Industrial-grade Open Source Base Layer Development

Yoshitake Kobayashi, Toshiba Corp.
Urs Gleim, Siemens AG

Embedded Linux Conference Europe, Prague, October 24, 2017
More updates will be announced at CIP session on October 24th @ 11:45am.
What is CIP?
What is CIP?

• One of the most conservative open source project in the Linux Foundation

• One of the most important projects for our civilization
What is CIP?

• One of the most conservative open source project in the Linux Foundation

• CIP aims to
  • Provide an open source base layer for CIP related embedded systems
  • Work closely with the upstream community

• CIP does not aim to
  • Create a new Linux distribution
Our Civilization is run by Linux
There are issues to be solved…

https://www.airpano.com/360Degree-VirtualTour.php?3D=San Francisco, USA
DebConf17, Montréal, CANADA
ELCE17, Prague, Czech Republic
A Railway System:

25-50 years products life-cycle

with very reluctant nature for product update and upgrade of hardware and base software platform
Railway Example

3 – 5 years development time

2 – 4 years customer specific extensions

1 year initial safety certifications / authorization

3 – 6 months safety certifications / authorization for follow-up releases (depending on amount of changes)

25 – 50 years lifetime
Industrial IoT: Edge and Fog Computing

Functionality is moving from the cloud to the “Edge”

- Increasing number of networked industrial-grade devices
- Security management requires harmonized software landscape

Application examples on IIoT infrastructure

- Plant analytics
- SCADA functionality
- Plant (device) mgmt.
- Local / real-time analytics
- IoT Gateways

Data collection
Pre-processing
Sensor / actor connectivity

Ecosystems also for backend
Controlled network zone
Multiple users with different roles at different levels

IoT: Internet of Things  IIoT: Industrial IoT  SCADA: Supervisory Control And Data Acquisition

ELCE17, Prague, Czech Republic
The Problems we face …

• The systems that support our modern civilization need to **survive for a VERY LONG TIME**. Until now the corresponding industrial grade super long term maintenance has been **done individually by each company**.

• These systems not only have to survive for a long time, they must be “**INDUSTRIAL GRADE**” (robust, secure and reliable). And at the same time the industry will also need to **catch up with the latest technology trends**.
The genesis of a collaborative project
Linux Foundation Projects

Driving joint efforts and backing them with people and budget.

The majority focusses in IT, enterprise, cloud technologies.

ELCE17, Prague, Czech Republic
The Solutions we need ...

- **We need a Collaborative framework** to maintain the same open source based system for many, many, many years to keep it secure, robust and reliable.

- AND most importantly, we need to do this collaboratively in the **upstream communities**, not locally.
Establishing an Open Source Base Layer of industrial-grade software to enable the use and implementation of software building blocks for Civil Infrastructure Systems

https://www.cip-project.org/

since April 2016
The backbone of CIP are the member companies

Member companies

- HITACHI
- TOSHIBA
- SIEMENS
- Renesas
- CodeThink

Open source projects (Upstream work)

Budget

Developers, maintainers

Optional: funding of selected projects

Contribution & usage / integration

CIP Super Long Term Support Project

CIP source code repositories

ELCE17, Prague, Czech Republic

19
What is CIP, again?
What is “Open Source Base Layer (OSBL)”?

• OSBL is a set of industrial grade core open source software components, tools and methods

Open Source Base Layer

• Open source based reference implementation
• Start from a minimal set for controllers in industrial grade systems

Non-CIP packages
Linux distribution (e.g. Debian) may extend/include CIP packages.

CIP Reference Filesystem image with SDK (CIP Core packages)

CIP SLTS Kernel

CIP Reference Hardware

User space
Hardware
Kernel
Development plan

CIP will increase the development effort to create a industrial grade common base-layer

Phase 1:
- Define supported kernel subsystems, arch.
- Initial SLTS component selection
- Select SLTS versions
- Set-up maintenance infrastructure (build, test)

Phase 2:
- Patch collection, stabilization, back port of patches for CIP kernel packages
- Support more subsystems
- Additional core packages

Phase 3:
- Domain specific enhancements, e.g. communication protocols, industrial IoT middleware
- Optionally: more subsystems
- Optionally: more core packages

ELCE17, Prague, Czech Republic
### Vision: Technical topics and related projects

*Topics will be added or removed to reflect CIP technical interests*

<table>
<thead>
<tr>
<th>Application support</th>
<th>Infrastructure and Services</th>
</tr>
</thead>
<tbody>
<tr>
<td>[App Framework]</td>
<td>[Yocto Project]</td>
</tr>
<tr>
<td>[HMI Framework]</td>
<td>[Debian build system]</td>
</tr>
<tr>
<td>[FW update]</td>
<td>[LTP test suite]</td>
</tr>
<tr>
<td>[App deployment]</td>
<td>[CIP test suite]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Middleware / Tools</th>
<th>Coherent Security Mechanisms</th>
</tr>
</thead>
<tbody>
<tr>
<td>Domain specific and IoT communication</td>
<td></td>
</tr>
<tr>
<td>[OPC UA]</td>
<td>[Configuration]</td>
</tr>
<tr>
<td>[TSN]</td>
<td>[Device management]</td>
</tr>
<tr>
<td>[Echonet]</td>
<td>[Industrial Zeroconf]</td>
</tr>
<tr>
<td>[OM2M]</td>
<td>[Userland Isolation]</td>
</tr>
<tr>
<td>[Avnu]</td>
<td>[LXC]</td>
</tr>
<tr>
<td>[Industrial special-purpose protocols]</td>
<td>[Groups]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Linux Kernel</th>
<th>RTOS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td></td>
</tr>
<tr>
<td>[Anomaly detection]</td>
<td>[Monitoring / error detection]</td>
</tr>
<tr>
<td>[Live patching]</td>
<td></td>
</tr>
<tr>
<td>[SIL2LinuxMP]</td>
<td>[Communication]</td>
</tr>
<tr>
<td>[SIL3 support]</td>
<td>[Jailhouse]</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Functional Safety</th>
<th>Userland Isolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>[Monitoring / error detection]</td>
<td>[LXC]</td>
</tr>
<tr>
<td>[SIL2LinuxMP]</td>
<td>[Cgroups]</td>
</tr>
<tr>
<td>[SIL3 support]</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Real-time support</th>
<th>Monitoring / Tracing</th>
<th>Heterogeneous Computing</th>
</tr>
</thead>
<tbody>
<tr>
<td>[GPGPU/FPGA real-time]</td>
<td>[Ftrace]</td>
<td>[SoC FPGA]</td>
</tr>
<tr>
<td>[RT/non-RT communication]</td>
<td>[ktap]</td>
<td></td>
</tr>
<tr>
<td>[Xenomai]</td>
<td>[RAS]</td>
<td></td>
</tr>
<tr>
<td>[PREEMPT-RT]</td>
<td>[Communication]</td>
<td></td>
</tr>
</tbody>
</table>

| Hardware / SoC (x86 or ARM based) | |
|----------------------------------| |

**Legend**

- To be specified / implemented by CIP
- Integration / cooperation

---

ELCE17, Prague, Czech Republic
CIP activities and status
Scope of activities

- **User space**
  - App container infrastructure (mid-term)
  - Domain Specific communication (e.g., OPC UA)
  - Safe & Secure Update
  - Real-time support
  - Super Long Term Supported Kernel (STLS)

- **Kernel space**
  - App Framework (optionally, mid-term)
  - Shared config. & logging
  - Monitoring
  - Security
  - Real-time / safe virtualization

- **Middleware/Libraries**
  - CIP Core Packages
  - Multimedia

- **Tools**
  - Build environment (e.g., bitbake, dpkg)
  - Test automation
  - Tracing & reporting tools
  - Configuration management
  - Device management (update, download)
  - Application life-cycle management

- **Concepts**
  - Functional safety architecture/strategy, including compliance w/ standards (e.g., NERC CIP, IEC61508)
  - Long-term support Strategy: security patch management
  - Standardization collaborative effort with others
  - License clearing
  - Export Control Classification

- **On-device software stack**

- **Product development and maintenance**
Kernel maintenance

- The first action taken by the CIP project is to select and maintain Linux kernels for very long time (+15 years). To achieve goal a group of experts has been assigned.
- PREEMPT_RT patches are added to the CIP kernel

Testing

- Civil infrastructure industry has high stability, reliability and security standards in order to ensure safety critical systems. The CIP Testing project has been formed to address this reality. So far the efforts are focused on testing the CIP kernel. In the future they will be extended to the complete CIP platform.

CIP Core

- This project focus to create reference minimal file system images that allow testing the CIP Core packages: a set of industrial-grade components that require super long-term maintenance.
CIP SLTS Kernel development (1/5)

CIP SLTS (linux-4.4.y-cip), Maintenance period 10 years and more (10-20 years)
- Official CIP SLTS kernel tree based on linux-stable.git
- Maintainer: Ben Hutchings
- Linux 4.4.83-cip8 released on 21st August 2017
CIP SLTS Kernel development (2/5)

- Kernel maintenance policy
  - [https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipkernelmaintenance](https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipkernelmaintenance)
  - Follow the stable kernel development rule as the basis
  - Feature backports are acceptable
    - All features has to be in upstream kernel before backport to CIP kernel
    - **CIP has “Upstream first” policy**
  - Validation will be done by CIP test infrastructure and/or members

- Current backported features on 4.4.y-CIP
  - Kernel Self Protection Project related features
    - Address Space Layout Randomization for user space process (ASLR)
    - GCC’s undefined behaviour Sanitizer (UBSAN)
    - Faster page poisoning
  - Board support patches for Renesas RZ/G and Siemens IoT2020
Next CIP SLTS kernel (tbd)

Mainline -> 4.4 -> Stable (linux-stable) -> CIP SLTS (linux-4.4.y-cip) -> Stable (linux-stable-x.y) -> NEXT CIP SLTS (TBD)

Backported patches

Maintained by Ben Hutchings

Feature backports

Stop backporting. Focus to security fix only

Approx. 2-3 years

Backported patches

Maintained by Ben Hutchings

Feature backports

Stop backporting. Focus to security fix only

Approx. 2-3 years
Out-of-tree drivers

- In general, all out-of-tree drivers are unsupported by CIP
- Users can use CIP kernel with out-of-tree drivers
  - If a bug is found in such a modified kernel, users will first demonstrate that it exists in the CIP kernel source release in order for the CIP maintainers to act on it.
CIP SLTS Kernel development (5/5)

• CIP Kernel maintenance (https://gitlab.com/cip-project/linux-cip)

• Ben Hutchings is the CIP Kernel maintainer.
  • CIP kernel v4.4.92-cip11 released on Oct 18th:
    • Regressions and security fixes that land in LTS.
    • Specific industrial grade hardware support.
  • Talk about CIP kernel at ELCE
    • Tue Oct 24th at 16:55
    • Congress Hall II
2. CIP SLTS real-time support (1/2)

CIP SLTS+PREEMPT_RT (will be separately maintained by CIP members)
- CIP kernel tree based on linux-stable-rt and patches from CIP SLTS
- Validation will be done by CIP
CIP SLTS real-time support (2/2)

• CIP has become a Gold Member of the Real Time Linux Project

• What’s next
  • Work together with the RTL Project
  • A CIP member is working to become the maintainer of 4.4.y-stable-rt, the base version of the CIP Kernel.

• More information
  • https://wiki.linuxfoundation.org/realtime/rtl/start
Milestones of CIP testing and current status

1. Board at desk - single dev
   • A setup that allows a developer to test the CIP kernel on the CIP selected hardware platform connected locally to her development machine using kernelCI tools.

2. CIP kernel testing
   • Test the CIP kernel on a regular basis and share the results with other CIP community members.

3. Define kernel testing as a service within CIP
   • Define the testing environment within CIP assuming that, in some cases, some members may share the tests, test results or laboratories while others may not.

4. From kernel testing to system testing
   • Once the testing environment has been ready and works for the kernel, explore how to extend it to the entire CIP platform.

https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting
CIP testing (2/3)

- CIP Testing project
  (https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting)

- B@D designed to:
  - Test Linux kernels and base systems.
  - Locally: no need of a centrally managed service.
  - On hardware connected to your dev machine.

- Latest status
  - CIP testing environment (B@D v1.0) just released
    (https://goo.gl/4RFrJ1)
    - Based on kernelci.org
    - Linux and Windows 10 as Host OS supported.
    - Shipped as a VM and Vagrant based environment.
    - Results and logs sharing capabilities.

- Check the source code involved
  - https://gitlab.com/cip-project/cip-testing/board-at-desk-single-dev/tree/master
Next Steps

• Discussing with AGL members for ALG and CIP testing collaboration

• During the coming months the team will focus on:
  • Defining how tests should look like.
  • Defining how results should be shared.
  • Increasing the test coverage of the CIP Kernel

• More updates at Embedded Linux Conference 2018
# CIP Core Packages (1/5)

## An example of minimal package set for CIP base layer

### Candidates for initial component set

<table>
<thead>
<tr>
<th>CIP Core Packages</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>CIP Kernel</strong></td>
<td></td>
</tr>
<tr>
<td>• Kernel</td>
<td></td>
</tr>
<tr>
<td>• Linux kernel + backported patches</td>
<td></td>
</tr>
<tr>
<td>• PREEMPT_RT patch</td>
<td></td>
</tr>
<tr>
<td>• Bootloader</td>
<td></td>
</tr>
<tr>
<td>• U-boot</td>
<td></td>
</tr>
<tr>
<td>• Shells / Utilities</td>
<td></td>
</tr>
<tr>
<td>• Busybox</td>
<td></td>
</tr>
<tr>
<td>• Base libraries</td>
<td></td>
</tr>
<tr>
<td>• Glibc</td>
<td></td>
</tr>
<tr>
<td>• Tool Chain</td>
<td></td>
</tr>
<tr>
<td>• Binutils</td>
<td></td>
</tr>
<tr>
<td>• GCC</td>
<td></td>
</tr>
<tr>
<td>• Security</td>
<td></td>
</tr>
<tr>
<td>• OpenSSL</td>
<td></td>
</tr>
</tbody>
</table>

### Keep these packages for Reproducible build

<table>
<thead>
<tr>
<th>Dev packages</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>• Flex</td>
<td></td>
</tr>
<tr>
<td>• Bison</td>
<td></td>
</tr>
<tr>
<td>• autoconf</td>
<td></td>
</tr>
<tr>
<td>• automake</td>
<td></td>
</tr>
<tr>
<td>• bc</td>
<td></td>
</tr>
<tr>
<td>• bison</td>
<td></td>
</tr>
<tr>
<td>• Bzip2</td>
<td></td>
</tr>
<tr>
<td>• Curl</td>
<td></td>
</tr>
<tr>
<td>• Db</td>
<td></td>
</tr>
<tr>
<td>• Dbus</td>
<td></td>
</tr>
<tr>
<td>• Expat</td>
<td></td>
</tr>
<tr>
<td>• Flex</td>
<td></td>
</tr>
<tr>
<td>• gawk</td>
<td></td>
</tr>
<tr>
<td>• Gdb</td>
<td></td>
</tr>
<tr>
<td>• Git</td>
<td></td>
</tr>
<tr>
<td>• Glib</td>
<td></td>
</tr>
<tr>
<td>• Gmp</td>
<td></td>
</tr>
<tr>
<td>• Gzip</td>
<td></td>
</tr>
<tr>
<td>• gettext</td>
<td></td>
</tr>
<tr>
<td>• Kbd</td>
<td></td>
</tr>
<tr>
<td>• Libibverbs</td>
<td></td>
</tr>
<tr>
<td>• Libtool</td>
<td></td>
</tr>
<tr>
<td>• Libxml2</td>
<td></td>
</tr>
<tr>
<td>• Mpclib</td>
<td></td>
</tr>
<tr>
<td>• Mpfr4</td>
<td></td>
</tr>
<tr>
<td>• Ncurses</td>
<td></td>
</tr>
<tr>
<td>• Make</td>
<td></td>
</tr>
<tr>
<td>• M4</td>
<td></td>
</tr>
<tr>
<td>• pax-utils</td>
<td></td>
</tr>
<tr>
<td>• Pciutils</td>
<td></td>
</tr>
<tr>
<td>• Perl</td>
<td></td>
</tr>
<tr>
<td>• pkg-config</td>
<td></td>
</tr>
<tr>
<td>• Popt</td>
<td></td>
</tr>
<tr>
<td>• Procps</td>
<td></td>
</tr>
<tr>
<td>• Quilt</td>
<td></td>
</tr>
<tr>
<td>• Readline</td>
<td></td>
</tr>
<tr>
<td>• syfsutils</td>
<td></td>
</tr>
<tr>
<td>• Tar</td>
<td></td>
</tr>
<tr>
<td>• Unifdef</td>
<td></td>
</tr>
<tr>
<td>• Zlib</td>
<td></td>
</tr>
</tbody>
</table>

**NOTE:** The maintenance effort varies considerably for different packages.

---

ELCE17, Prague, Czech Republic 37
CIP Core Packages (2/5)

Current status of the Base layer development

1. Define an initial component set
2. Define component version
3. Contribute to upstream project
4. Start maintenance for SLTS

1.5 Talk to open source communities
CIP Core

- **CIP Core is now become CIP official project**
  - CIP Core aims to provide a way to create and test installable images
- **Goal**
  - **Input:** Debian sources/binaries and cip kernel
  - **Build mechanism:** Bitbake and/or Debian build system
  - **Output:** Minimum deployable base system image for testing
- **Current status**
  - Minimal rootfs can be build for the following hardware
    - Renesas RZ/G1M (iwg20m)
    - BeagleBone Black
    - Cyclone-V
    - QEMUx86

**Source code:** [https://gitlab.com/cip-project/cip-core](https://gitlab.com/cip-project/cip-core)
Creating Debian-based image (Current status)

Deby: [https://github.com/meta-debian/meta-debian](https://github.com/meta-debian/meta-debian)
Creating Debian-based image (Other options)

ISAR: https://github.com/ilbers/isar
ELBE: https://elbe-rfs.org/

4 CIP Core Packages (5/5)
## Potential build tools for CIP Core (Comparison Elbe, Isar and Deby)

<table>
<thead>
<tr>
<th></th>
<th>Elbe</th>
<th>Isar</th>
<th>Deby</th>
<th>Remarks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Base system</strong></td>
<td><strong>Debian binary packages (no rebuilding)</strong></td>
<td>Binary packages <em>cross-built</em> from Debian <em>source</em> packages</td>
<td><strong>Bitbake</strong></td>
<td><strong>Poky</strong></td>
</tr>
<tr>
<td><strong>Build system</strong></td>
<td>Custom</td>
<td><strong>Bitbake</strong></td>
<td><strong>Poky</strong></td>
<td><strong>ELBE-XML for project description</strong></td>
</tr>
<tr>
<td><strong>Host tools</strong></td>
<td><strong>Debian</strong>: debootstrap, qemu, elbe-pbuilder</td>
<td><strong>Debian</strong>: multistrap, dpkg-buildpackage, qemu</td>
<td><strong>Poky</strong></td>
<td><strong>Common function to unpack Debian source packages</strong></td>
</tr>
<tr>
<td><strong>Metadata</strong></td>
<td>✔ ELBE-XML for project description</td>
<td>✔ Recipes for building product packages</td>
<td>✔ Recipes for image generation</td>
<td>✔ Common function to unpack Debian source packages</td>
</tr>
<tr>
<td><strong>Compilation</strong></td>
<td>Native</td>
<td><strong>Cross</strong></td>
<td><strong>Cross</strong></td>
<td>✔ Re-use Debian binaries and QA</td>
</tr>
<tr>
<td><strong>Benefits</strong></td>
<td>✔ Based on Debian packages (stability, long-term maintenance)</td>
<td>✔ Generate images by installing binary packages</td>
<td>✔ Manage multiple products as a custom setting (layers or configuration files)</td>
<td>✔ Based on Debian packages (stability, long-term maintenance)</td>
</tr>
</tbody>
</table>

Summary and conclusion
Summary

• The CIP Open Source Base Layer of industrial-grade software materializes

• CIP today focusses on
  • **Kernel maintenance:** maintaining Linux kernels for very long time (+15 years) including real-time support
  • **Testing:** providing a test infrastructure and evolve tests
  • **CIP Core packages:** a set of industrial-grade components that require super long-term maintenance including the required build tool chains
Conclusion

• Our Civilization needs an Open Source Base Layer of industrial-grade software
• CIP provides this, based on Linux
• Sustainability is ensured by
  • The backing of big industrial and semiconductor companies
  • Close cooperation with and build on mature Open Source projects (Debian, PREEMPT_RT, kernelci, …)
  • Providing elaborated tool chains
  • Ensuring in-depth tests
• CIP gets traction in the member companies
Contact Information and Resources

To get the latest information, please contact:

- CIP Mailing list: cip-dev@lists.cip-project.org

Other resources

- CIP Web site: https://www.cip-project.org
- CIP Wiki: https://wiki.linuxfoundation.org/civilinfrastructureplatform/
- CIP source code
  - CIP GitLab: http://www.gitlab.com/cip-project
CIP whitepaper

• Year One Update + Whitepaper Release
  • https://www.cip-project.org/blog/2017/05/31/cip-year-one-update-whitepaper-release

• Everyone can download the whitepaper
  • https://wiki.linuxfoundation.org/__media/civilinfrastructureplatform/whitepaper_short.pdf
CIP developers meeting/gathering at ELCE

• CIP developers meeting/gathering at ELCE which it will be immediately after the talk. Link: https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipconferences/cipwselce2017#cip-developers-meeting-gathering

• Date & Time: October 24th, 13:00-14:00

• Room: Liben Room, Mezzanine Level
Thank you!
Questions?