Reproducible System Composition – Combining Linux, Xen & Zephyr on One Embedded Hardware

Thomas Mittelstädt, Robert Bosch GmbH
Philipp Ahmann, Robert Bosch GmbH

@ProjectElisa
“The mission of the project is to define and maintain a common set of elements, processes and tools that can be incorporated into Linux-based, safety-critical systems amenable to safety certification.”

“The scope of the project includes software and documentation development under an OSI-approved license supporting the mission, including documentation, testing, integration and the creation of other artifacts that aid the development, deployment, operation or adoption of the project.”

from the technical charter
Working Groups (WGs) - Horizontal

- Safety Architecture
  - Red Hat

- Linux Features
  - Mobileye™/Intel

- Tool investigation & Code Improvement
  - Elektrobit, Codethink

- Open Source Engineering Process
  - Codethink

- Systems
  - BOSCH
Working Groups (WGs) - Verticals

Aerospace

Automotive

Medical Devices

OpenAPS elements
1. Continuous glucose monitor
2. Computer
3. Battery
4. Radio stick
5. Insulin pump

Dana Lewis’ OpenAPS project: https://youtu.be/kgu-AYSnyZ8
meta-elisa: Various starting points provided

- Plain and native from source
  https://github.com/elisa-tech/meta-elisa

- Using docker container
  https://github.com/elisa-tech/wg-automotive/tree/master/Docker_container

- With cached build using SSTATE
  modify "conf/local.conf" after the "source" command
  before the "bitbake" command

- Download binaries directly from build server
  https://gitlab.com/elisa-tech/meta-elisa-ci
Full description in the blog

https://elisa.tech/blog/2023/04/05/elisa-ci-enablement-automation-tools-for-easier-collaboration/
ELISA Working Groups - Fit in an exemplary system

- **Linux Features, Architecture** and **Code Improvements** should be integrated into the reference system directly.

- **Tools** and **Engineering process** should serve the reproducible product creation.

- **Medical, Automotive, Aerospace** and future WG use cases should be able to strip down the reference system to their use case demands.
Interaction with other communities (outside of ELISA)

- Open source projects focusing on safety-critical analysis
  ![Xen Project](image1.png) ![Zephyr](image2.png)

- Open source projects with safety-critical relevance and comparable system architecture considerations
  ![Automotive Grade Linux](image3.png) ![SOAFEE](image4.png) ![SDV](image5.png)

- Further community interactions
  ![Yocto](image6.png) ![SPDX](image7.png) ![Linaro](image8.png)

“If you have an apple and I have an apple and we exchange these apples then you and I will still each have one apple. But if you have an idea and I have an idea and we exchange these ideas, then each of us will have two ideas.”

— George Bernard Shaw
“When it comes to prototyping systems, the existing guidelines are limited; reproducing demos is hard and time consuming.”
Static Partitioning with Xen, LinuxRT, and Zephyr: a concrete end-to-end example

Stefano Stabellini
Embedded Linux Conference 2022
“A product will run on real hardware.”
Major problems to setup XEN systems

- Select target board with
  - Hardware support for XEN, especially SMMU controller
  - XEN community support
  - Documentation for build and setup
  - Licenses compliant to OSS project
- Setup of Yocto build environment
  - Amount of computer resources
  - Network and Host dependencies
- Finding valid descriptions
- Build image parts based on descriptions
- Finding community support at occurring build problems
- Understanding XEN setup and structure
Evaluated targets

- Renesas RCAR 3.0 family ([link to Wiki of eLinux](#))
  - + XEN hardware support
  - + Functional XEN systems (also graphic)
  - - Proprietary licenses for essential parts like graphic
  - - Not available at standard market

- Xilinx Zynqmp and Ultrascale family ([link to product page](#))
  - + XEN hardware support
  - + Functional XEN systems
  - + Good documentation and open source support of Xilinx
  - - Zcu102 well supported, but additional complexity due to FPGA programming
  - - Graphic at Zcu102 atm not able to be handled by XEN
Evaluated targets

- Qemu systems for Xilinx ([link](#) with some hints for setup with XEN at Xilinx boards)
  - + XEN support
  - + Functional XEN systems
  - + no hardware needed
  - - Only for development, not for hardware related demo cases
- Raspberry Pi systems
  - - Hardware support not sufficient for security requirements of XEN
- NXP Imx.8 systems
  - + Good hardware support for hypervisor like XEN
  - - Less community support
Used hardware

- Board ZCU102 ([link](#) to description)
  - Reference manual ([link](#))
  - SD card 16GB for boot loader
  - USB Stick 16GB for demonstrator setup
  - USB-Ethernet-Adapter (DLink)

- Environment for setup
  - Local DHCP server (VM with system networkd)
  - Putty for serial console
  - USB Keyboard (for TTY console)
  - HDMI screen
Overview to XEN demonstrator

- Build sources
- Local tools
- SW parts
- Hardware

- Xilinx Yocto 2022.2 for XEN
- BSP v2022.2 Zcu 102
- Apertis Build
- RCAR demonstrator (meta-xt-prod-devel-rcar)
- Building Xen Hypervisor with PetaLinux 2022.2

- Local network with DHCP
- Bootloader
- XEN system
- Simple Petalinux
- Apertis (Debian)
- Zephyr
- Demo setups

- USB ethernet adapter
- Local Ethernet
- SD Card
- USB Stick
- Hardware ZCU 102
- Putty (serial console)
Development steps

- Setup of Yocto environment
  - Native build dependent on Linux system
    - Better: Docker image to build [link to build description]
- Work with system image files, not directly with SD cards or other media
- Build XEN images at Yocto environment
- Get BSP with prebuilt images from Xilinx (also other systems are possible for DomU)
- Create system images with Xilinx compliant scheme
- Copy files for XEN Hypervisor to image file
- Create fitting boot scripts with XEN image builder
- Start XEN and Domain-0 and check system access to USB, Ethernet and SD card
- Create XEN configuration files for XEN demonstrations and setup related binaries
- Test XEN demonstrations
System images for SD card and USB stick

- Scheme
  - MSDOS partition scheme
  - Boot partition (fat)
    - Boot.bin (Boot loader (U-Boot) and FPGA stream)
    - Files for XEN Hypervisor (Domain-0) (Ramdisk)
    - Files for DomU (Petalinux, Zephyr, Apertis) (Ramdisk)
    - XEN configuration files to demonstrate DomU setups
    - Boot scripts to start customized Domain-0
  - Data partition (ext4)
    - /etc to make configuration of Domain-0 persistent
  - Root file system partition of some DomU (ext4)
External parts of system images

- Xen Hypervisor ([link](#) for build description)
  - Image, ramdisk, device tree
  - Boot.bin
- Petalinux ([link](#) for binaries from "BSP")
  - Image, ramdisk, (device tree: not used for XEN)
- Zephyr (atm got from Demo for Renesas RCAR, [link](#) for build description)
  - Image
  - Configuration file for XEN
- XEN configuration files (created on description at [link](#))
- Apertis (Debian based, specific image, but general build instructions at [link](#))
  - Image, ramdisk, (device tree: not used for XEN)
- XEN image builder ([link](#) for download and usage)
Overview to XEN demonstrations

- Simple: DomU with Petalinux as ramdisk image, no additional devices
- Zephyr: DomU with simple Zephyr application (2 tasks), no additional devices
- Apertis: DomU with small Apertis system as ramdisk image, no additional devices
- Paravirtualized network: DomU with Petalinux as ramdisk image, additional NET interface
  - Shell script to create network bridge at Domain-0
- Paravirtualized block device: DomU with Petalinux, booted from paravirtualized block device
  - Partition 3 used to boot the system
- Passthrough of SD card controller: DomU with Petalinux as ramdisk image, additional block device
  - Domain-0 has to be started with "passthrough" option of SD card controller
- Passthrough of ethernet controller: DomU with Petalinux as ramdisk image, additional network interface
  - Domain-0 has to be started with "passthrough" option of ethernet controller
Simple: DomU with Petalinux as ramdisk image, no additional devices

- Starting with bootloader
- Booting with XEN and Domain-0
- Check list of started Domains
- Starting "Simple" demo with Petalinux ramdisk image
- Switch console to Guest0
- No Network interfaces
- No block devices but ramdisks
- Getting back console to Domain-0 (XEN)
- Destroy DomU Guest0
- Guest0 has been removed from XEN domain list

Video: https://youtu.be/NLmKtPcSl78
Zephyr: DomU with simple Zephyr application (2 tasks), no additional devices

- End of booting **Domain-0**
- Starting “Zephyr” demo with Zephyr image
- Check list of started Domains
- Switch console to **DomZ** (Zephyr)
- Getting back console to **Domain-0** (XEN)
- Destroy DomU "DomZ"
- "DomZ" has been removed from XEN domain list

Video: https://youtu.be/CBrFMDZJbJA
Apertis: DomU with small Apertis system as ramdisk image, no additional devices

- End of booting Domain-0
- Starting "Apertis" demo with Apertis ramdisk image
- Check list of started Domains
- Switch console to Apertis
- No Network interfaces
- No block devices
- Getting back console to Domain-0 (XEN)
- "Apertis" has been removed from XEN domain list

Video: https://youtu.be/Jb6hqR9PG3Y
Paravirtualized network: DomU with Petalinux as ramdisk image, additional NET interface

- End of booting Domain-0
- Check Network Connection at Domain-0
- Create a network bridge to connect DomU Guest0
- Starting Petalinux ramdisk image as Guest0 with paravirtualized network setup
- Switch console to Guest0
- Check Network Connection at Guest0
- Guest0 (192.168.1.94) tries to ping Domain-0 (192.168.1.98)
- Guest0 has also connection to internet (google.de)
- Guest0 can access Domain-0 (192.168.1.98) via ssh connection
- Getting back console to Domain-0 (XEN)
- Domain-0 can access Guest0 (192.168.1.94) via ssh connection
- Guest0 is not destroyed, but initiates a normal shutdown.
- Guest0 has been removed from XEN domain list

Video: https://youtu.be/v2WtPtryPrc
Paravirtualized block device: DomU with Petalinux, booted from paravirtualized block device

- End of booting Domain-0
- Starting "Paravirtualized block device" demo with Petalinux booting from P3 of file system (USB-Stick)
- Check list of started Domains
- Switch console to Guest0
- /dev/xvda is the paravirtualized block device (P3 of USB stick)
- Guest0 is not destroyed, but initiates a normal shutdown.

Passthrough of SD card controller: DomU with Petalinux as ramdisk image, additional block device

- Interrupt the automatic boot process at bootloader
- Configure passthrough of SD card device as "passthrough"
- Booting with XEN and Domain-0
- /dev/mmcblk0 (SD card) is not assigned to Domain-0
- Starting Petalinux ramdisk image as Guest0 with passthrough configuration of SD card device
- Switch console to Guest0
- /dev/mmcblk0 is the passed through SD card device and keep only the bootloader
- Try to mount P1 of SD card device and have a look
- Guest0 is not destroyed, but initiates a normal shutdown.

Video: https://youtu.be/YXHfr8yFzy4
Passthrough of ethernet controller: DomU with Petalinux as ramdisk image, additional network interface

- Interrupt the automatic boot process at bootloader
- Configure passthrough of internal ethernet controller as "passthrough"
- Booting with XEN and Domain-0
- Check Network Connection at Domain-0 with used USB-Ethernet adapter
- Starting Petalinux ramdisk image as Guest0 with passthrough configuration of internal ethernet controller
- Switch console to Guest0
- Check network connection at Guest0
- Guest0 has connection to internet (google.de)
- Getting back console to Domain-0 (XEN)
- Domain-0 (192.168.1.81) tries to ping Guest0 (192.168.1.95)
- Domain-0 can access Guest0 (192.168.1.95) via ssh connection
- Guest0 exits from ssh connection, getting back console to Domain-0
- Switch console to Guest0
- Guest0 can access Domain-0 (192.168.1.81) via ssh connection

Video: https://youtu.be/hAwUO2-O79w
Getting involved...

- Join main technical and weekly calls of interest:
  - Main Technical List: devel@lists.elisa.tech
  - Safety Architecture Workgroup: safety-architecture@lists.elisa.tech
  - Open-Source Engineering Process WG: osep@lists.elisa.tech
  - Linux Features for Safety-Critical Systems WG: linux-features@lists.elisa.tech
  - Medical Devices Workgroup: medical-devices@lists.elisa.tech
  - Automotive Workgroup: automotive@lists.elisa.tech
  - Systems Workgroup: systems@lists.elisa.tech
  - (Full list at: https://lists.elisa.tech/g/linux-features/subgroups)

- Contribute content, review materials and add your comments to:
  - ELISA Technical Community Google Drive: https://drive.google.com/open?id=1Y6Uwqt5VEDEZjpRe0C8ClbdtXPgLwIG
  - ELISA github repository: https://github.com/elisa-tech/workgroups
  - ELISA github issue tracker: https://github.com/elisa-tech/workgroups/issues
THANK YOU!

EMBEDDED OPEN SOURCE SUMMIT