Application Note

Akademie věd České republiky
Ústav teorie informace a automatizace AV ČR, v.v.i.

# Trenz TEBF0808 + TE0808-04-6EB21A SoM Running Petalinux 2018.2 Kernel with Debian Buster File System

Lukas Kohout
kohoutl@utia.cas.cz

## Revision history

| Rev. | Date | Author | Description |
|---|---|---|---|
| 0 | 20. 04. 2021 | L. Kohout | Initial draft |
| 1 | | | |
| 2 | | | |
| | | | |

## Content



## Acknowledgement

This work has been supported from project Arrowhead Tools, project number ECSEL 826452 and corresponding Czech NFA (MSMT) institutional support project 8A19009.

# 1 Introduction

This document describes an installation procedure of the Linux image on the Trenz Electronic TE0808 HW platform (TEBF0808 carrier board [1] with Trenz Electronic TE0808-04-6EB21A System on Module [2]) and how to develop SW applications running on this platform.

# 2 Description

The Linux image is based on Petalinux 2018.2 kernel and Debian Buster (version 10.9) file system. It is a bit copy of SD card for TE0808 HW platform. It consists of two partitions.

- The first partition uses FAT32 file system and its size is 256 MB. This partition stores *BOOT.bin* file and Petalinux 2018.2 kernel image (file *image.ub*). Once the image is written on the SD card, this partition is readable and writable from Windows OS and Linux based OS.
- The second partition uses EXT4 file system and it stores Debian Buster OS files. Once the image is written on the SD card, this partition can be read or written from Linux OS, but Windows OS cannot it access natively. To access it from Windows OS you can try Microsoft WSL 2 technology:

  https://docs.microsoft.com/en-us/windows/wsl/wsl2-mount-disk

  or you can use some third-party driver/application:

  https://www.easeus.com/partition-master/access-ext4-partition.html.

  There is also possible to access the partition indirectly with SFTP or FISH protocols via Ethernet when the system is running.

# 3 Used tools and Resources

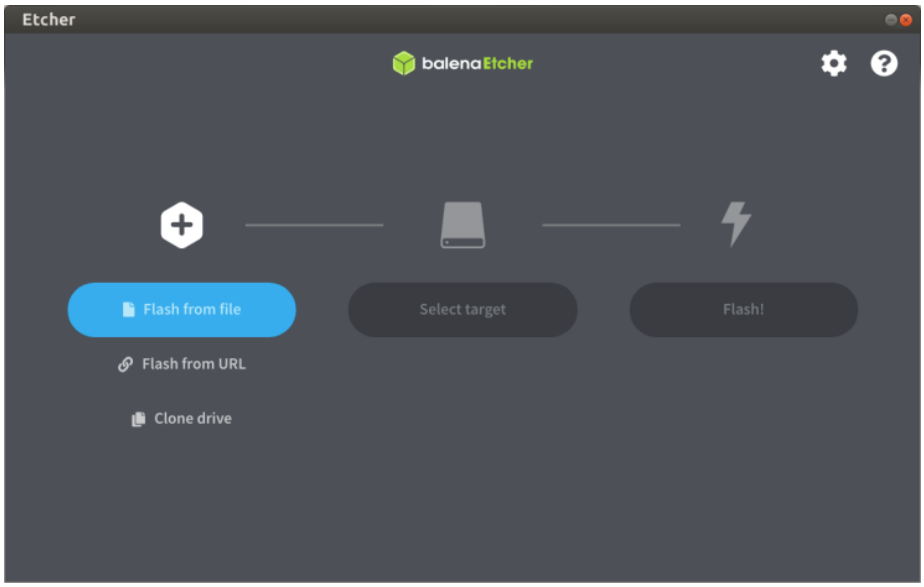
- Trenz Electronic TEBF0808 carrier board [1].
- Trenz Electronic TE0808-04-6EB21A SoM. It contains Xilinx Zynq UltraScale+ device with 4GB DDR4 memory [2].
- 8 GB or larger SD card. As the file system runs from the SD card it should be as fast as possible, recommended minimum is class 10.
- SD card reader for PC.
- BalenaEtcher tool to write image to SD card (Windows, Linux or MAC):

  https://www.balena.io/etcher/
- Mini USB cable for serial terminal or JTAG connection.
- USB keyboard and USB mouse.
- Ethernet cable.
- DisplayPort cable.
- Full HD Monitor with DisplayPort input.
- Xilinx Software Development Kit 2018.2, it is a part of Xilinx Vivado Design Suite 2018.2 (Web Pack edition is sufficient). To download the tool go to Xilinx web page:
- https://www.xilinx.com/support/download/index.html/content/xilinx/en/downloadNav/vivado-design-tools/archive.html.

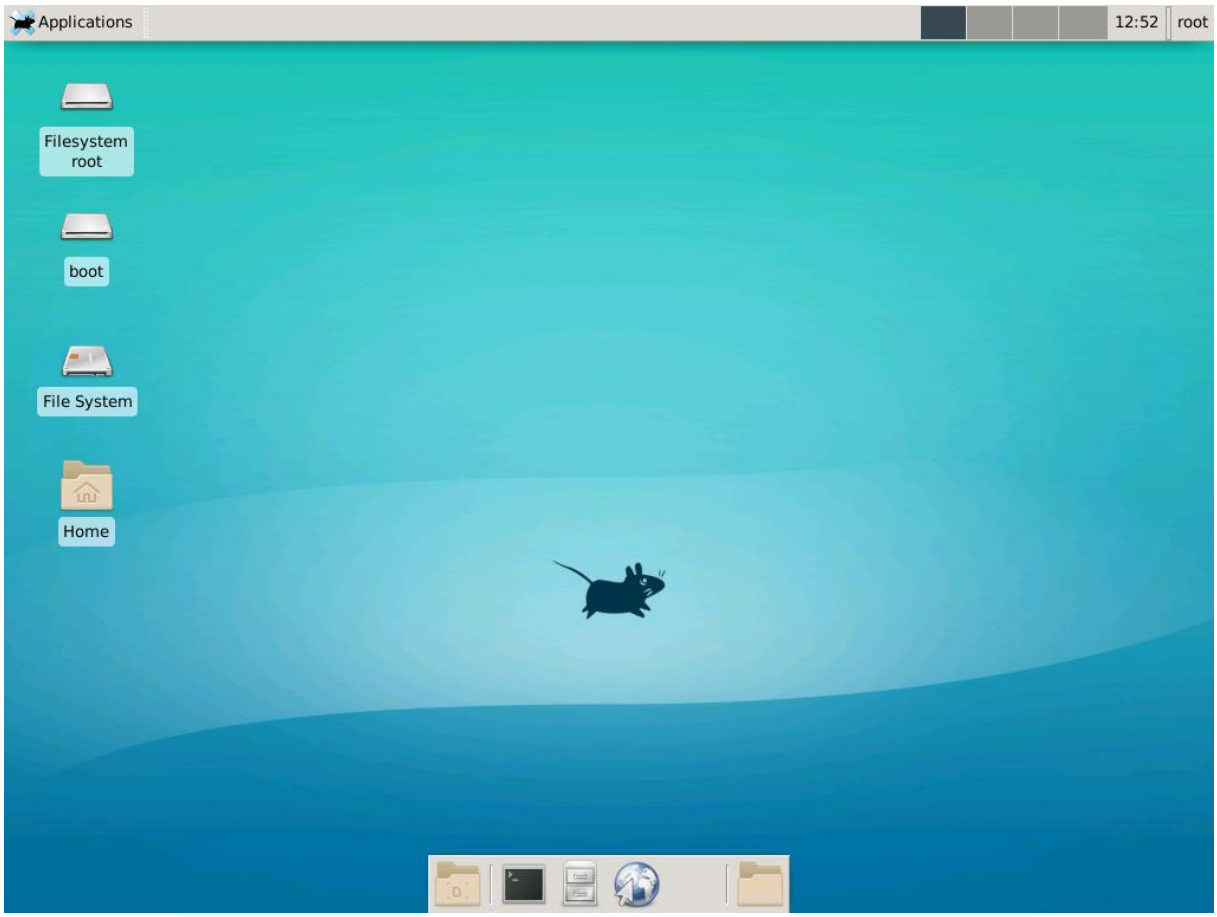
# 4 Installation

To install Linux image to the SD card follow steps bellow:

1. Unpack the image from ZIP file *te0808-debian-buster.zip*. You should get file *te0808-debian-buster-2021-04-20.img*.




2. Start BalenaEtcher tool, downloadable from https://www.balena.io/etcher/.
3. In BalenaEtcher tool click on *Flash from file* button and browse to file *te0808-debian-buster-2021-04-20.img*.
4. In BalenaEtcher tool click on *Select target* button and select the SD card drive you want to write.
5. In BalenaEtcher tool click on *Flash!* button. The tool asks you to permit the operation. Wait until the flashing will be finished. If you perform the flashing operation in Windows OS, the system offers you to format EXT4 partition of the SD card because it does not know the file system. **Do not do this!**
6. Insert the SD card to the reader on the TE0808 platform.
7. Configure TEBF0808 carrier board
   - Set *S4* to "off off off on"
   - Set *S5* to "on on on on"
   - All other switches and jumpers let in their defaults.
8. Connect mini USB cable to JTAG, XMOD1 module on the TEBF0808 carrier board.
9. Connect USB keyboard and USB mouse.
10. Connect Full HD monitor via DisplayPort.
11. Connect the board to the local net with your PC.
12. Power the board on.
13. Start serial terminal, putty for instance. The settings are:
    - Baud rate – 115200
    - Data bits – 8
    - Stop bits – 1
    - Parity – none
    - Flow control – none
14. Push button S1 and then push button S2.
15. On the terminal there could be seen a booting system. The login name is *root* and password is *root*. Be aware that the combination *root/root* is security threat and you should use it only in a closed testing environment.




16. On the monitor you can see Xfce 4 desktop of the Debian OS.
17. The system is configured to get an IP address from DHCP server. To print the IP address use command *ifconfig* from the terminal.
18. Explore home folder of user *root*

```
cd /root
ls -la
```

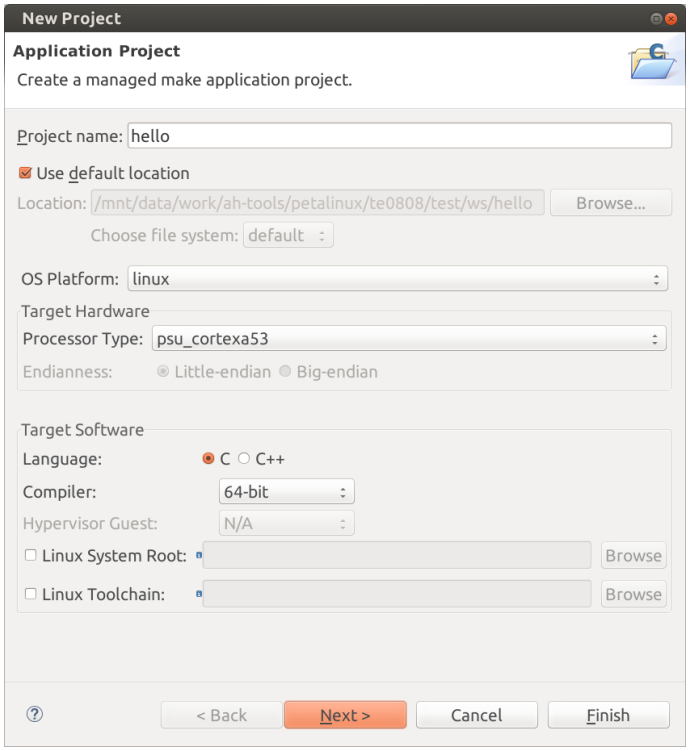
**IMPORTANT NOTE:** Never stop the board just by powering it off. As the file system is located on relatively slow SD card and the OS uses postponed method of writing, there can be unfinished writes. If you only power the board off, you can damage the system. Hence always stop the system by command *halt* from the terminal and wait until it indicates it is halted. After that you can power the board off. In case you want to reset the system safely, you can use command *reboot*.

# 5 Hello World Application

To develop Linux applications it is used the same toolchain as we are using for developing standalone applications. It is Xilinx Software Development Kit 2018.2. To create *Hello World* application follow steps bellow:

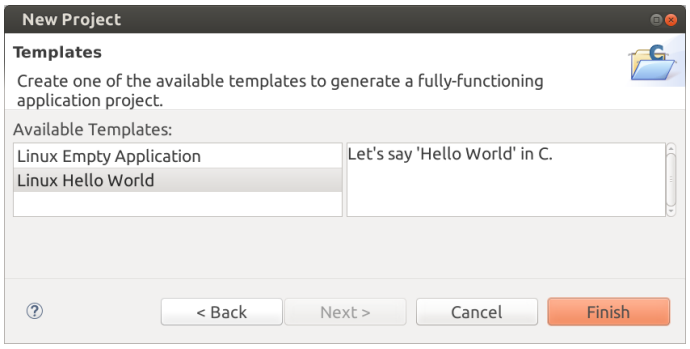
1. Create new empty workspace folder in your user space, *ws* for instance.
2. Start Xilinx SDK 2018.2, set workspace path to *ws* folder.
3. Create a new application, menu:
   *File -> New -> Application Project*
   Project name: *hello*

OS Platform: linux
Processor Type: *psu_cortexa53*
Language: *C*
Compiler: 64-bit




Click on *Next* > button.

Select *Linux Hello World* and click on *Finish* button.




4. Compile the project, menu *Project->Build All*.
5. Get an IP address of the TE0808 HW platform. If you want to deploy, debug and execute the compiled application, the TE0808 HW platform has to be in the same network as your PC is, because it is performed via Ethernet. To get currently set IP address of the system use command *ifconfig* from the terminal:

```
root@zynqmp:~# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 10.42.0.143  netmask 255.255.255.0  broadcast 10.42.0.255
        inet6 fe80::20a:35ff:fe00:2201  prefixlen 64  scopeid 0x20<link>
        ether 00:0a:35:00:22:01  txqueuelen 1000  (Ethernet)
...
```

In case that the DHCP server is not available in your local network, you can set IP address manually. For instance, you are using your local network in IP range

10.42.0.x, where the IP address of your gateway is 10.40.0.1, IP address of your PC is 10.42.0.10 and you want to set the IP address of the TE0808 HW platform to 10.42.0.11. To set the IP address of the TE0808 board execute from command line:

a) Set your IP address:

```
ifconfig eth0 10.42.0.11 netmask 255.255.255.0 up
```

b) Set your default gateway:

```
route add default gw 10.42.0.1
```

c) If you know the IP address of the name server, you can set it as well. But it is not needed in this example.

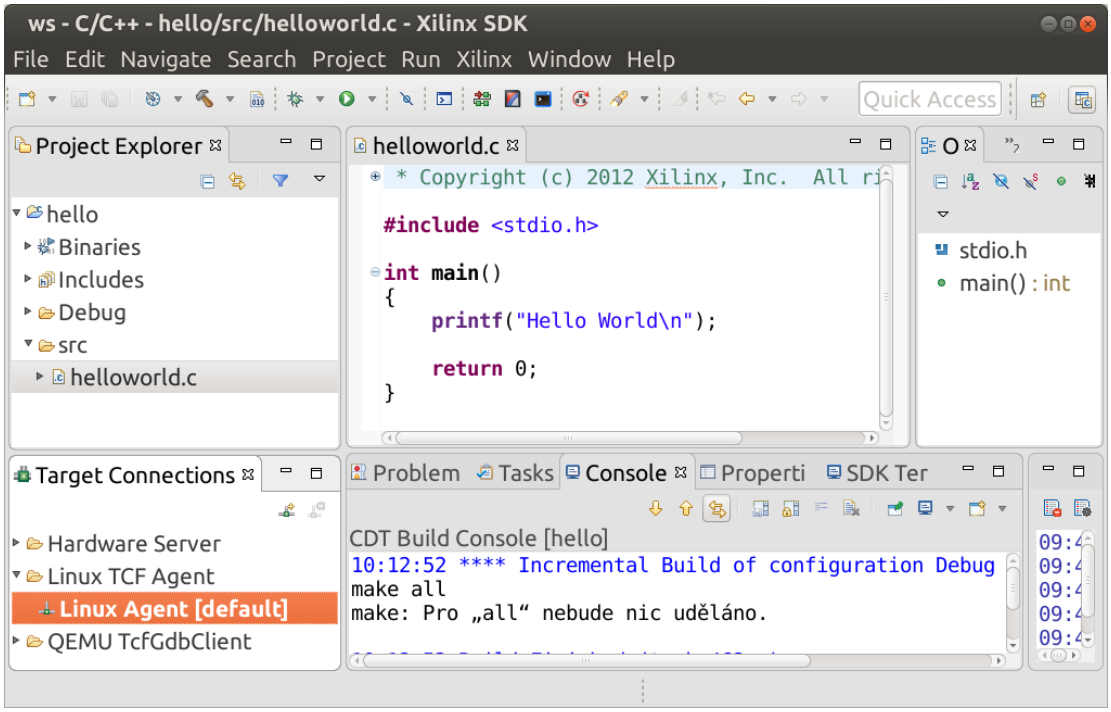
```
echo "nameserver 10.42.0.1" > /etc/resolv.conf
```

Test:

```
ping google.com
```

6. Set TCF Agent client to connect TE0808 HW platform, on TE0808 HW platform Linux has already run TCF Agent server. In Xilinx SDK double click on:

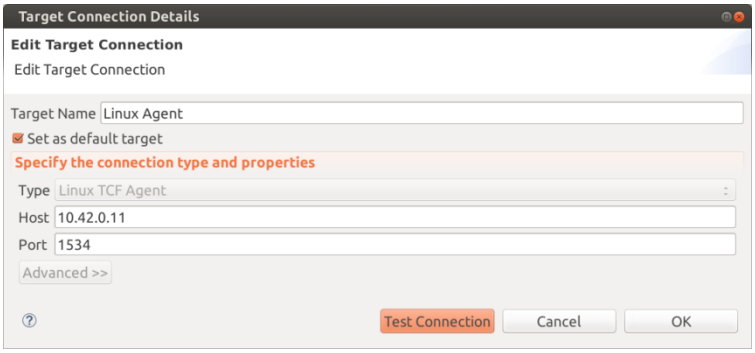
*Target Connections -> Linux Agent [default]*




Project name: *Linux Agent*

Host: *10.42.0.11* (set IP of your TE0808 HW platform)

Port 1534



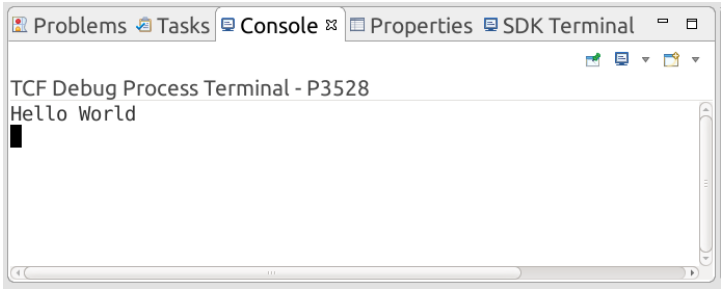
Click on *Test Connection* button, it should say:

*Successfully established connection to 'Linux TCF Agent' on the host '10.42.0.11'*

7. Run compiled application on TE0808 HW platform. In *Project Explorer* select *hello* and right click on it:

   *Run As->Launch on Hardware (System Debugger)*

   The executable binary file of the *hello* application is deployed to the TE0808 HW platform and executed. Observe Xilinx SDK console.

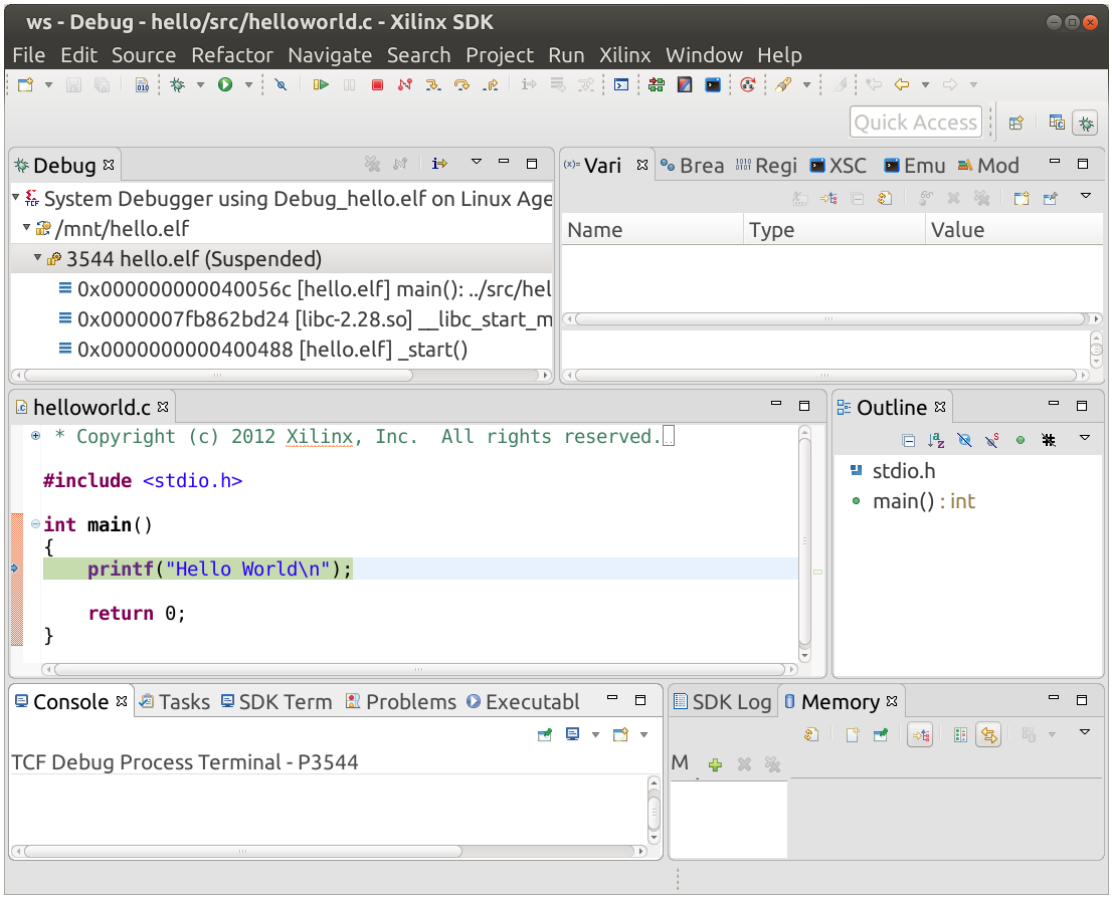



   The application binary file is located in folder */mnt* on the TE0808 HW platform. To run the application directly on the TE0808 platform execute from terminal:

```
cd /mnt
./hello.elf
```

8. Debug compiled application on TE0808 HW platform. In *Project Explorer* select *hello* and right click on it:

   *Debug As->Launch on Hardware (System Debugger)*



# 6 Package Content

```
.
├── doc
└── te0808-debian-buster.zip
```

# 7 References


[1] Trenz Electronic, „UltraITX+ Baseboard for Trenz Electronic TE080X UltraSOM+,“ [Online]. Available: https://shop.trenz-electronic.de/en/TEBF0808-04A-UltraITX-Baseboard-for-Trenz-Electronic-TE080X-UltraSOM.

[2] Trenz Electronic, „UltraSOM+ MPSoC Module with Zynq UltraScale+ XCZU6EG-1FFVC900E, 4 GB DDR4,“ [Online]. Available: https://shop.trenz-electronic.de/en/TE0808-04-6BE21-A-UltraSOM-MPSoC-Modul-with-Zynq-UltraScale-XCZU6EG-1FFVC900E-4-GB-DDR4.

## Disclaimer

This disclaimer is not a license and does not grant any rights to the materials distributed herewith. Except as otherwise provided in a valid license issued to you by UTIA AV CR v.v.i., and to the maximum extent permitted by applicable law:

(1) THIS APPLICATION NOTE AND RELATED MATERIALS LISTED IN THIS PACKAGE CONTENT ARE MADE AVAILABLE "AS IS" AND WITH ALL FAULTS, AND UTIA AV CR V.V.I. HEREBY DISCLAIMS ALL WARRANTIES AND CONDITIONS, EXPRESS, IMPLIED, OR STATUTORY, INCLUDING BUT NOT LIMITED TO WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, OR FITNESS FOR ANY PARTICULAR PURPOSE; and

(2) UTIA AV CR v.v.i. shall not be liable (whether in contract or tort, including negligence, or under any other theory of liability) for any loss or damage of any kind or nature related to, arising under or in connection with these materials, including for any direct, or any indirect, special, incidental, or consequential loss or damage (including loss of data, profits, goodwill, or any type of loss or damage suffered as a result of any action brought by a third party) even if such damage or loss was reasonably foreseeable or UTIA AV CR v.v.i. had been advised of the possibility of the same.

Critical Applications:
UTIA AV CR v.v.i. products are not designed or intended to be fail-safe, or for use in any application requiring fail-safe performance, such as life-support or safety devices or systems, Class III medical devices, nuclear facilities, applications related to the deployment of airbags, or any other applications that could lead to death, personal injury, or severe property or environmental damage (individually and collectively, "Critical Applications"). Customer assumes the sole risk and liability of any use of UTIA AV CR v.v.i. products in Critical Applications, subject only to applicable laws and regulations governing limitations on product liability.