#

CIP User Manual

Table of contents

Overview

yet to update

Civil Infrastructure Platform

The Civil Infrastructure Platform (“CIP”) is a collaborative, open source project hosted by the Linux Foundation. The CIP project is focused on establishing an open source “base layer” of industrial grade software to enable the use and implementation of software building blocks in civil infrastructure projects. Currently, civil infrastructure systems are built from the ground up, with little re- use of existing software building blocks.

The CIP project intends to create reusable building blocks that meet the safety, reliability and other requirements of industrial and civil infrastructure.
CIP works closely with the upstream community and does not aim to create a new Linux distribution.

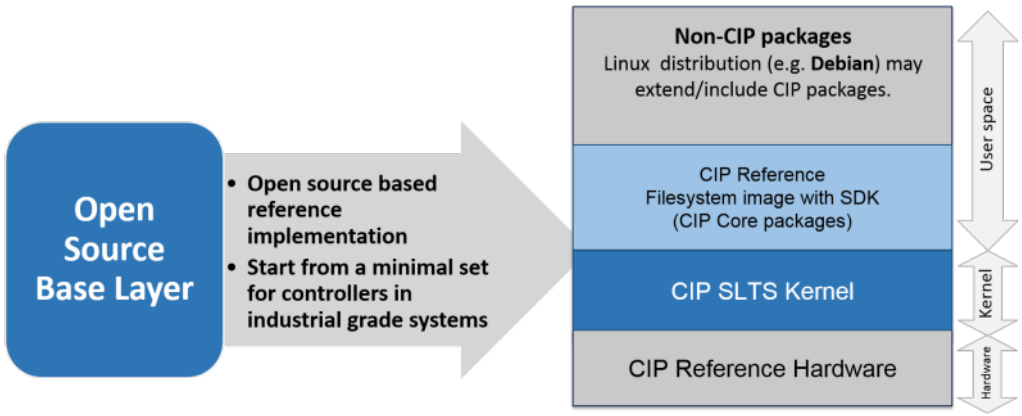
Reference Hardware

The CIP project has selected a number of hardware platforms to be used as reference platforms for the project’s software.

Refer the below link for CIP project reference hardwares
https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting/cipreferencehardware

Open Source Base Layer (OSBL)

OSBL is a set of industrial grade core open source software components, tools and methods. It is composed of the CIP kernel source code, and the CIP Core source packages.



CIP Kernel

CIP supports and maintains the kernel for a long time (+10 years).

Refer the below link for CIP kernel maintenance
https://wiki.linuxfoundation.org/civilinfrastructureplatform/cipkernelmaintenance

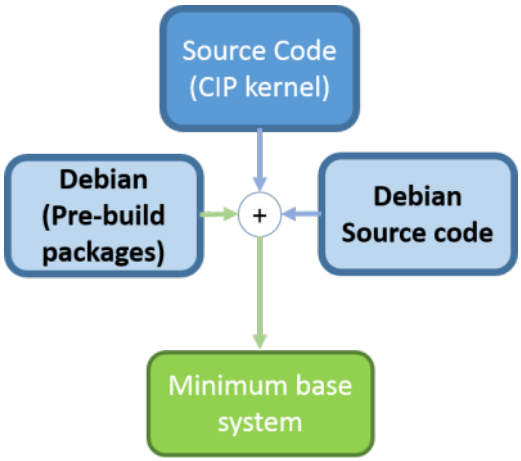
Currently supported CIP SLTS kernels

The current released CIP kernels are as follows.

Version	Maintainer(s)	First Release	Projected EOL
SLTS v4.19	Nobuhiro Iwamatsu & Pavel Machek	2019-01-11	2029-01
SLTS v4.19-rt	Pavel Machek	2019-01-11	2029-01
SLTS v4.4	Nobuhiro Iwamatsu & Pavel Machek	2017-01-17	2027-01
SLTS v4.4-rt	Pavel Machek	2017-11-16	2027-01

CIP core

CIP core provides example file system images using available build and image generation tools. It focuses on user land software and tools.
Currently, CIP is using meta-debian for Deby, and ISAR for isar-cip-core.



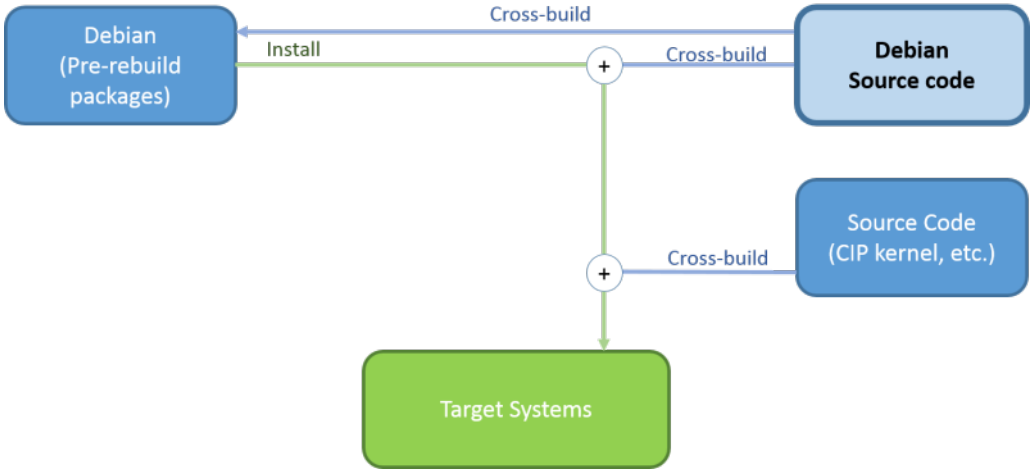
CIP core has two profiles:

- The tiny profile is built from Debian source code and is useful for devices with storage restrictions, extreme performance and flexibility requirements, and low-complexity applications.
- The generic profile is built from Debian binary packages and covers devices that require more functionality, have less performance and flexibility requirements, and more storage.

	Generic profile	Tiny profile
Approach	Binary packages	Source packages
Tool	Isar	Deby

Tiny profile(Deby)

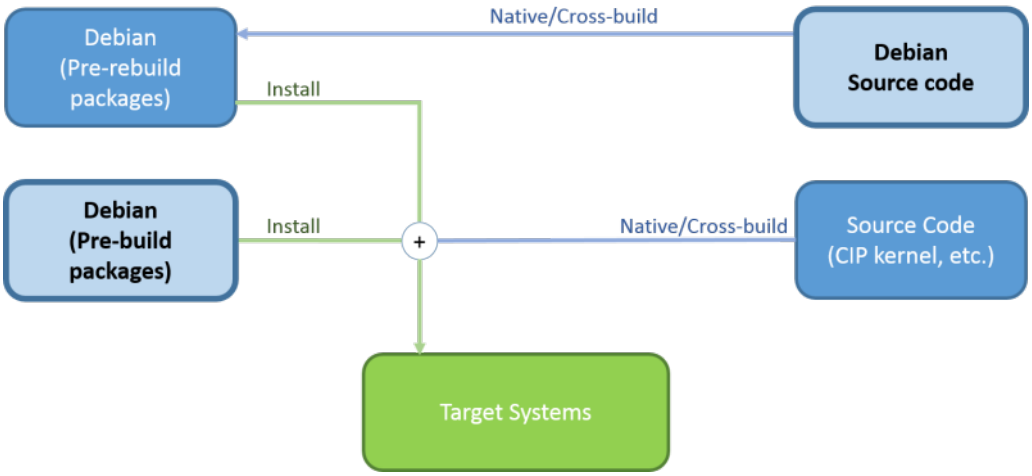
Deby is built with poky and meta-debian, a layer for the poky build system that allows cross-building file system images from Debian source packages. Deby does not use Yocto/OE source code.



Refer Deby CIP source repository at https://gitlab.com/cip-project/cip-core/deby

Generic Profile (isar-cip-core)

ISAR uses bitbake to generate the file system image by reusing Debian binaries and rebuilding packages that need modifications for the target board.



Refer ISAR CIP source repository at https://gitlab.com/cip-project/cip-core/isar-cip-core

Package list The list of CIP Core packages and the process to add or remove packages is described here.

How to Create CIP Images

ISAR CIP Core

Source repository:
https://gitlab.com/cip-project/cip-core/isar-cip-core

Refer below link for build steps of ISAR CIP core
https://gitlab.com/cip-project/cip-core/isar-cip-core/-/blob/master/README.md

Deby CIP Core

Source repository:
https://gitlab.com/cip-project/cip-core/deby

Refer below link for build steps of Deby CIP core
https://gitlab.com/cip-project/cip-core/deby/-/blob/cip-core-buster/README.md

Default User Accounts

Default user account is root.

Porting CIP for New Hardware

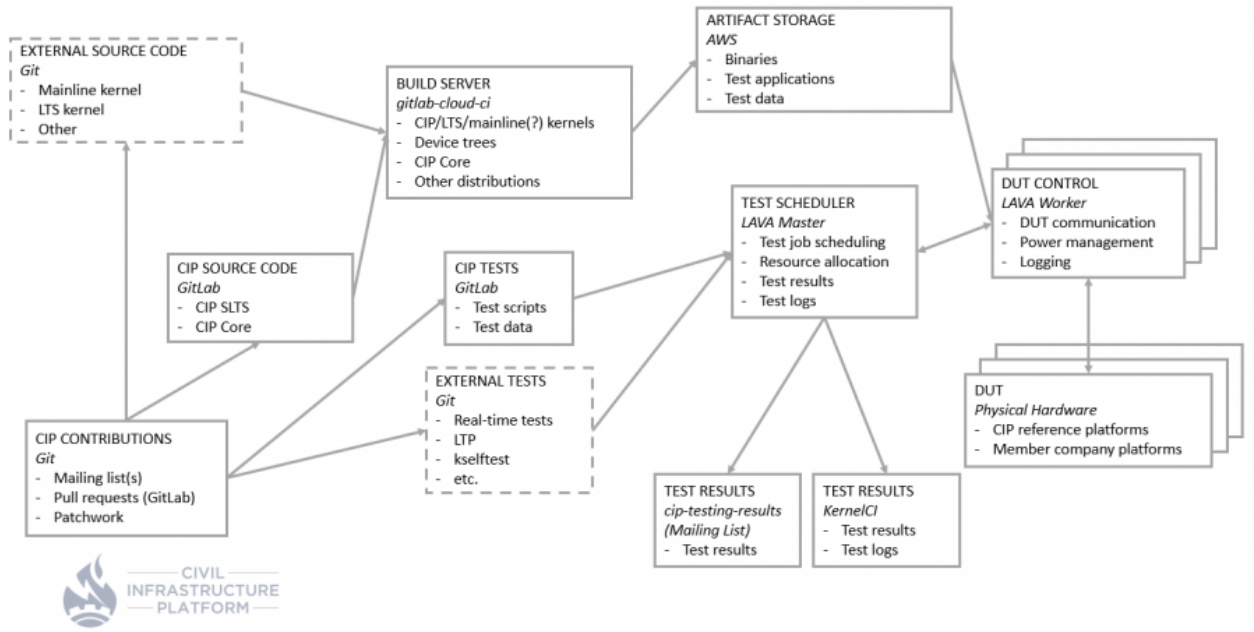
TBD

CIP Testing

- CIP uses B@D (Board at Desk) to run automated testing on local Beaglebone Black or Renesas RZ/G1M iwg20m platform.

- CIP's centralised testing can run tests without having local access to a platform and it is useful as list of reference platforms grows.
- CIP also uses Continuous Integration (CI) testing to automatically test CIP software on CIP hardware.

The block diagram below provides an overview of CIP's centralised test infrastructure.



LAVA Testing

CIP have set up their own instance of LAVA (Linaro Automated Validation Architecture). LAVA is a continuous integration system for deploying operating systems onto physical and virtual hardware for running tests.

Refer below link for more information on LAVA testing.
https://wiki.linuxfoundation.org/civilinfrastructureplatform/ciptesting/centalisedtesting/ciplava

Kernel CI Testing

yet to update

CIP Core Testing

yet to update

CIP Software Updates

CIP aims to provide super long term support and it is important for CIP to have a reference software update mechanism.

Current CIP software update uses Hawkbit server to store the swupdate related files. Client uses SWUpdate and librsync to communicate with Hawkbit.

It supports the following functions

Image types (You have to select 1 type)
- raw update
- binary delta update using librsync

Security options (You can enable both of them at the same time)
- signed update
- encrypted update

Refer below link for more details on software updates
https://gitlab.com/cip-project/cip-documents/-/blob/master/event/2019/sw_updates_wg_mini-summit.pdf

CIP System Monitoring

Yet to update ## Monitoring System Logs ## IPS & IDS Component Logs

References

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

Hawkbit: https://www.eclipse.org/hawkbit/
SWUpdate: https://sbabic.github.io/swupdate/index.html
LAVA: https://lava.ciplatform.org/