TE0955 Reference Design

Public Docs

Exported on 08/31/2026

Table of Contents

1 Overview

Versal PS Design with Linux Example. HW-Manager.
Wiki Resources page: http://trenz.org/te0955-info

1.1 Key Features

- Vitis/Vivado 2024.2
- PetaLinux
- SD
- eMMC
- ETH
- USB
- I2C
- MAC from EEPROM
- User LEDs

1.2 Revision History

Expand List

Date	Vivado	Project Built	Authors	Description
2025-06-27	2024.2	TE0955-te0955-rd_noprebuilt-vivado_2024.2-build_1_20250707150046.zip TE0955-te0955-rd-vivado_2024.2-build_1_20250707150046.zip	Markus Kirberg / Mohsen Chamanbaz	• 2024.2 release

1 Design Revision History

1.3 Release Notes and Know Issues

Issues	Description	Workaround	To be fixed version

2 Known Issues

1.4 Requirements

1.4.1 Software

Software	Version	Note
Vivado	2024.2	needed
Vitis	2024.2	needed, Vitis is included in Vivado installation
PetaLinux	2024.2	needed
Vitis HLS	2024.2	needed (used for MIPI-Camera Pipeline) Vitis HLS is included optionally in Vivado installation

3 Software

1.4.2 Hardware

Basic description of TE Board Part Files is available on TE Board Part Files[1].

Complete List is available on "<project folder>\board_files*_board_files.csv"

Design supports following modules:

Expand List

1 https://wiki.trenz.local/display/PD/TE+Board+Part+Files

Module Model	Board Part Short Name	PCB Revision Support	DDR	QSPI Flash	EMMC	Others	Notes
TE0955-01-P001*	23_1lse_8gb	REV01	8GB	256MB	32GB	NA	NA
TE0955-01-EGBE32-A	23_1lse_8gb	REV01	8GB	256MB	32GB	NA	NA

4 Hardware Modules

*used as reference

Design supports following carriers:

Carrier model	Notes
AMB0010*	Because of using this carrier board the device tree needed to be changed. To see more detail please refer to Device Tree (see page 21)
TEB0955	This carrier board is not yet available in the shop.

*used as reference

Additional HW Requirements:

Additional Hardware	Notes
Micro USB Cable for JTAG/UART	Check Carrier Board and Programmer for correct type
Cooler	It is strongly recommended that the RFSoC should be used with heat sink.
Ethernet cable	
SD card	16GB
PC	

5 Additional Hardware

1.5 Content

For general structure and of the reference design, see Project Delivery - AMD devices[2]

1.5.1 Design Sources

Type	Location	Notes
Vivado	<project folder>\block_design <project folder>\constraints <project folder>\ip_lib <project folder>\board_files	Vivado Project will be generated by TE Scripts
Vitis	<project folder>\sw_lib	Additional Software Template for Vitis and apps_list.csv with settings automatically for Vitis app generation
PetaLinux	<project folder>\os\petalinux	PetaLinux template with current configuration

6 Design sources

1.5.2 Additional Sources

Type	Location	Notes
init.sh	<project folder>\misc\sd\	Additional Initialisation Script for Linux

7 Additional design sources

1.5.3 Prebuilt

File	File-Extension	Description
BIF-File	*.bif	File with description to generate Bin-File

2 https://wiki.trenz.local/display/PD/Project+Delivery+-+AMD+devices

File	File-Extension	Description
BIN-File	*.bin	Flash Configuration File with Boot-Image (Versal-FPGAs)
BIT-File	*.pdi	FPGA Configuration File
Boot Script-File	*.scr	Distro Boot Script file
Debug Probes-File	*.ltx	Definition File for Vivado/Vivado Labtools Debugging Interface
Diverse Reports	---	Report files in different formats
Hardware-Platform-Description-File	*.xsa	Exported Vivado hardware description file for Vitis and PetaLinux
LabTools Project-File	*.lpr	Vivado Labtools Project File
OS-Image	*.ub	Image with Linux Kernel (On Petalinux optional with Devicetree and RAM-Disk)
Software-Application-File	*.elf	Software Application for Zynq or MicroBlaze Processor Systems

8 Prebuilt files (only on ZIP with prebuilt content)

1.5.4 Download

Reference Design is only usable with the specified Vivado/Vitis/PetaLinux version. Do never use different Versions of Xilinx Software for the same Project.

Reference Design is available on:

- TE0955 "Test Board" Reference Design[3]

3 https://www.trenz-electronic.de/Downloads/?path=Trenz_Electronic/Modules_and_Module_Carriers/4x5.64/TE0955/Reference_Design

2 Design Flow

> Reference Design is available with and without prebuilt files. It's recommended to use TE prebuilt files for first launch.

Trenz Electronic provides a tcl based built environment based on Xilinx Design Flow.

See also:

- AMD Development Tools#XilinxSoftware-BasicUserGuides[4]
- Vivado Projects - TE Reference Design[5]
- Project Delivery[6]

The Trenz Electronic FPGA Reference Designs are TCL-script based project. Command files for execution will be generated with "_create_win_setup.cmd" on Windows OS and "_create_linux_setup.sh" on Linux OS.

TE Scripts are only needed to generate the vivado project, all other additional steps are optional and can also executed by Xilinx Vivado/Vitis GUI. For currently Scripts limitations on Win and Linux OS see: Project Delivery Currently limitations of functionality[7]

> **Caution!** Win OS has a 260 character limit for path lengths which can affect the Vivado tools. To avoid this issue, use Virtual Drive or the shortest possible names and directory locations for the reference design (for example "x:\<project folder>")

1. _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell:

_create_win_setup.cmd/_create_linux_setup.sh

```
------------------------Set design paths-----------------------------
-- Run Design with: _create_win_setup
-- Use Design Path: <absolute project path>
---------------------------------------------------------------------
-------------------------TE Reference Design-------------------------
---------------------------------------------------------------------
-- (0)  Module selection guide, project creation...prebuilt export...
```

4 https://wiki.trenz.local/display/PD/AMD+Development+Tools
5 https://wiki.trenz.local/display/PD/Vivado+Projects+-+TE+Reference+Design
6 https://wiki.trenz.local/display/PD/Project+Delivery
7 https://wiki.trenz-electronic.de/display/PD/Project+Delivery+-+AMD+devices#ProjectDeliveryAMDdevices-Currentlylimitationsoffunctionality

```
-- (1)  Create minimum setup of CMD-Files and exit Batch
-- (2)  Create maximum setup of CMD-Files and exit Batch
-- (3)  (internal only) Dev
-- (4)  (internal only) Prod
-- (c)  Go to CMD-File Generation (Manual setup)
-- (d)  Go to Documentation (Web Documentation)
-- (g)  Install Board Files from Xilinx Board Store (beta)
-- (a)  Start design with unsupported Vivado Version (beta)
-- (x)  Exit Batch (nothing is done!)
----
Select (ex.:'0' for module selection guide):
```

2. Press 0 and enter to start "Module Selection Guide"
3. Create project and follow instructions of the product selection guide, settings file will be configured automatically during this process.
    - optional for manual changes: Select correct device and Xilinx install path on "design_basic_settings.cmd" and create Vivado project with "vivado_create_project_guimode.cmd"

> Note: Select correct one, see also Vivado Board Part Flow[8]

4. Create hardware description file (.xsa file) for PetaLinux project and export to prebuilt folder

run on Vivado TCL (Script generates design and export files into "\prebuilt\hardware\")

```
\prebuilt\hardware\")">
TE::hw_build_design -export_prebuilt
```

> Using Vivado GUI is the same, except file export to prebuilt folder.

5. Create and configure your petalinux project with exported .xsa-file, see PetaLinux KICKstart[9]
    - use TE Template from "<project folder>\os\petalinux"
    - use exported .xsa file from "<project folder>\prebuilt\hardware\<short name>" . **Note:** HW Export from Vivado GUI creates another path as default workspace.
6. Configure the **boot.scr** file as needed, see Distro Boot[10]
7. Generate Programming Files with Vitis

8 https://wiki.trenz.local/display/PD/Vivado+Board+Part+Flow
9 https://wiki.trenz.local/display/PD/PetaLinux+KICKstart
10 https://wiki.trenz.local/display/PD/Distro+Boot

a. Copy petaLinux build image files to prebuilt folder
   i. copy **u-boot.elf, system.dtb, bl31.elf**, **image.ub** and **boot.scr** from "<plnx-proj-root>/images/linux" to prebuilt folder

> "<project folder>\prebuilt\os\petalinux\<ddr size>" or "<project folder>\prebuilt\os\petalinux\<short name>"

b. Generate Programming Files

run on Vivado TCL (Script generates applications and bootable files, which are defined in "test_board\sw_lib\apps_list.csv")

```
TE::sw_run_vitis -all
TE::sw_run_vitis (optional; Start Vitis from Vivado GUI or start with TE Scripts on Vivado TCL)
```

> TCL scripts generate also platform project, this must be done manually in case GUI is used. See Vitis[11]

8. Generate Programming Files with Petalinux (alternative), see PetaLinux KICKstart[12]

11 https://wiki.trenz.local/display/DRAFT/Vitis
12 https://wiki.trenz.local/display/PD/PetaLinux+KICKstart

3 Launch

3.1 Programming

> Check Module and Carrier TRMs for proper HW configuration before you try any design.

Xilinx documentation for programming and debugging: Vivado/Vitis/SDSoC-Xilinx Software Programming and Debugging[13]

Note: Depending on Boot Mode settings, QSPI boot with Linux image on SD or complete SD Boot is possible.

3.1.1 Get prebuilt boot binaries

1. _create_win_setup.cmd/_create_linux_setup.sh and follow instructions on shell:
2. Press 0 and enter to start "Module Selection Guide"
   a. Select assembly version
   b. Validate selection
   c. Select create and open delivery binary folder

> Note: Folder "<project folder>/_binaries_<Article Name>" with subfolder "boot_<app name>" for different applications will be generated

3.1.2 QSPI-Boot mode

Option for **BOOT.bin** on QSPI Flash and **image.ub, dtbos (folder)** and **boot.scr** on **SD** or **USB**.

1. Connect **JTAG** and power on carrier with module
2. Open Vivado Project with "vivado_open_existing_project_guimode.cmd" or if not created, create with "vivado_create_project_guimode.cmd"

13 https://wiki.trenz-electronic.de/display/PD/AMD+Development+Tools#AMDDevelopmentTools-AMDSoftwareProgrammingandDebugging

run on Vivado TCL (Script programs BOOT.bin on QSPI flash)

```
TE::pr_program_flash -swapp u-boot
```

To program with Vitis/Vivado GUI, use special FSBL (fsbl_flash) on setup

3. Copy **image.ub, dtbos (folder)** and **boot.scr** on **SD** or **USB**
   - use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder,see: Get prebuilt boot binaries (see page 13)
   - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
4. Set Boot Mode to **QSPI-Boot** and insert **SD** or **USB**.

3.1.3 SD-Boot mode

1. Copy **image.ub, boot.src, dtbos (folder)** and **BOOT.bin** on **SD**
   - use files from "<project folder>_binaries_<Article Name>\boot_linux" from generated binary folder, see: Get prebuilt boot binaries (see page 13)
   - or use prebuilt file location, see "<project folder>\prebuilt\file_location.txt"
2. Set Boot Mode to SD-Boot.
3. Insert SD-Card in SD-Slot.

3.1.4 JTAG

Not used on this example.

3.2 Usage

1. Prepare HW like described on section Programming (see page 13)
2. Connect UART USB (most cases same as JTAG)
3. Select SD Card as Boot Mode (or QSPI - depending on step 1)

Note: See TRM of the Carrier, which is used.

> Starting with Petalinux version 2020.1, the industry standard "Distro-Boot" boot flow for U-Boot was introduced, which significantly expands the possibilities of the boot process and has the primary goal of making booting much more standardised and predictable. The boot options described above describe the common boot processes for this hardware; other boot options are possible.
> For more information see Distro Boot[14]

4. Power On PCB
   boot process

   1. Versal Boot ROM loads PLM from SD/QSPI into OCM,

   2. PLM init the PS, programs the PL using the bitstream and loads PMU, ATF and U-boot from SD/QSPI into DDR,

   3. U-boot loads Linux (**image.ub**) from SD/QSPI/... into DDR

3.2.1 Linux

1. Open Serial Console (e.g. putty)
   - Speed: 115200
   - select COM Port

   > Win OS, see device manager, Linux OS see dmesg |grep tty (UART is *USB1)

2. Linux Console:

   > Note: Wait until Linux boot finished

3. You can use Linux shell now.

```
i2cdetect -y -r 0   (check I2C 0 Bus)
dmesg | grep rtc    (RTC check)
udhcpc              (ETH0 check)
lsusb               (USB check)
```

4. Option Features
   - init.sh scripts

14 https://wiki.trenz.local/display/PD/Distro+Boot

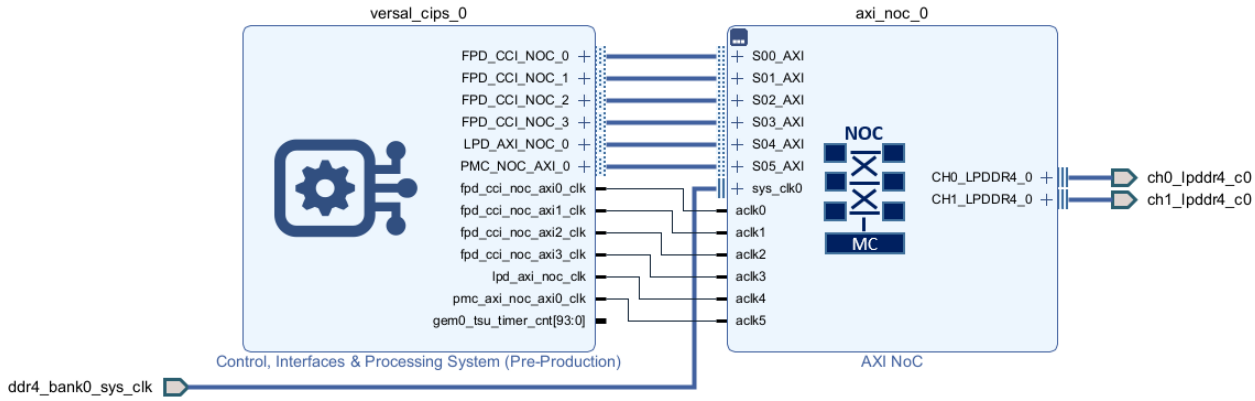
- add init.sh script on SD, content will be load automatically on startup (template included in "<project folder>\misc\SD")

3.2.2 Vivado HW Manager

4 System Design - Vivado

4.1 Block Design

4.1.1 PCB REV03



1 Block Design PCB REV03

4.1.2 PS Interfaces

Activated interfaces:

Type	Note
DDR	
QSPI	MIO
SD0/eMMC	MIO
SD1/SD3.0	MIO
PMC_I2C	MIO

Type	Note
UART1	MIO
GEM0	MIO
USB0	MIO, USB2.0

4.2 Constrains

4.2.1 Design specific constraints

_i_io.xdc

5 Software Design - Vitis

For Vitis project creation, follow instructions from:

Vitis[15]

5.1 Application

Template location: "<project folder>\sw_lib\sw_apps\"

5.1.1 versal_plm

Xilinx default PLM firmware.

5.1.2 versal_psm

Xilinx default PSM firmware.

5.1.3 hello_te0955

Hello TE0955 is a Xilinx Hello World example as endless loop instead of one console output.

5.1.4 u-boot

U-Boot.elf is generated with PetaLinux. Vitis is used to generate BOOT.bin.

15 https://wiki.trenz.local/display/PD/Vitis

6 Software Design - PetaLinux

For PetaLinux installation and project creation, follow instructions from:

- PetaLinux KICKstart[16]

6.1 Config

Start with **petalinux-config** or **petalinux-config --get-hw-description**

Changes:

- Identification
  - CONFIG_SUBSYSTEM_HOSTNAME="Trenz"
  - CONFIG_SUBSYSTEM_PRODUCT="TE0955"

6.2 U-Boot

Start with **petalinux-config -c u-boot**
Changes:

- Read MAC from EEPROM:
  - CONFIG_DM_RTC=y
    CONFIG_NVMEM=y

6.3 Fixes for BL31 (Petalinux 2023.2 Bug)

create **arm-trusted-firmware_%.bbappend** in meta-user/recipes-bsp/arm-trusted-firmware with content

meta-user/recipes-bsp/arm-trusted-firmware/arm-trusted-firmware_%.bbappend

```
ATF_CONSOLE = "pl011_1"
```

16 https://wiki.trenz-electronic.de/display/PD/PetaLinux+KICKstart

6.4 Device Tree

Due to the use of the modified AMB0010 carrier board, the device tree code for eMMC needed to be changed.

project-spec\meta-user\recipes-bsp\device-tree\files\system-user.dtsi

```
/include/ "system-conf.dtsi"

#include <dt-bindings/gpio/gpio.h>

/*------------------ eMMC --------------------*/
&sdhci0 {
    //bus-width = <8>; uncommented for testing on AMB0010 with V_IO_CFG 3.3V

    //below lines for testing on AMB0010 with V_IO_CFG 3.3V
    no-1-8-v;
    max-frequency = <25000000>;
};

/*------------------ SD --------------------*/
&sdhci1 {
    no-1-8-v;
};

/*------------------ QSPI --------------------*/
&ospi {

    bus-num = <2>;
    num-cs = <1>;

    flash@0 {
        compatible = "mt35xu02g", "micron,m25p80", "jedec,spi-nor";
        reg = <0>;

        cdns,read-delay = <0x0>;
        cdns,tshsl-ns = <0x0>;
        cdns,tsd2d-ns = <0x0>;
        cdns,tchsh-ns = <0x1>;
        cdns,tslch-ns = <0x1>;

        spi-rx-bus-width = <8>;
```

```
        spi-tx-bus-width = <8>;
        spi-max-frequency = <20000000>; //TODO: check why only 20MHz AMD also does
this....

        #address-cells = <1>;
        #size-cells = <1>;
    };
};

/*------------------ ETH PHY --------------------*/
&gem0 {
    phy-handle = <&phy0>;

    nvmem-cells = <&eth0_addr>;
    nvmem-cell-names = "mac-address";

    //required otherwise petalinux gives a static address here
    /delete-property/ local-mac-address;

    mdio {
        phy0: phy0@3 {
            device_type = "ethernet-phy";
            reg = <3>;

            //only needed because of reset-gpios present
            //compatible = "ethernet-phy-id0022.1652"; //uboot: [mii read PHYAD 2].
[mii read PHYAD 3]

            //reset-names = "ETH_RESET";
            //reset-gpios = <&gpio0 49 GPIO_ACTIVE_LOW>; //careful this is the #RST
            //reset-assert-us = <10000>; //minimum duration according to datasheet
10ms
            //reset-deassert-us = <2000>;
        };
    };
};


/*------------------ GPIO MISC --------------------*/
&gpio0 {
    gpio-line-names =
        "", "", "", "", "", "", "", "", "", "",
        "", "", "", "", "GPIO3_LS", "GPIO1_LS", "", "", "", "",
        "", "", "GPIO2_LS", "DC_OM_I2C";
};
&gpio1 {
    gpio-line-names =
        "", "", "", "", "", "", "", "", "", "",
        "", "", "", "", "", "", "", "", "", "",
        "", "", "", "", "", "", "", "", "", "",
        "", "", "", "", "", "", "", "GPIO0", "", "",
        "", "", "", "", "", "", "", "", "RST#", "",
```

```
        "", "", "", "", "", "", "", "", "", "";
};

/*------------------ USB --------------------*/
&dwc3_0 {
    dr_mode = "host";
};

/*------------------ I2C --------------------*/

&i2c2 {
    status = "okay";

    eeprom: eeprom@53 {
        compatible = "microchip,24aa025", "atmel,24c02";
        reg = <0x53>;

        #address-cells = <1>;
        #size-cells = <1>;
        eth0_addr: eth-mac-addr@FA {
            reg = <0xFA 0x06>;
        };
    };
};
```

6.5 Kernel

Start with **petalinux-config -c kernel**

Changes:

- ---

6.6 Rootfs

Start with **petalinux-config -c rootfs**

- Misc Apps:
  - CONFIG_libgpiod-tools=y
  - CONFIG_startup=y
- For additional test tools:
  - CONFIG_packagegroup-xilinx-benchmarks=y
- Dropbear instead of OpenSSH
  - CONFIG_packagegroup-core-ssh-dropbear=y
- For auto login:

- CONFIG_imagefeature-serial-autologin-root=y
- CONFIG_imagefeature-debug-tweaks=y
- CONFIG_imagefeature-empty-root-password=y
- CONFIG_ADD_EXTRA_USERS="root:root;petalinux:petalinux;"

6.7 Applications

See "<project folder>\os\petalinux\project-spec\meta-user\recipes-apps\"

6.7.1 startup

Script App to load init.sh from SD Card if available.

7 Appx. A: Change History and Legal Notices

7.1 Document Change History

To get content of older revision got to "Change History" of this page and select older document revision number.

Date	Document Revision	Authors	Description
2026-02-26	v.16 (see page 4)	Mohsen Chamanbaz[17]	• Initial release
--	all	Mohsen Chamanbaz[18]	--

9 Document change history.

7.2 Legal Notices

7.3 Data Privacy

Please also note our data protection declaration at https://www.trenz-electronic.de/en/Data-protection-Privacy

7.4 Document Warranty

The material contained in this document is provided “as is” and is subject to being changed at any time without notice. Trenz Electronic does not warrant the accuracy and completeness of the materials in this document. Further, to the maximum extent permitted by applicable law, Trenz Electronic disclaims all warranties, either express or implied, with regard to this document and any information contained herein, including but not limited to the implied warranties of merchantability, fitness for a particular purpose or non infringement of intellectual property. Trenz Electronic shall not be liable for errors or for incidental or

17 https://wiki.trenz.local/display/~M.Chamanbaz
18 https://wiki.trenz.local/display/~M.Chamanbaz

consequential damages in connection with the furnishing, use, or performance of this document or of any information contained herein.

7.5 Limitation of Liability

In no event will Trenz Electronic, its suppliers, or other third parties mentioned in this document be liable for any damages whatsoever (including, without limitation, those resulting from lost profits, lost data or business interruption) arising out of the use, inability to use, or the results of use of this document, any documents linked to this document, or the materials or information contained at any or all such documents. If your use of the materials or information from this document results in the need for servicing, repair or correction of equipment or data, you assume all costs thereof.

7.6 Product Disclaimer

Products (incl. Software and Firmware) are exclusively designed, manufactured, and distributed for industrial applications. They are not intended, authorized, or certified for use in critical safety applications, medical devices, aerospace, nuclear, life-support, traffic-control, military, or other environments where malfunction or failure could result in personal injury, death, or significant property or environmental damage.

Trenz Electronic GmbH (“Manufacturer”) excludes any liability for inappropriate use in such contexts. The Customer bears all responsibility for verifying suitability and compliance with applicable regulations when deploying in any critical-use scenario. The Customer also agrees to indemnify and hold harmless the Manufacturer from any third-party claims or liabilities arising from such use.

7.7 Copyright Notice

No part of this manual may be reproduced in any form or by any means (including electronic storage and retrieval or translation into a foreign language) without prior agreement and written consent from Trenz Electronic.

7.8 Technology Licenses

The hardware / firmware / software described in this document are furnished under a license and may be used /modified / copied only in accordance with the terms of such license.

7.9 Environmental Protection

To confront directly with the responsibility toward the environment, the global community and eventually also oneself. Such a resolution should be integral part not only of everybody's life. Also enterprises shall be conscious of their social responsibility and contribute to the preservation of our common living space. That is why Trenz Electronic invests in the protection of our Environment.

7.10 REACH, RoHS and WEEE

REACH

Trenz Electronic is a manufacturer and a distributor of electronic products. It is therefore a so called downstream user in the sense of REACH[19]. The products we supply to you are solely non-chemical products (goods). Moreover and under normal and reasonably foreseeable circumstances of application, the goods supplied to you shall not release any substance. For that, Trenz Electronic is obliged to neither register nor to provide safety data sheet. According to present knowledge and to best of our knowledge, no SVHC (Substances of Very High Concern) on the Candidate List[20] are contained in our products. Furthermore, we will immediately and unsolicited inform our customers in compliance with REACH - Article 33 if any substance present in our goods (above a concentration of 0,1 % weight by weight) will be classified as SVHC by the European Chemicals Agency (ECHA)[21].

RoHS

Trenz Electronic GmbH herewith declares that all its products are developed, manufactured and distributed RoHS compliant.

WEEE

Information for users within the European Union in accordance with Directive 2002/96/EC of the European Parliament and of the Council of 27 January 2003 on waste electrical and electronic equipment (WEEE).

Users of electrical and electronic equipment in private households are required not to dispose of waste electrical and electronic equipment as unsorted municipal waste and to collect such waste electrical and electronic equipment separately. By the 13 August 2005, Member States shall have ensured that systems are set up allowing final holders and distributors to return waste electrical and electronic equipment at least free of charge. Member States shall ensure the availability and accessibility of the necessary collection facilities. Separate collection is the precondition to ensure specific treatment and recycling of waste electrical and electronic equipment and is necessary to achieve the chosen level of protection of human health and the environment in the European Union. Consumers have to actively contribute to the success of such collection and the return of waste electrical and electronic equipment. Presence of hazardous substances in electrical and electronic equipment results in potential effects on the environment and human health. The symbol consisting of the crossed-out wheeled bin indicates separate collection for waste electrical and electronic equipment.

Trenz Electronic is registered under WEEE-Reg.-Nr. DE97922676.

2025-08-01

19 http://guidance.echa.europa.eu/
20 https://echa.europa.eu/candidate-list-table
21 http://www.echa.europa.eu/