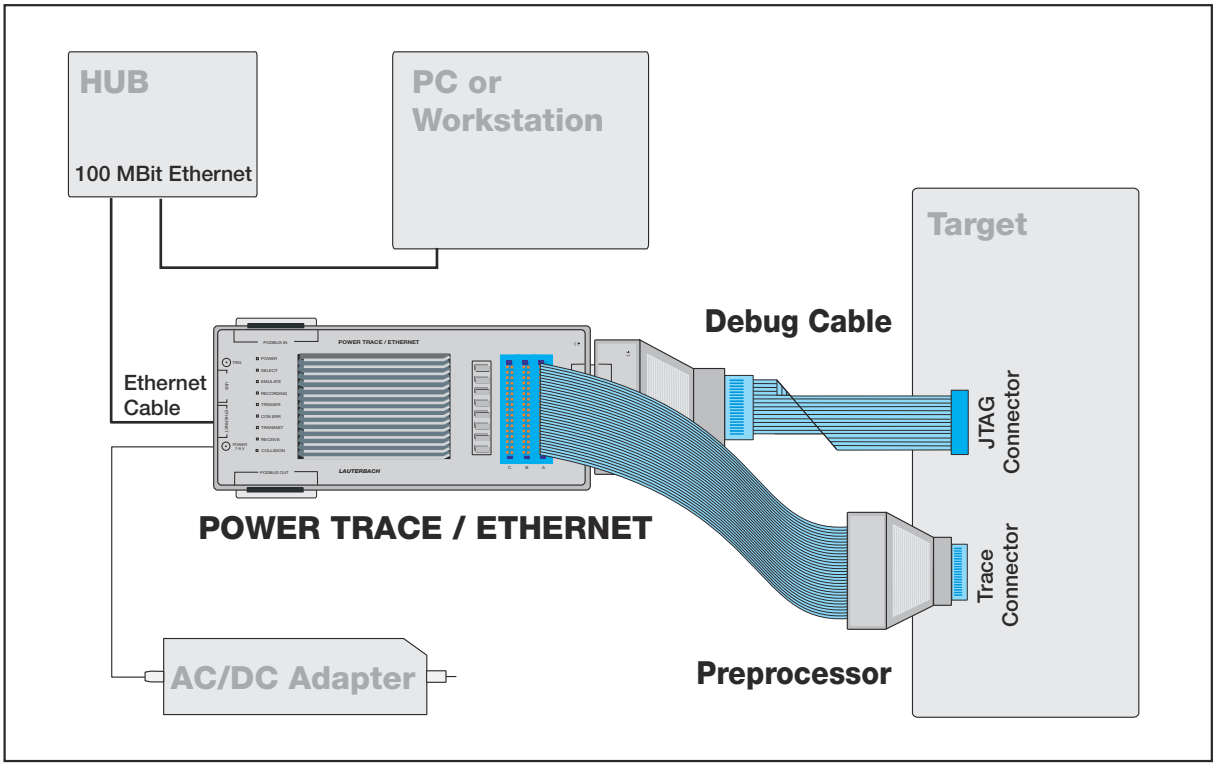




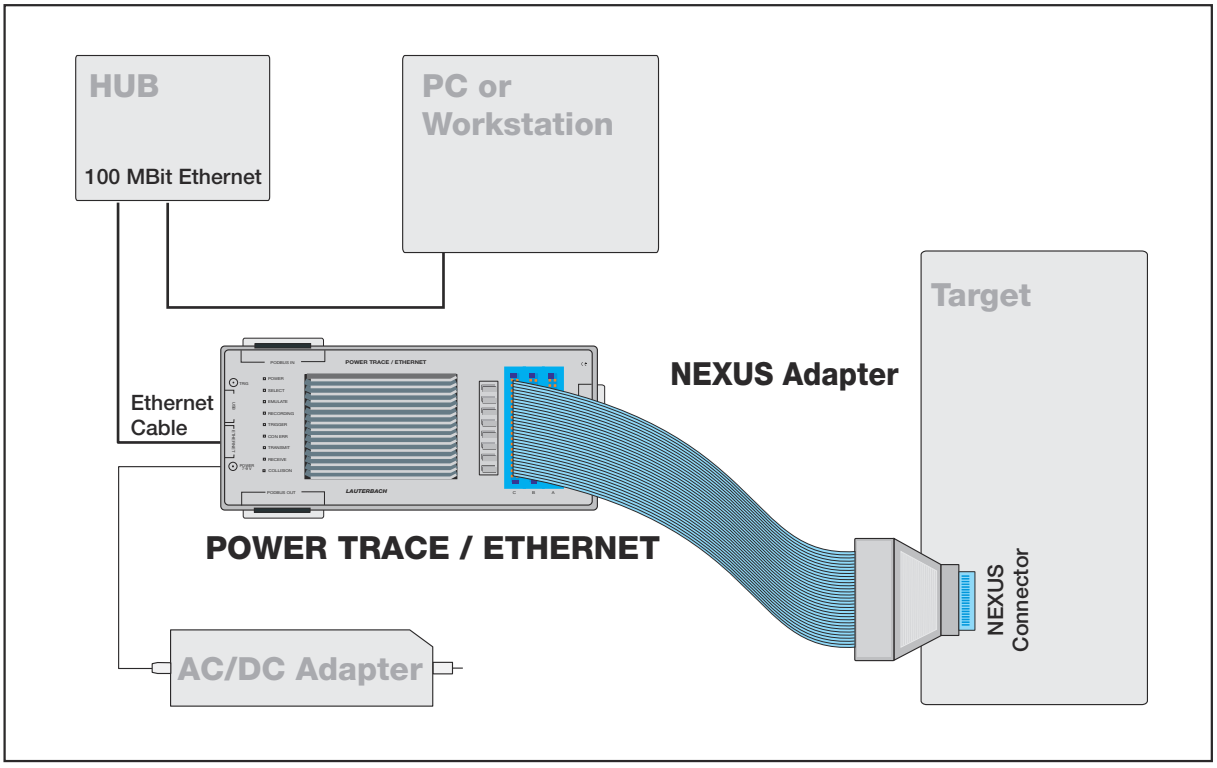


100 MBit ethernet or USB 2.x interface to host computer

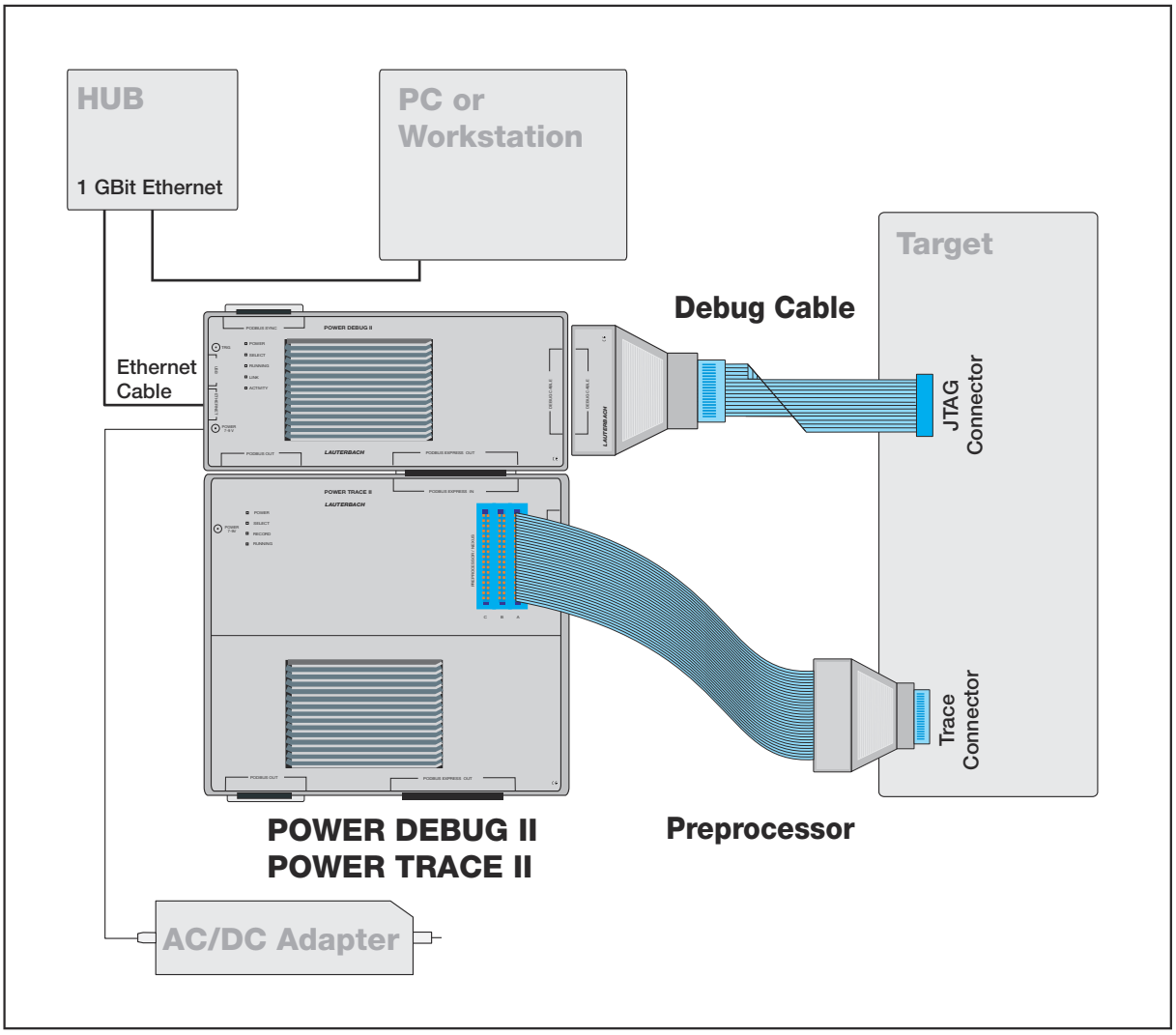
All-in-one debug and trace module with 256 MByte or 512 MByte of trace memory



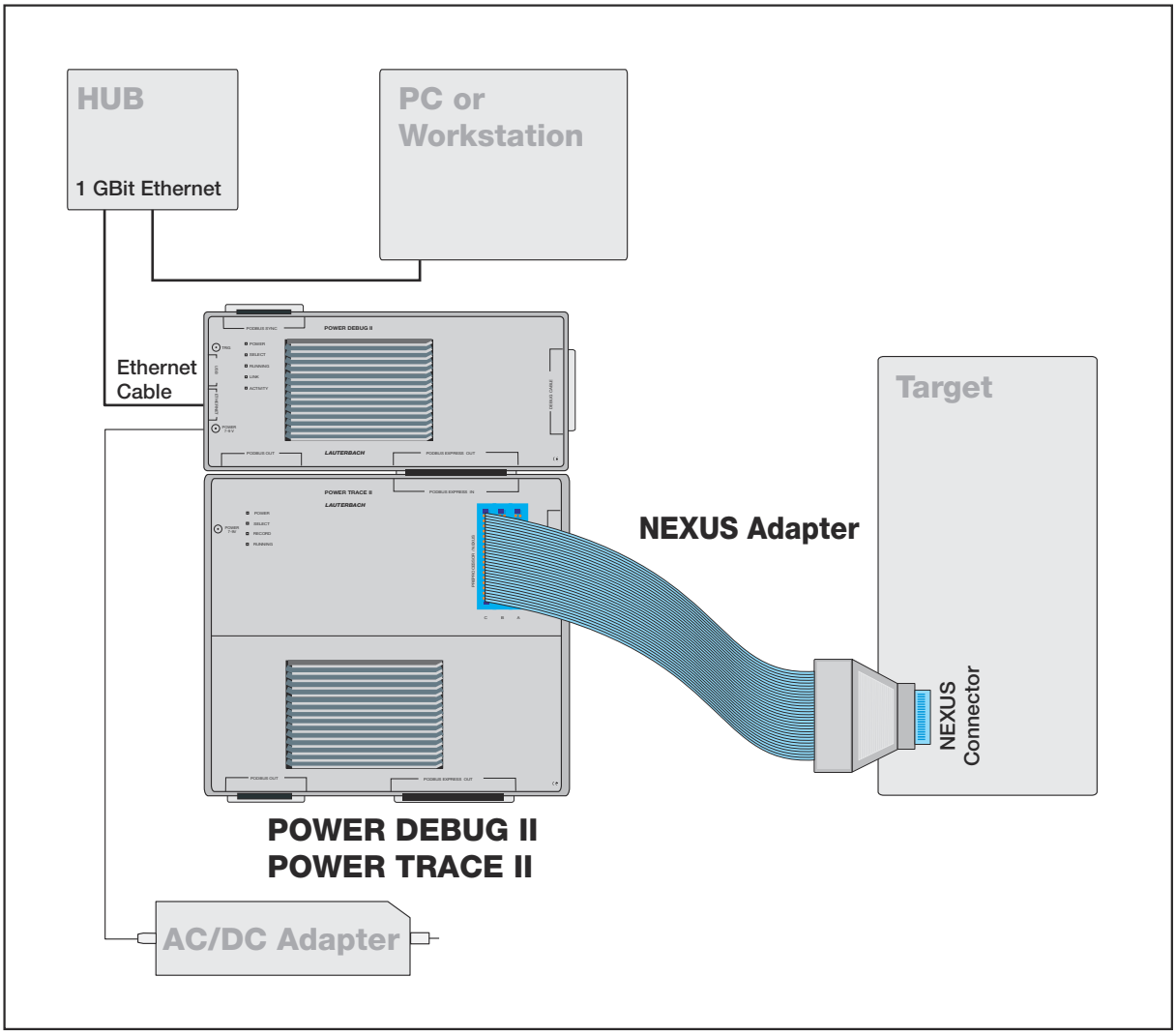
Trace memory with a depth of 256 MByte or 512 MByte.

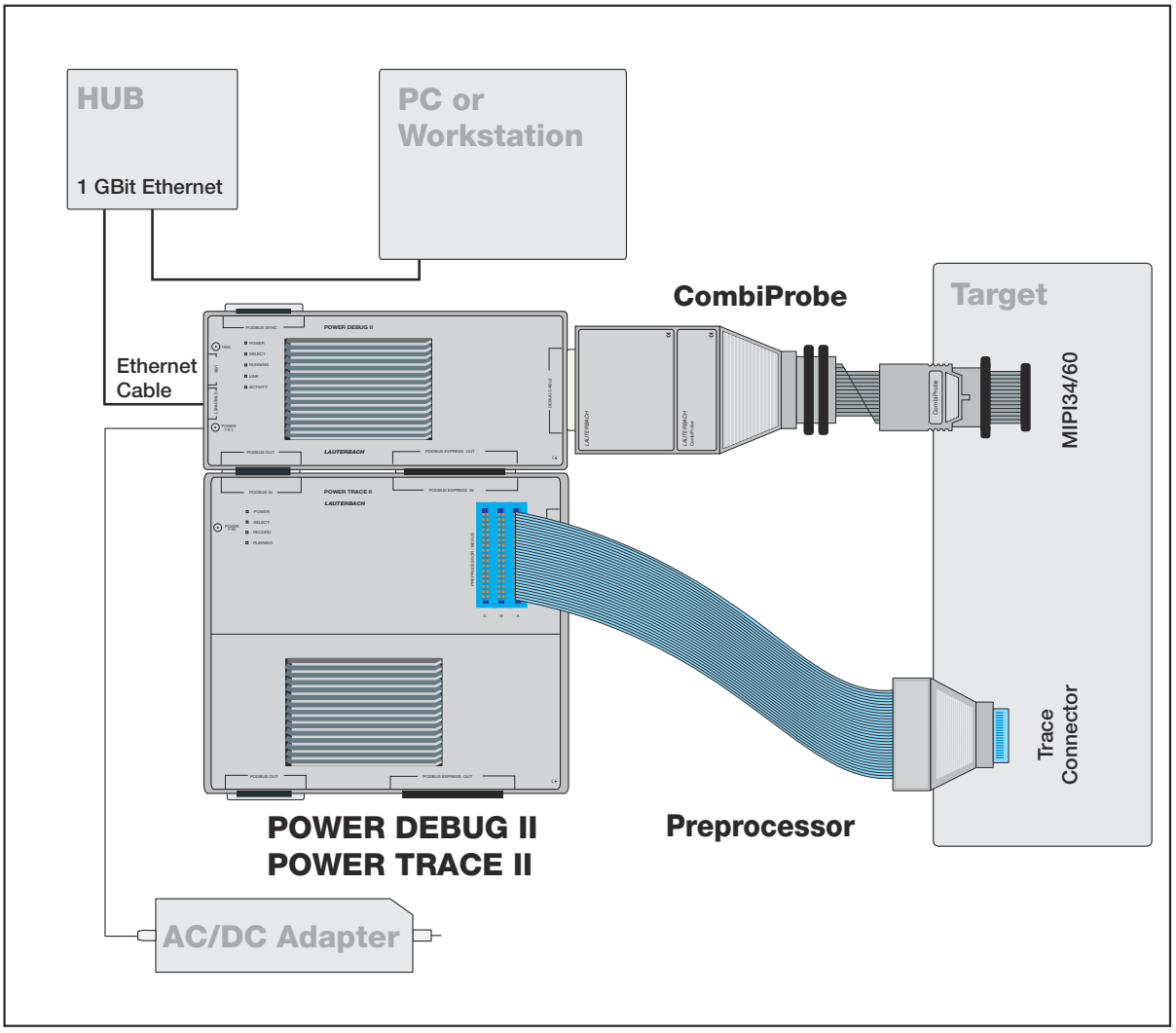


**POWER DEBUG II** can be extended by a POWER TRACE II with 1 GByte, 2 GByte or 4 GByte trace memory.

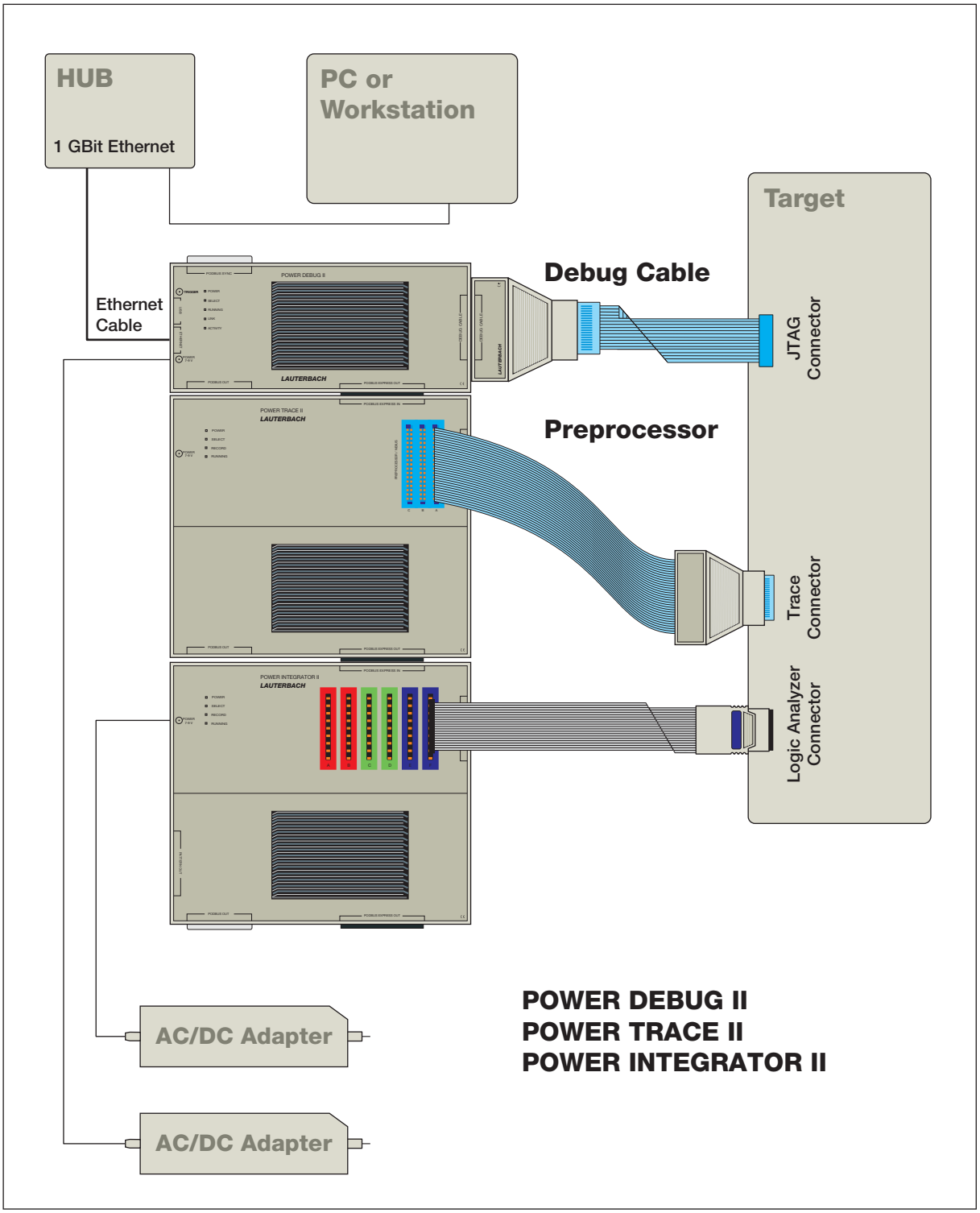


**POWER DEBUG II** can be extended by a POWER TRACE II for NEXUS with 1 GByte, 2 GByte or 4 GByte trace memory.





**POWER DEBUG II** can be extended by a POWER TRACE II with 1 GByte, 2 GByte or 4 GByte trace memory, and a POWER INTEGRATOR II Logic Analyzer with a 1 GByte, 2 GByte or 4 GByte logic analyzer memory.



**NOTE:**

- For the first two devices, only one AC/DC adapter is required.
  Each additional device requires an additional AC/DC adapter.
- An additional device in a PODBUS device chain **cannot** be damaged if it is **not** connected to its required AC/DC adapter.
- In case of a missing AC/DC adapter, an error message is displayed in the **AREA** window.
  To view the error message, choose **View** menu > **Message Area**.
  Or type **AREA** at the TRACE32 command line.

Typical configuration: