Intelligent Solutions

msc technologies
an Avnet Company

Engineering Leadership

MSC-LDK Manual
BSP 01040-ElkhartLake V1.9.0

2022-07-30

Copyright Notice

Trademarks

All used product names, logos or trademarks are property of their respective owners.

Certification

MSC Technologies is certified according to DIN EN ISO 9001:2000 standards.

Audience

This guide is intended for software developers already familiar with Linux development.

Life-Cycle-Management

MSC Technologies GmbH products are developed and manufactured according to high quality standards. Our life-cycle-management assures long term availability through permanent product maintenance. Technically necessary changes and improvements are introduced if applicable. A product-change-notification and end-of-life management process assures early information of our customers.

Disclaimer

Although this document has been generated with the utmost care no warranty or liability for correctness or suitability for any particular purpose is implied. The information in this document is provided "as is" and is subject to change without notice.

Contents

List of Figures

List of Tables

Listings

1. History

1.1. Document

Revision	Changes
1.9.0	• No changes
1.8.0	• No changes
1.7.0	• No changes
1.6.1	• Added chapter “Qt5”
1.6.0	• Added chapter “Hardware monitoring sensors” • Added chapters “Enabling WiFi” and “Use the board as an access point” • Various fixes and cleanups
1.5.0	• Added chapter “I2C Bus Identification” • Added chapter “Reduced Cleanup Time” • Various fixes
1.4.0	• Removed chapter “Microsoft Azure”
1.3.0	• Added chapter “Power Analysis” • Added chapter “MSC-IO” • Added section “Setup Optional Docker Container”
1.2.0	• Added chapter “Microsoft Azure” • Added chapter “Real-Time” • Added chapter “Security” • Added chapter “Tips And Tricks” • Added section “ConnMan Configuration” • Updated section “Deploying Images To The Hardware” (new image type .wic) • Updated section “Enhancing The Images” • Updated section “Installation Of MSC-LDK”
1.1.0	• Added section “Traceable and Reproducable Images”

Revision	Changes
1.0.0	• Added section “Using LXQt” • Added section “Using the SDK images” • Added section “Bug Reporting”

1.2. MSC-LDK

Revision	Changes
1.9.0	• Updated to yocto 3.3 (hardknott) • Various fixes in MSC tools • Setup can download layer repositories relative to it's own origin (–repos-relative-to-this) • Added msc-system-info
1.8.0	• Updated to yocto 3.2 (gatesgarth)
1.7.0	• Updated to yocto 3.1.2 (dunfell) • Added support for ElkartLake and TigerLake BSPs • Various fixes in MSC tools
1.6.1	• `setup.py` adds wrappers for some Yocto tools in the build directory, e.g. `./bitbake` • `make install` has option `CLEAN_DESTDIR=y` which purges the installation directory before installing new files • `make install` installs necessary files to use NXP's manufacturing tool `uuu` • Using `openssh` instead of `dropbear` • Fixed terminal's size being printed on serial line after login into slow systems • Various fixes in MSC tools
1.6.0	• Updated to yocto 3.0.1 (zeus) • Added layers meta-chrome-instead-of-otter and boot2qt • Added support for Ryzen and .iMX8 BSPs • "make install" copies .wic.bmap files for use with bmaptool
1.5.0	• Updated to yocto 2.7 (warrior) • Added layer-marvell • Removed meta-iot-cloud and meta-java dependencies for msc-image-lxqt • Added layer meta-bsp-arm-verification • Various improvements to build MSC-LDK BSPs in a Docker container • Various improvements to build feature branches of the BSPs
1.4.0	• Updated to yocto 2.5 (sumo) • Added layer-marvell • Removed meta-iot-cloud and meta-java dependencies for msc-image-lxqt

Revision	Changes
1.3.0	• Updated to yocto 2.4 (rocko) • ApolloLake, Braswell, Skylake and Baytrail BSPs have been merged to IntelCombined BSP • Added layers-debug for simplify kernel debugging • Various fixes for building MSC-LDK when /bin/sh is dash instead of bash. • Added experimental setup option –with-tmp-image to speed up cleanup of build directories • Uses MSC-IO v3 to support ARM and setup MSC-IO and EAPI by ACPI BIOS entries • zsh is used as login shell • Four instead of one text consoles are available (access with Alt-F1...Alt-F4) • On x86 the port for the serial console is guessed instead of hardcoded /dev/ttyV0 which might be unavailable. • Various tool changes and smaller bug fixes.
1.2.0	• Updated to yocto 2.2 (morty) • All kernels are compiled with CONFIG_DYNAMIC_DEBUG for easier hardware bring-up • X desktop is run as user “msc” instead of “root” • Supporting real-time kernel • LXQt is the preferred desktop environment instead of sato • setup supports –bsp-build-dir-name • setup supports –layers-security • setup supports installation of 3rdparty layers • Builds of different BSPs don’t share the sstate-cache any longer • MSC-LDK repository has been relocated
1.1.0	• Using setup.py instead of setup.sh • setup.py supports configuration of the BSP exactly as a previous image has been built (traceable and reproducible images) • Added certified azure libraries to standard image • Supporting speaking names for BSPs, e.g. Q7-BT or Baytrail instead of C984 • Added fallback mirror server ftp4.ebv.com

Revision	Changes
1.0.0	• Updated to yocto 2.0 (jethro) • Added layer lxqt. • Simplified adding additional layers to the BSP builds. • Simplified adding new BSPs. • Moved recipes into own layers (e.g. meta-msc-ldk-core.git). • Added support for “develop” and feature branches of MSC-LDK development.

1.3. BSP 01040-ElkhartLake

Revision	Changes
1.9.0	• Updated to kernel 5.14.0
1.6.0	Initial

2. Introduction

This document is intended to be used by developers creating or adapting Linux systems for MSC Technologies hardware with the MSC Linux Development Kit. The MSC-LDK provides an environment to create Linux kernels, bootloaders and root filesystems. It is based on the Yocto 3.3 project.

2.1. Scope

This document gives a hand in:

- Setup of the MSC-LDK.
- Building the Linux kernel, bootloaders and root filesystems.
- Deploying images to the hardware.
- Using the hardware features with Linux.

2.2. Out Of Scope

Detailed information about Yocto is not part of this document but available at:
https://www.yoctoproject.org.

2.3. Features

Features of the MSC-LDK are:

- Strong versioning. Every package is defined by it's version before downloading and building it.
- Support for different image installations (e.g. USB, SATA, RAMDISK).
- Support for different image types based on one configuration (headless or with GUI).
- Everything can be build from sources. But a cache is provided so already built packages can be reused.

2.4. Conventions

This section describes the conventions used in this manual.

Warning: This format is used to highlight material involving possibility of injury or equipment damage.

Caution: This format is used to highlight information that will help you prevent equipment failure or loss of data.

Note: This format is used to highlight information of importance or special interest.

Link: Look also at the given page or chapter for additional informations for the specific topic.

Typographical conventions:

`Courier New 9Pt`	Screen text, user-typed command-line entries, or source code.
[Ctrl]+C	Two or more keys that must be pressed simultaneously.

2.5. Product Support

MSC engineers and technicians are committed to provide support to our customers whenever needed. If the information provided there does not solve your problem, please contact our Technical Support:
WWW: http://www.msc-technologies.eu/de/support.html

3. Getting Started

3.1. Requirements

- Linux x86 development host (32bit or 64bit).
- Ubuntu 18.04 (LTS), but other distributions up to Ubuntu 19.04 also work. Note that Yocto also allows Ubuntu 16.04 for zeus, but MSC discovered problems with some -native packages, e.g. zsh.
- Internet access for downloading packages (HTTP, FTP, Git and SSH).
- Registration on the MSC Git server.
- Lots of free disc space for the initial build (>128 GB).
- Python3 with 'pip' installed (at least Python v3.3).

3.2. Registration On The MSC Git Server

Downloading any files from the MSC Git server requires a registration on:
http://www.msc-technologies.eu/register.html.

Registered user may apply to specific Git repositories here by sending an email with their public SSH key and desired project name to mailto://support@msc-technologies.eu

3.2.1. Creating An SSH Key

If there is no SSH key already created (~/.ssh/id_rsa.pub), it can be generated like this. Press "Enter" on passphrase).

```
user@devhost:$ ssh-keygen -t rsa
Generating public/private rsa key pair.
Created directory '/home/user/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/user/.ssh/id_rsa.
Your public key has been saved in /home/user/.ssh/id_rsa.pub.
The key fingerprint is:
f3:f0:17:08:58:96:25:f4:bb:c1:60:f4:61:20:c8:b3 user@host
The key's randomart image is:
+--[ RSA 2048]----+
|    . ...B++     |
|     +  *.= .    |
|      o. + o     |
|     E  . + o    |
|        S = .    |
|         = o .   |
|          + .    |
|           .     |
|                 |
```

Share the public key in ~/.ssh/id_rsa.pub with MSC during Git registration.

The private key `~/.ssh/id_rsa` should never be shared with somebody else.

The SSH key must not have a passphrase. It will be used in background communication and therefore there is no possibility to enter the passphrase. Trying to fetch repositories from the MSC Public GIT Server would fail with no hint that the passphrase is missing.

3.2.2. Configuring For HTTP proxies

Some source files will be downloaded from HTTP servers. If a proxy must be used, two environment variables have to be set.

```
export http_proxy=http://my-proxy:3128
export https_proxy=http://my-proxy:3128
```

4. Using MSC-LDK

4.1. Yocto/MSC-LDK Terms

Yocto and therefore MSC-LDK uses a sophisticated approach to generate Linux images.

- A *target* is the hardware or CPU module on which the generated Linux software is to be run.
- An *image* contains all the files necessary for execution by the targeted hardware, e.g. the Linux kernel and the root filesystem.
- Software that is part of a Linux image is called a *package*.
- A package is generated from sources by a *recipe*, which is a description of where to download the sources and how to compile them within Yocto.
- A *layer* is a collection of recipes. Layers are stackable and can extend recipes defined in other layers.
- A *BSP* provides the necessary layers to MSC-LDK to support the target's hardware.
- MSC-LDK is mainly an installer of Yocto, MSC specific layers and BSP layers.

4.2. Setup Optional Docker Container

As Yocto and MSC-LDK use some host tools, a docker container is available for simplified installation.

This description assumes that the docker container will be used only on a single-user workstation. It must not be used on a multi-user server as **YOUR** private SSH keys are installed in the image.

If a proxy has to be used to access the network, the file `Dockerfile` needs to be adjusted. Uncomment these lines and fill in the IP address and port of the proxy:

```
#ARG http_proxy=http://<ip>:<port>
#ARG https_proxy=http://<ip>:<port>
#ENV http_proxy ${http_proxy}
#ENV https_proxy ${http_proxy}
#RUN echo "Acquire::http::Proxy \"${http_proxy}\";" >/etc/apt/apt.conf.d/80proxy && \
#    echo "Acquire::https::Proxy \"${https_proxy}\";" >>/etc/apt/apt.conf.d/80proxy
```

e.g. to:

```
ARG http_proxy=http://172.23.75.27:3128
ARG https_proxy=https://172.23.75.27:3128
ENV http_proxy ${http_proxy}
ENV https_proxy ${http_proxy}
RUN echo "Acquire::http::Proxy \"${http_proxy}\";" >/etc/apt/apt.conf.d/80proxy && \
    echo "Acquire::https::Proxy \"${https_proxy}\";" >>/etc/apt/apt.conf.d/80proxy
```

To install the container:

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/docker-msc-ldk
user@devhost:$ cd docker-msc-ldk
user@devhost:$ git checkout v1.9.0
```

```
user@devhost:$ mkdir -p src && \
    rm -rf rootfs/home/.ssh && \
    mkdir -p rootfs/home/.ssh && \
    cp ~/.ssh/id_rsa rootfs/home/.ssh && \
    cp ~/.ssh/id_rsa.pub rootfs/home/.ssh && \
    docker build -t=msc-ldk . && \
    rm -rf rootfs/home/.ssh
```

To use the container on an Ubuntu host:

```
user@devhost:$ docker run --privileged -t -i \
    --dns $(nmcli -f 'IP4.DNS' -m multiline device show 2>&1 | sed -rn 's/IP4.DNS\[1\]: *(.*)
        /\1/p') \
    --name msc-ldk \
    -h docker \
    -v `pwd`/src:/src \
    msc-ldk \
    /bin/bash
```

To use the container on a CentOS host:

```
user@devhost:$ docker run --privileged -t \
    --dns $(sed -rn '0,/nameserver/ s/nameserver (.*)/\1/p' /etc/resolv.conf) \
    --name msc-ldk \
    -h docker \
    -v `pwd`/src:/src \
    msc-ldk \
    /bin/bash
```

Inside the docker container MSC LDK has to be cloned as described in Installation Of MSC-LDK. Inside the cloned directory BSPs can be installed and built as shown in sections Installation Of MSC-LDK BSPs and Building Images, respectively.

When the docker container is no longer used, don't forget to release its resources:

```
user@devhost:$ docker stop msc-ldk
user@devhost:$ docker rm msc-ldk
```

4.3. Installation Of MSC-LDK

The MSC-LDK must be installed on a partition with at least 128 GB free space. As a lot of source files will be accessed, it is recommended to use an EXT4 partition with the mount options `noatime,nodiratime` set.

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/msc-ldk
user@devhost:$ cd msc-ldk
user@devhost:$ git checkout v1.9.0
```

No files will be installed in other directories.

NOTE: some scripts of the recipes use an 'echo -e <somewhat>' command. bitbake calls the buildscripts with /bin/sh as shell. If your hostsystem uses bash as sh everything works fine. But if a shell with less functionality like dash is used, it is necessary to setup bash as sh. This can be done on most debian derivated systems by:

```
user@devhost:$ sudo dpkg-reconfigure dash
```

The question has to be answered with "no"

4.4. Directory Layout

Table 4.1. – MSC-LDK Directory Layout

Directory	Contents
`build/01040*`	BSP build directory.
`doc`	This contains all the generated Yocto documentation. (only if build with: make doc)
`downloads/`	All downloaded sources are stored here. This directory can be shared with other MSC-LDK installations.
`scripts/`	Build helper scripts.
`sources/01040/*.git`	BSP specific layers. (will be created in chapter 4.5)
`sources/meta-freescale.git/`	Freescale specific layers, only on Freescale architectures.
`sources/meta-imx.git/`	i.MX specific layers, only on Freescale i.MX architectures.
`sources/meta-intel-*.git/`	Intel specific layers, only for Intel architectures.
`sources/meta-msc-ldk-*.git/`	MSC-LDK layers.
`sources/meta-openembedded-*.git/`	Additional useful tools that are not part of Yocto.
`sources/meta-qt5*.git/`	Qt5 layers used by msc-image-lxqt.
`sources/yocto.git/`	Yocto sources used by MSC-LDK.
`sstate-cache/`	All built files are stored here and reused on next build. This directory can be shared with other MSC-LDK installations.

4.4.1. MSC-LDK Layers

MSC-LDK consists of several layers. Only the required layers will be activated for the Yocto build process.

Figure 4.1. – MSC-LDK Layers

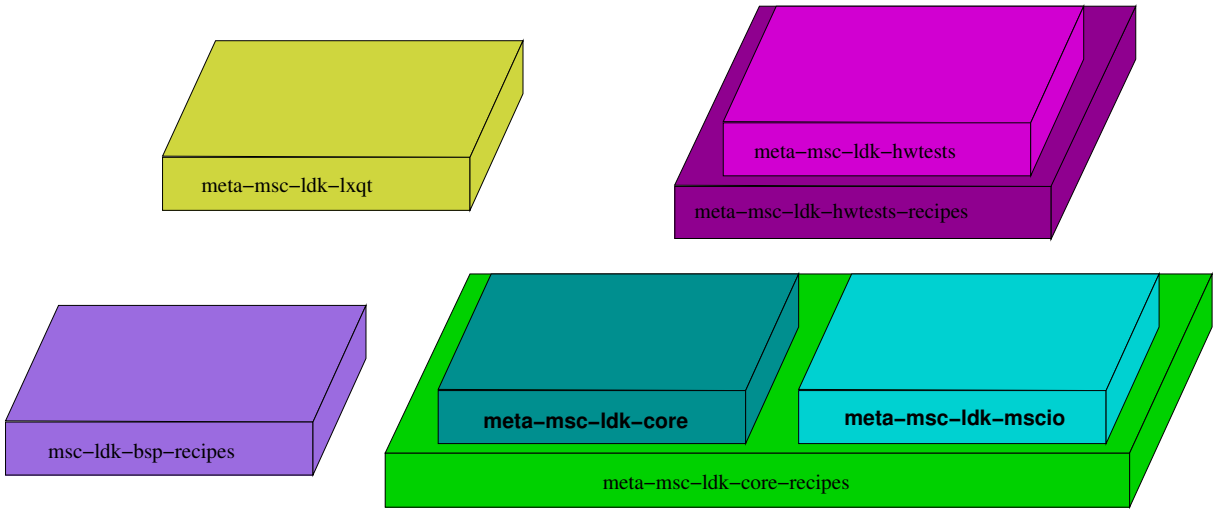


Table 4.2. – MSC-LDK Layer Description

Layer	Description
`msc-ldk-bsp-recipes`	BSP related layer, e.g. Kernel patches for a board. Mandatory for a given board.
`meta-msc-ldk-core-recipes`	Required recipes to build MSC specific applications.
`meta-msc-ldk-core`	MSC specific image recipes and package groups. Depends on `meta-msc-ldk-core-recipes`.
`meta-msc-ldk-hwtests-recipes`	Hardware test specific recipes.
`meta-msc-ldk-hwtests`	Board specific image msc-image-hwtests-<BOARD>. Depends on `meta-msc-ldk-hwtests-recipes`.
`meta-msc-ldk-mscio`	MSC-IO layer. Depends on `meta-msc-ldk-core-recipes`.
`meta-msc-ldk-lxqt`	Support for the Lightweight Qt Desktop Environment LXQt (`http://lxqt.org`).

4.5. Installation Of MSC-LDK BSPs

`setup.py` is used to download the required Yocto layers (they will be installed into the sources subdirectory):

This BSP can be built only for 64bit.

4.5.1. 64bit Target Linux

To generate 64bit executables for the CPU module, do:

```
user@devhost: $ ./setup.py --bsp=01040
```

4.6. BSP Aliases

`setup.py` supports one of the following aliases for the option `--bsp=01040,ElkhartLake,Q7-EL,C6C-EL,SM2S-EL`

4.7. Reduced Cleanup Time

Building MSC-LDK images from scratch will create a lot of files in `build/*/tmp/`, for example the sources, object files and executables. Deleting them will take at least 30 minutes. This has a lot of impact on continuous integration build systems. To reduce the cleanup to a few seconds, MSC-LDK can place the `tmp` directory in an image file with `setup.py's` option `-with-tmp-image`. Building then takes only a few percent more while cleaning up the build directory is only a matter of seconds. As the image file is a sparse file, no more space is effectively being used than building it directly even when the image size is listed as 512 GiB.

To use the `tmp` directory manually, run `make mount_tmp_image` and `make umount_tmp_image` in the build directory.

This feature requires sudo permissions for at least the commands `mount`, `chown` and `umount`.

4.8. Building Images

To build all supported images for the installed BSP, do:

```
user@devhost:msc-ldk$ ./setup.py --bsp=01040
user@devhost:msc-ldk$ cd build/01040
user@devhost:01040$ make
Loading cache: 100% |############################| ETA:  00:00:00
Loaded 2703 entries from dependency cache.
[...]
```

Depending on the internet connection and the development host a first build may take several hours. To speed it up on further installations, share the directories `downloads` and `sstate-cache`.

All generated images can be collected in a specific directory with:

```
user@devhost:msc-ldk$ make install_images DESTDIR=/tmp/msc-ldk-images
```

Yocto tools like bitbake or bitbake layers can be used directly by calling them within the build directory with `./` prefix, e.g. `./bitbake`. They can also be called via the wrapper script `./build.sh` instead, e.g. `./build.sh bitbake`.

4.9. Traceable And Reproducable Images

One of the key features of Yocto is the strong versioning of the resulting images. Each package uses a predefined version, e.g. busybox 1.32.0. When compiling an image, yocto also prints the used GIT layer versions, e.g.

```
BB_VERSION            = "1.50.0"
BUILD_SYS             = "x86_64-linux"
NATIVELSBSTRING       = "universal"
TARGET_SYS            = "aarch64-poky-linux"
MACHINE               = "sm2s-imx8"
DISTRO                = "poky"
DISTRO_VERSION        = "3.3"
TUNE_FEATURES         = "aarch64 armv8a crc cortexa72-cortexa53 crypto"
TARGET_FPU            = ""
meta-poky
meta-yocto-bsp        = "hardknott-5.10.35-2.0.0-msc:58
    cbdaecf75b0248f96780b6882e8d4f232d038a"
meta                  = "develop:5e0a9f47096075b8783bf4193c14f27b6b5d9a21"
meta                  = "hardknott-5.10.35-2.0.0-msc:58
    cbdaecf75b0248f96780b6882e8d4f232d038a"
meta-oe
meta-networking
meta-python           = "hardknott-imx-5.10.35-2.0.0-msc:72236970
    e95af9fd7e98a49491de8e269d593f36"
meta-python2.git      = "hardknott:810d6d842f103eb59f18b06426106462b15de7e2"
meta-qt5.git          = "hardknott-imx-5.10.35-2.0.0:
    a00af3eae082b772469d9dd21b2371dd4d237684"
meta
meta-integrity
meta-signing-key
meta-tpm
meta-tpm2             = "hardknott-msc:ba04fbba711217e2a7c10270fe932146564f6229"
meta-msc-ldk-core-recipes.git = "develop:89435982a08b4f9f04039e8c4c67722022303087"
meta-msc-ldk-core.git = "develop:deea84b114e34e83db96f9dfdadab2bf1cd6d996"
meta-msc-ldk-mscio.git = "develop:d82e6498f8107e4cab9147c719214dc80254730f"
meta-freescale.git    = "hardknott-5.10.35-2.0.0:80
    dbe4bd63bd537fc9cfda2e009f8543464b4698"
meta-freescale-distro.git = "hardknott-5.10.35-2.0.0:916
    df6d24c0a33a3b1533bde70b6a2724ec77af4"
meta-bsp
meta-sdk
meta-ml               = "hardknott-5.10.35-2.0.0-msc:
    e9dd56937e4a32da92c11d983025694cfb9db18e"
meta-nxp-demo-experience.git = "hardknott-5.10.35-2.0.0:46107357
    abd2d2da9ffd702c87fce3984a422435"
meta-msc-arm-extensions.git = "msc-hardknott:
    cb439ec96b2ad749c38918410b026019a1e48ed6"
meta-msc-ldk-marvell.git = "develop:8200c219ae0cf263f11b88737d36ae9bd7304ea9"
meta-multimedia
meta-gnome            = "hardknott-imx-5.10.35-2.0.0-msc:72236970
    e95af9fd7e98a49491de8e269d593f36"
```

For further improvement, MSC-LDK has these additional features to recreate the image **after** it has been built and shipped:

- The used layers and the setup line how the BSP was configured is stored in the image's file `/etc/version_layer`. After compilation, the file can be also found in the build directory under `tmp/work/01040-ElkhartLake-poky-linux/<ImageType>/1.0-r0/rootfs/etc` Replace *<ImageType>* with your image type, e.g. `msc-image-sato`.
- The setup tool allows to checkout exactly these layers and configure the BSP as before. To use it, call setup.py with only one argument `--version-file`, e.g.

```
./setup.py --version-file ~/version_layer
```

Modifications of `conf/local.conf` are not yet traced.

This will checkout exactly the versions used by `version_layer`. It is then no longer possible to use `scripts/update.py` to pull the latest changes on the branch. A fresh checkout of MSC-LDK is necessary. The directories `downloads` and `sstate-cache` can be moved or copied to improve build speed.

Time stamps in the image will be updated, e.g. in /etc/issue.

4.10. Building Documentation

To build all Yocto specific documentation, do:

```
user@devhost:msc-ldk$ make doc
```

It will be stored within `doc/`.

The documentation is also available online on the Yocto homepage:
https://www.yoctoproject.org.

4.11. Image Types

To suit different use-cases of the MSC-LDK, various image types with different package selections are provided.

4.11.1. Headless

The headless image `msc-image-base` contains only console and framebuffer support without any X11 based GUI.

It can be selectively built by:

```
user@devhost:msc-ldk$ cd build/01040*
user@devhost:msc-ldk/build/01040*$ make msc-image-base
```

msc-image-base and msc-image-sato currently don't support predictable network names. Therefore no IP addresses are assigned by DHCP. To change this:

1. Disable predictable network interface names with the kernel command line option `net.ifnames=0`
2. Adjust the lines `iface ethX inet dhcp` in `/etc/network/interfaces` and replace `ethX` with the predictable network interface names listed by `ifconfig`, e.g. `enp1s0`.

4.11.2. LXQt

The image `msc-image-lxqt` uses the lightweight Qt Desktop environment with Qt 5 instead of SATO. It also includes the HTML5 capable web browser Otter Browser. To use it, the command line option `--layers-lxqt` must be added when executing `setup.py`, e.g.

It can be selectively built by:

```
user@devhost:msc-ldk$ ./setup.py --bsp=01040 --layers-lxqt
user@devhost:msc-ldk$ cd build/01040*-lxqt
user@devhost:msc-ldk/build/01040*-lxqt$ make msc-image-lxqt
```

4.11.3. LXQt SDK

The image `msc-image-lxqt-sdk` extends LXQt with the necessary tools to build Qt5 applications directly on the target. It is not built automatically.

It must be built by:

```
user@devhost:msc-ldk$ cd build/01040*-lxqt
user@devhost:msc-ldkbuild/01040*-lxqt$ make msc-image-lxqt-sdk
```

4.11.4. Sato

The image `msc-image-sato` contains the X11 window manager Sato with some Qt applications.

It can be selectively built by:

```
user@devhost:msc-ldk$ cd build/01040*
user@devhost:msc-ldk/build/01040*$ make msc-image-sato
```

For network configuration see this workaround

4.11.5. Sato SDK

The image `msc-image-sato-sdk` contains the X11 window manager Sato with some Qt applications and development tools like gcc compilers or git to build applications on the target. It is not built automatically.

It must be built by:

```
user@devhost:msc-ldk$ cd build/01040*
user@devhost:msc-ldk/build/01040*$ make msc-image-sato-sdk
```

4.12. Deploying Images To The Hardware

Any image type can be installed the same way as described here with base.
Just replace `base` (for headless) in the filename with `lxqt` (LxQt desktop), `sato` (standard Yocto GUI) or `qt5` (qt5/weston).

4.12.1. USB

Using .hddimg Images

To create a bootable USB stick with a modifyable root filesysystem, copy the image `msc-image-base-intel-corei7-64.hddimg` to an USB stick (e.g. on `/dev/sdh`) with:

```
sudo dd if=tmp/deploy/images/intel-corei7-64/msc-image-base-intel-corei7-64.hddimg of=/dev/sdh1
```

Then insert the stick into the target and reset it. The bootloader will then display a menu with the options `boot` and `install`. If no entry is selected within 1 second, the system will automatically boot into the live-system `boot`. Automatically booting can be interrupted by pressing the *Tab* key.

The hddimg file is configured to have about 10 MiB free space available independent of the size of the USB stick.

When selecting `install` there might be various messages `udevd: failed to execute /etc/udev/scripts/mount.sh '/et/udev/scripts/mount.sh': No such file or directory`. They can be ignored.

The hddimg file must be copied onto the first partition (`sdh1`) and not the whole disk (`sdh`), otherwise the BIOS might not recognize the card.

Use .wic Images

Alternatively instead of `.hddimg` images, `.wic` images can be copied to an USB stick. These images don't feature an installation option. And unlike the `.hddimg` images where the rootfs is mounted via loopback from the image file `rootfs.img` the root filesystem is mounted directly from the stick. So they are a better fit to use the image right ahead.

The `.wic` images use GPT for partitioning (https://en.wikipedia.org/wiki/GUID_Partition_Table). One property of the partitioning style is, that some meta data is expected at the end of the disc. Often is an image copied to a larger disc. This causes error messages in the kernel log when the Linux kernel mounts the partition:

```
# dmesg | grep GPT
[    6.345504] GPT:Primary header thinks Alt. header is not at the end of the disk.
[    6.355685] GPT:2169825 != 7708671
[    6.361382] GPT:Alternate GPT header not at the end of the disk.
[    6.370027] GPT:2169825 != 7708671
[    6.375741] GPT: Use GNU Parted to correct GPT errors.
```

The script `fix-gpt-error` can be used on the development host to correct this problem after using dd to copy the image to the USB stick:

```
user@devhost:msc-ldk$ sudo dd if=tmp/deploy/images/intel-corei7-64/
    msc-image-base-intel-corei7-64.wic of=/dev/sdh
user@devhost:msc-ldk$ scripts/fix-gpt-error /dev/sdh
[sudo] password for <user>:
'Alternate GPT header not at end of disk' problem fixed for '/dev/sdh'.
```

4.12.2. SD Card

The image `msc-image-base-intel-corei7-64.hddimg` can also be used for SD cards. See section USB for installation instructions.

4.12.3. SATA

Proceed as described in section USB but select the menu `install`. Follow the instructions to install the Linux system to the SATA drive.

4.12.4. Other

Other possible deployment types are:

- For booting from DVD or other read-only media, use the images with the extension `.iso`.
- For booting the root fileysystem as a ram disk or to manually install it into a partition, use the images with the extension `.cpio.gz`.

4.13. Login

Login is enabled via console or serial console (/dev/ttyS0, 115200 baud/8 bits/no parity). The headless and Sato images also have telnet login enabled.

Table 4.3. – User Accounts

Account	Password	Comment
root	mscldk	LXQt/Sato GUI require no login. No password is necessary for hardware test image.
msc	msc	Standard user with sudo permissions.

4.14. Enabled Services

Enabled services are:

- telnetd
- Serial console on /dev/ttyS0 with 115200 baud/8 bits/no parity

4.15. Enhancing The Images

4.15.1. Finding New Packages

Additional tools or functionalities that are needed inside the BSP can be included in the image by adding one or several packages that provide those tools. To identify the needed package, search publicly available recipes at
https://layers.openembedded.org/layerindex/branch/master/recipes/

For each recipe the containing layer is listed. If this layer is not already included in the MSC LDK, it has to be added according to Adding Own Layers. See Traceable And Reproducable Images for instructions on how to list the layers used in your BSP.

The package can then be added to the image by following the instructions in Adding Other Packages.

4.15.2. Adding Other Packages

Further packages can be included in the images by adding these lines to:
`build/01040*/conf/local.conf`:

```
IMAGE_INSTALL_append = " \
   my-foo-package \
"
```

Further information can be found here:
http://www.yoctoproject.org/docs/3.3/ref-manual/ref-manual.html#var-IMAGE_INSTALL

This can be automatized by calling `setup.py` with the argument `--local-conf` and an existing file whose content should be appended to `build/01040*/conf/local.conf`. A non-existing file is silently ignored.

```
user@devhost:msc-ldk$ cat << _EOF >conf.append
IMAGE_INSTALL_append = " \
   my-foo-package \
   "
_EOF

user@devhost:msc-ldk$ ./setup.py --bsp=01040 --local-conf-append=conf.append
```

4.15.3. Adding Own Layers

The images can be further enhanced or configured by adding own layers. Extend the file `build/01040*/conf/bblayers.conf` by these lines:

```
BBLAYERS += " \
  /home/user/my-own-msc-ldk-layer/ \
"
```

Then create the layer as described here:
http://www.yoctoproject.org/docs/3.3/dev-manual/dev-manual.html#creating-your-own-layer

This can be automatized by calling `setup.py` with the argument `--add-layer` and the layer's URL. The layer will then be downloaded to `sources/addons/`

```
user@devhost:msc-ldk$ ./setup.py --bsp=01040 --add-layer=https://github.com/OSSystems/meta-browser
```

4.15.4. Network Configuration

Network Interface Names

MSC-LDK uses predictable network interface names (https://www.freedesktop.org/wiki/Software/systemd/PredictableNetworkInterfaceNames/). This setup has the advantage that the network interfaces have the same name on every boot of the system.

To switch back to the old network interface names (eth0, ...), there are two ways:

1. Pass the kernel command line parameter `net.ifnames=0`
2. Create an empty file `/etc/udev/rules.d/80-net-name-slot.rules` (this will overrule `/lib/udev/rules.d/80-net-name-slot.rules`)

System Wide Proxy Configuration

The package proxy-config installs the script `/etc/profile.d/proxy.sh`.

The content of proxy.sh can be configured by setting variables in `build/01040*/conf/local.conf`:

```
NO_PROXY     = "localhost,127.0.0.0/8"
HTTP_PROXY   = "http://proxy.server.com:3128"
HTTPS_PROXY  = "http://proxy.server.com:3128"
FTP_PROXY    = ""
SOCKS_SERVER = ""
```

4.15.5. Time Zone Setup

The default time zone is set to `Europe/Berlin`. The time zone can customized by setting the Yocto variable `DEFAULT_TIMEZONE` in `build/01040*/conf/local.conf`. To set the time zone to Rome, use the following:

```
DEFAULT_TIMEZONE = "Europe/Rome"
```

The available time zones can be found in `/usr/share/zoneinfo`.

The time zone is entered in `/etc/timezone` and there is a symbolic link: `/etc/localtime->/usr/share/zoneinfo/Europe/Berlin`.

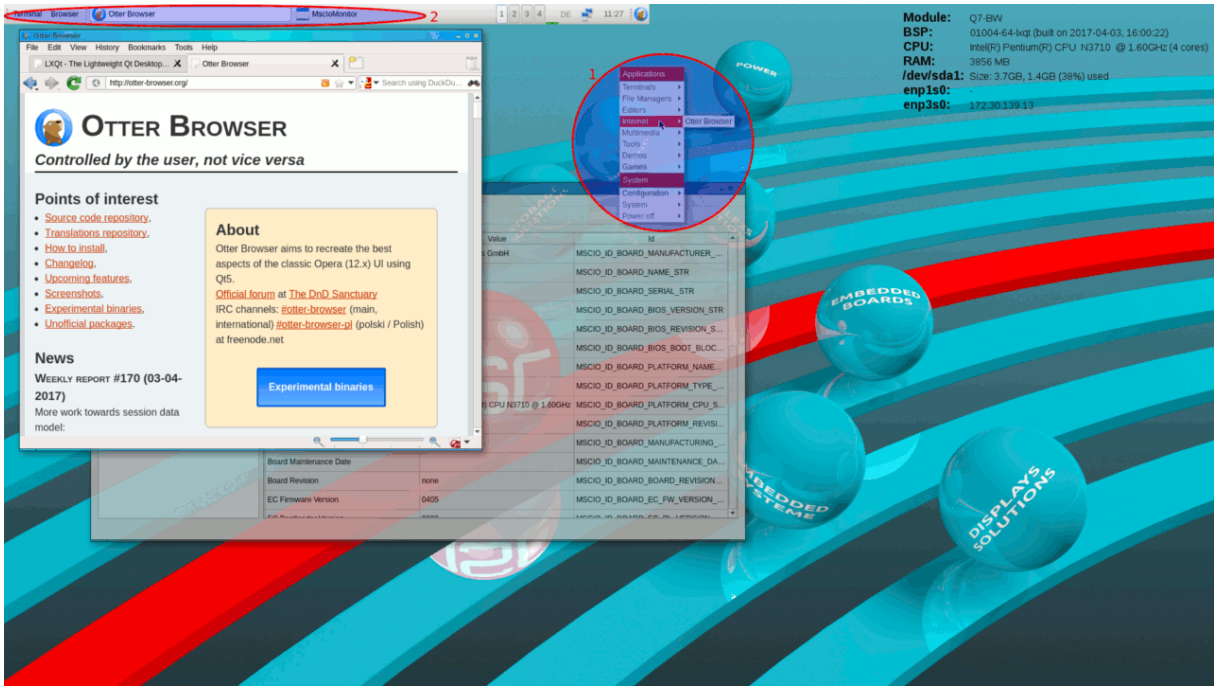
It is also possible to change the time zone of a running system to e.g. New York, using the following shell commands:

```
# rm -f /etc/localtime
# ln -s /usr/share/zoneinfo/America/New_York /etc/localtime
# echo "America/New_York" > /etc/timezone
```

4.16. Using LXQt

LXQt is a lightweight desktop for Qt based systems (http://lxqt.org).

Figure 4.2. – LXQt - Mainwindow



The pre-configured applications are accessible using the right mouse button.

- Application Launcher (1) - If the right mouse button is pressed on the desktop, the pre-configured application list is opened for launching terminal, browser, editor etc.
- Window List (2) - The top panel lists the windows applications.

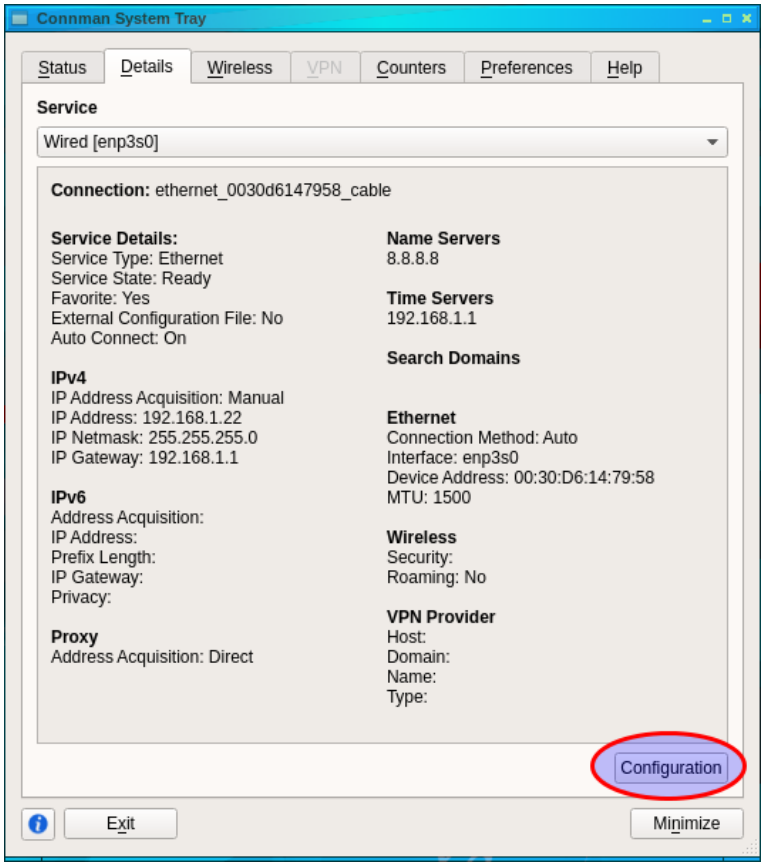
4.17. ConnMan Configuration

The internet connection daemon ConnMan (https://01.org/connman) is used in LXQt images to manage the internet connections.
The ConnMan Property Editor can be launched using a right-click on the LXQt desktop:

Figure 4.3. – Start ConnMan Property Editor

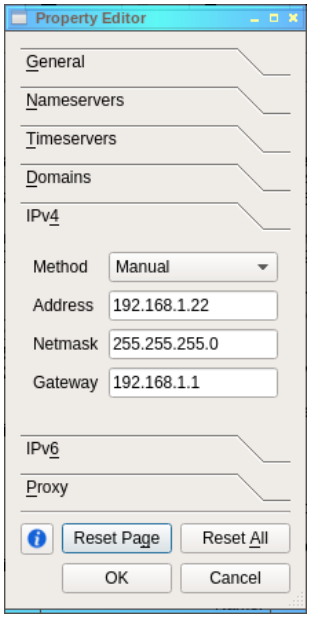
There is a detailed overview for each connection when the Details tab is selected:

Figure 4.4. – View ConnMan Properties



Pressing the Configuration button in the Details tab opens the configuration dialog that allows to do a manual IP assignment (it is important to enter Address, Netmask and Gateway – otherwise the setting is not applied):

Figure 4.5. – Manual IP Assignment



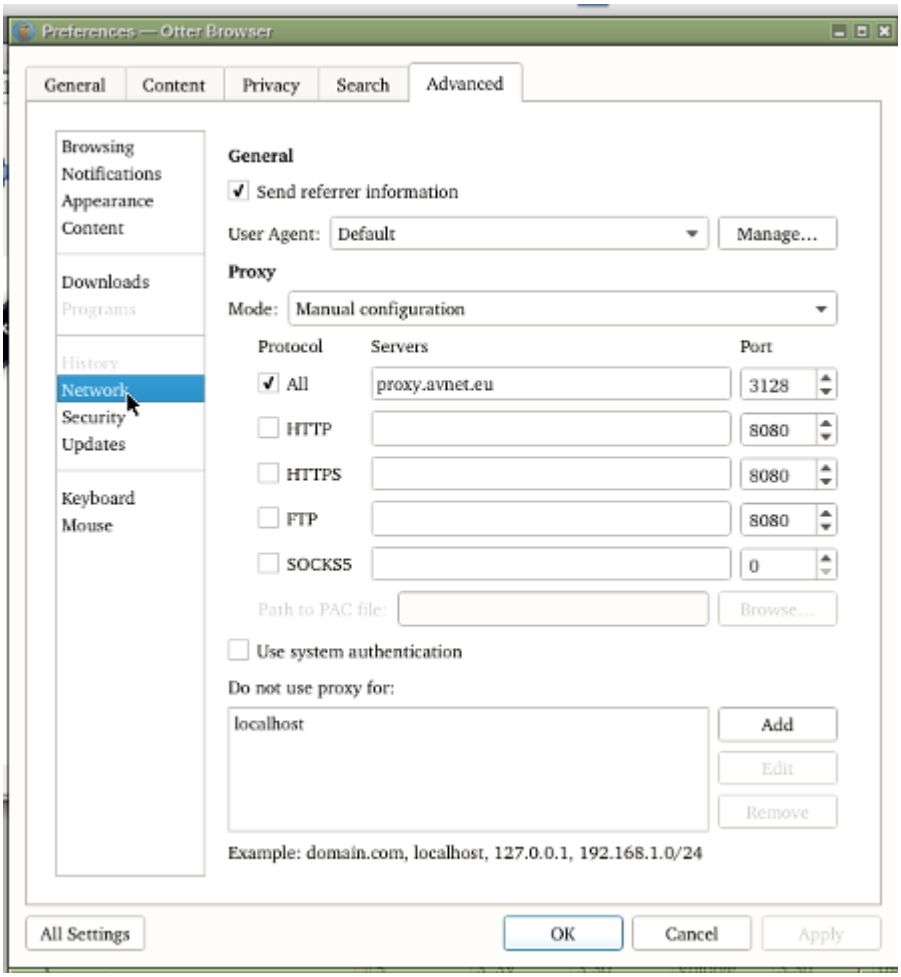
The tool `connman-info` can be used to show retrieve network interface details:

```
# connman-info
[/net/connman/service/ethernet_0044d6143d58_cable]
    Name = Wired
    IPv6 = {}
    IPv4.Configuration = {Method=dhcp}
    Nameservers = [192.168.1.49, 192.168.1.57]
    IPv6.Configuration = {Method=off}
    IPv4 = {Netmask=255.255.255.0, Method=dhcp, Gateway=192.168.1.201, Address=192.168.1.13}
    Domains.Configuration = []
    Timeservers = [192.168.1.201]
    Domains = []
    State = ready
    Ethernet = {MTU=1500, Interface=enp3s0, Method=auto, Address=00:44:D6:14:3d:58}
    Security = []
    Immutable = false
    Favorite = true
    Timeservers.Configuration = []
    Proxy.Configuration = {Method=direct}
    Type = ethernet
    AutoConnect = true
    Provider = {}
    Proxy = {Method=direct}
    Nameservers.Configuration = []
```

4.17.1. Proxy Configuration For The otter-browser

The Otter Web Browser has no proxy pre-configured. On some systems it is therefore necessary to manually setup the proxy configuration (under Tools/Preferences/Network). It is also possible to do a system wide proxy setup (see section 4.15.4).

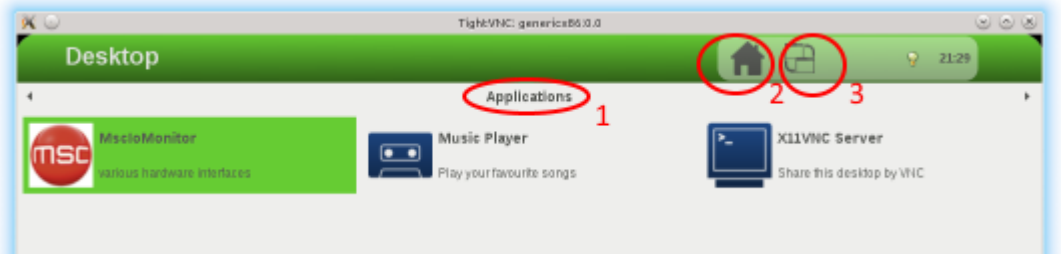
Figure 4.6. – Otter Browser - Proxy Setup



4.18. Using Sato

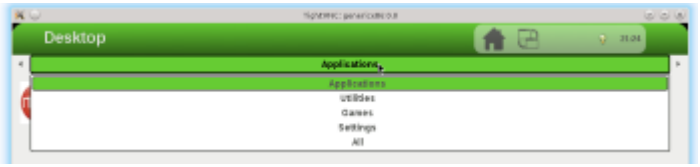
Sato is a window manager in which applications run full-screen, optimized for use with touch-displays.

Figure 4.7. – Sato - Mainwindow



- Dropdown Box (1) - This lists the various application groups.

  Figure 4.8. – Sato - Groups

- Button (2) - This displays the application group.
- Dropdown Box (3) - This lists the running GUI applications.

  Figure 4.9. – Sato - Selecting Running Apps

4.19. Using The SDK Images

The `msc-image-*-sdk` images contain everything for building applications on the target, e.g. gcc, system header files, make, cmake, git, gdb etc. This simplifies development because the application can be developed before any yocto recipes for it are created.

4.19.1. Setting Up The Development Host

The development host is used for editing the source files or running analyzes, e.g. code coverage. The source files could be shared using NFS.

Create an entry in `/etc/exports` to the base of your sources. Use your own UID/GID for `anonuid` and `anongid`.

```
/home/developer/src  *(rw,all_squash,no_subtree_check,anonuid=2000,anongid=2000)
```

File 4.1 – /etc/exports

Restart the NFS server, e.g. with:

```
sudo service nfs-kernel-server restart
```

4.19.2. Setting Up The Target

The sources are ideally mounted with the same hierarchy as on the development host. Replace `192.168.150.2` with the IP address of the development host.

Add an entry in /etc/fstab so the user MSC can mount the directory:

```
sudo /bin/bash
echo "192.168.150.2:/home/develop/src /home/develop/src auto user,exec 0 0" >>/etc/fstab
mkdir -p /home/develop/src
exit


mount /home/develop/src
```

4.19.3. Building hello_world Manually

Create the file `hello_world.c` in `/home/develop/src`.

```
#include <stdio.h>

int main(int argc, char *argv[])
    {
    printf("Hello world\n");
    return 0;
    }
```

File 4.2 – hello_world.c

Build and run it with:

```
gcc -o hello_world hello_world.c && ./hello_world
```

To debug it, rebuild it with debugging information and run it with:

```
gcc -g -o hello_world hello_world.c && gdb ./hello_world
```

4.19.4. Building hello_world With CMake

Create the file `CMakeLists.txt` in `/home/develop/src`.

```
project(hello_world)

add_executable(
  hello_world
  hello_world.c
)
```

File 4.3 – CMakeLists.txt

Build and run it with:

```
mkdir Release
cd Release
# This will take a while as cmake is slow on NFS
cmake ..
make && ./hello_world
```

4.19.5. Building Qt Based hello_world (Only In msc-image-lxqt-sdk)

Create the file `hello_world.cpp` in `/home/develop/src`.

```
#include <QApplication>
#include <QLabel>

int main(int argc, char *argv[])
{
    QApplication a(argc, argv);
    QLabel label("Hello world");

    label.show();
    return a.exec();
}
```

File 4.4 – hello_world.cpp

Create the file `hello_world.pro` in `/home/develop/src`.

```
QT      += core gui widgets
TARGET   = hello_world
TEMPLATE = app
SOURCES += hello_world.cpp
```

File 4.5 – hello_world.pro

Build and run it with:

```
qmake hello_world.pro && make && ./hello_world
```

4.20. German Keyboard Layouts

To support different keyboard layouts even when RAMDISK images are used (cpio.gz), the keyboard can be selected using a kernel command line, e.g. set in `EFI/BOOT/grub.cfg`.

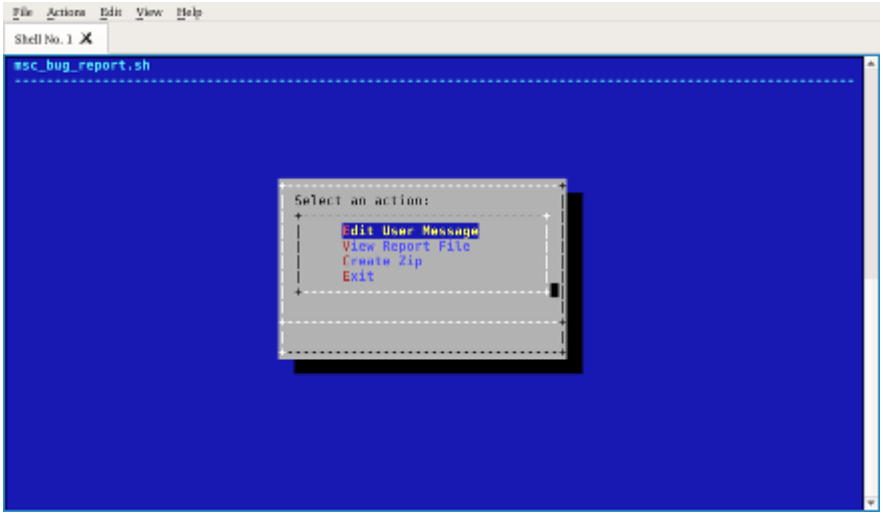
To enable german keyboard layout, add the kernel command line option: `keyboard=de-latin1-nodeadkeys`

4.21. Bug Reporting

To simplify collecting information necessary for effectively responding to bug reports, use the tool `msc_bug_report.sh` to generate bug report message. It will collect all necessary information like hardware, kernel logs etc.

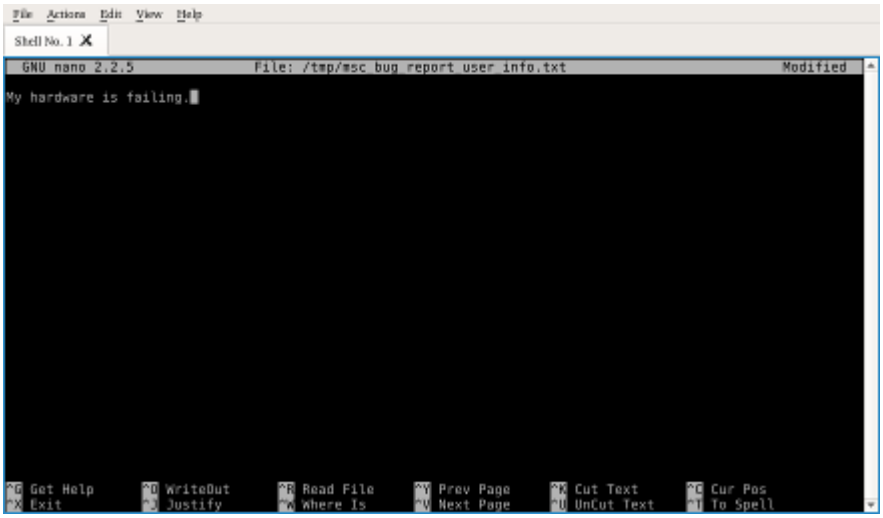
- Run `msc_bug_report.sh`.

Figure 4.10. – Bug Reporting - Start



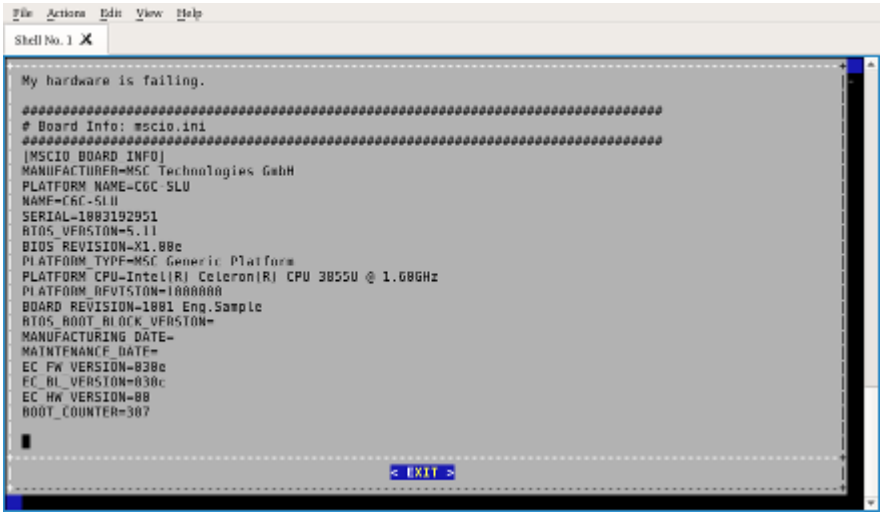
- Select “Edit User Message”.
- Enter bug report message and press Ctrl-O and Ctrl-X.

Figure 4.11. – Bug Reporting - Start



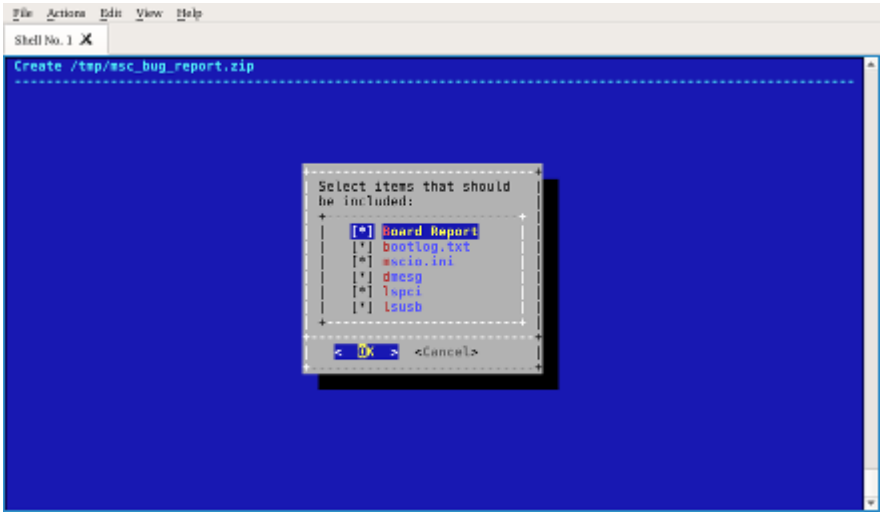
- Optionally you can then view the message with the board report (hardware information).

Figure 4.12. – Bug Reporting - View



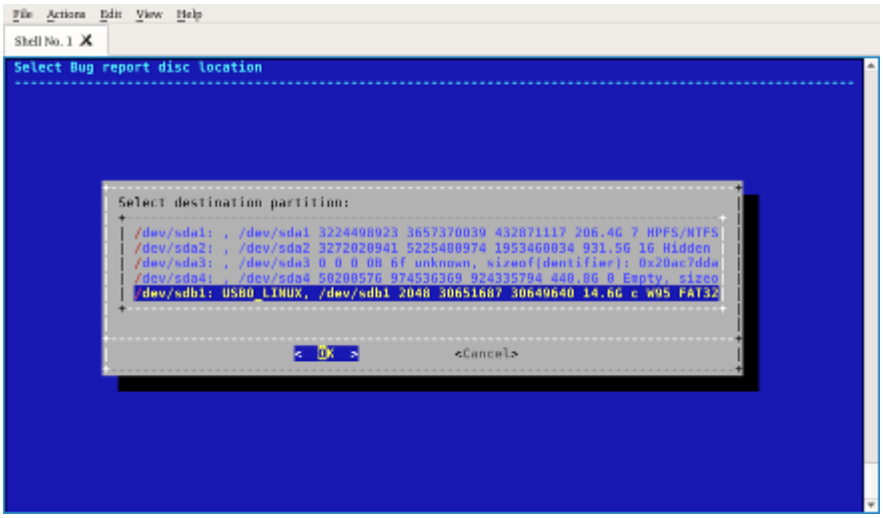
- Press “Create a zip file” and select the components you want to send (e.g. bootlog, mscio.ini, last kernel logs (dmesg) or the installed hardware).

Figure 4.13. – Bug Reporting - Create Zip



- Press “Save ZIP to a disc” and select the filesystem where to store the zip file. It is recommended to use a USB stick.

Figure 4.14. – Bug Reporting - Save Zip



- Send the files `msc_bug_report_brief.txt` and `msc_bug_report.zip` to MSC (http://www.msc-technologies.eu/de/support.html)

4.22. Hotfixes And Updating MSC-LDK

Typically twice a year a full MSC-LDK release is created. A release may contain an updated Yocto or other updated layers as well as new supported boards. For each release an own branch is used (e.g. `v1.0.0`) which is tagged with the date encoded (e.g. `LC984_20150421_V0_4_0`, 21st April 2015), too. The release is checked out using the version syntax (`git checkout v1.9.0`) as described above.

Sometimes an intermediate hot-fix is necessary which doesn't modify the resulting image but fixes changed repository locations of third party software or similar light changes. Hot-fixes are tagged with a newer date stamp (e.g. `LC984_20160113_V0_4_0`). A hot-fix can be checked out explicitly using these tags.

When MSC-LDK is checked out freshly all hot-fixes are applied automatically. To update an older checkout and to pull all the newer hot-fixes, run `scripts/update.py` from the MSC-LDK root directory. This will update MSC-LDK and all layers. Depending on the kind of hot-fix running `setup.py` again might be necessary. When a hot-fix has been checked out explicitly, running update will not make sense and it will fail with an error.

After the first call of `setup.py` no manual "git checkout" must be performed as its layers need to be in synchronization with MSC-LDK. Either use `update.py` or clone MSC-LDK again. The subdirectories `download` and `sstate-cached` can be moved to other MSC-LDK installations or shared by symbolic links.

Early adaptors of new BSPs might want to use the master branch instead of a released branch. Here `update.py` must also be used.

5. Package List

The MSC-LDK contains various MSC specific packages which are described here.

5.1. eapi-so

This package provides the libEApi.so library implementing the EAPI standard for accessing the hardware.

Dependencies: mscio.

5.2. mscio

This package provides mscio-setup and libMscIo for hardware initialization and enumeration of non-plug-and-play hardware devices.

In the images msc-image-base, msc-image-sato and msc-image-lxqt, the application mscio-setup is loaded automatically on startup, visualized by this message:

```
Loading MSC-IO: Done
```

This can be disabled by setting the kernel command line option `mscio_disable=y`.

5.3. system-test-controller

This package provides the system-test-controller application which controls system and hardware tests defined in .ini files in /etc/system-test-controller.cfg.

A. Glossary

BSP Board Support Package. 21, 25

EAPI Embedded Application Programmer Interface. 42

EFI Extensible Firmware Interface. 44

MSC-LDK MSC Linux Development Kit. 17, 21

B. Bootloader Configuration

B.1. RAMDISK

It is possible to install the Linux images on a DOS formatted USB stick or SD card and boot them via EFI.

Advantages are:

- No changes are persistent, so the system boots every time the same environment.
- The boot media can be removed once Linux is running.
- The boot media can be setup by a windows host without further tools.

Disadvantages are:

- More RAM is required as the Linux kernel and root file-system is kept there.
- Booting large images take longer as the BIOS has to load everything.

To install, copy the configuration `grub.cfg` and the bootloader `bootx64.efi` into the directory `EFI/BOOT` of the boot media. Then run:

```
user@devhost:msc-ldk$ make install_images DESTDIR=<mountpoint_bootmedia>/images
```

```
default=msc-image-base-elkhartlake-64
timeout=2

menuentry 'msc-image-base-elkhartlake-64'{
          linux /images/bzImage-intel-corei7-64.bin LABEL=base loglevel=3 vga=1 vmalloc=256M ↩
              ramdisk_size=196608 root=/dev/ram
          initrd /images/msc-image-base-intel-corei7-64.cpio.gz
}
```

File B.1 – EFI/BOOT/grub.cfg

C. X11/SATO Configuration

C.1. Screens

The screens can be configured on the fly with the `xrandr` tool.

At first determine the list of possible modes:

```
root@genericx86:~ # xrandr
Screen 0: minimum 320 x 200, current 1920 x 1080, maximum 32767 x 32767
LVDS1 connected 1024x768+0+0 (normal left inverted right x axis y axis) 0mm x 0mm
   1024x768       60.0*+
   800x600        60.3     56.2
   640x480        59.9
VGA1 connected 1920x1080+0+0 (normal left inverted right x axis y axis) 598mm x 336mm
   1920x1080      60.0*+
   1680x1050      60.0
   1280x1024      75.0     60.0
   1440x900       75.0     59.9
   1280x960       60.0
   1152x864       75.0
   1280x720       60.0
   1024x768       75.1     70.1     60.0
   832x624        74.6
   800x600        72.2     75.0     60.3     56.2
   640x480        75.0     72.8     66.7     60.0
   720x400        70.1
HDMI1 disconnected (normal left inverted right x axis y axis)
DP1 disconnected (normal left inverted right x axis y axis)
VIRTUAL1 disconnected (normal left inverted right x axis y axis)
```

The available screens depends on the CPU module being used.

Then reconfigure the screen with:

```
root@genericx86:~ # xrandr --output VGA1 --mode 800x600
```

D. Supported Periphery

Table D.1. – Supported periphery

Periphery	Status	Comment
ACPI	implemented	
Audio (HDA)	implemented	see Audio (HDA)
Audio (I2S)	implemented	see Audio (I2S)
BIOS Flash	implemented	see BIOS Flash
CAN	implemented	
Ethernet	implemented	
eMMC	implemented	
GPIOs	implemented	
Graphic output	implemented	
Hardware sensors	implemented	see Hardware Monitoring
I2C	implemented	see I2C
LPC	t.b.d.	
NVRAM	implemented	see NVRAM
PCIe	implemented	
Powermanagement	t.b.d.	
RTC	implemented	
SATA	implemented	
SD/MMC	implemented	see SD/MMC
SMBus	implemented	see SMBus
SPI	implemented	see SPI
TPM	implemented	
UART	implemented	see UART
USB-2.0	implemented	
USB-3.0	implemented	
Watchdog	implemented	see Watchdog

D.1. Audio (HDA)

LxQT supports sound via PulseAudio, all other images use ALSA directly. Pulse and ALSA are configured and used slightly differently.

D.1.1. PulseAudio

Sound controls can be controlled via the GUI with `pavucontrol-qt` or command-line with `pactl`.

Playback

To play sound, do:

```
aplay /usr/share/sounds/alsa/Front_Center.wav
```

To select the analog output device, do:

```
pactl set-card-profile 0 output:analog-stereo
```

To select the HDMI output device, do:

```
pactl set-card-profile 0 output:hdmi-stereo
```

Recording

To record from the microphone, do:

```
# Stop recording with Ctrl-C
pactl set-card-profile 0 output:analog-stereo+input:analog-stereo
arecord record.wav
```

Miscellaneous

- If the message “Connection refused” is printed when using audio tools on an SSH terminal, try setting the environment variable `DISPLAY` to `:0` as SSH’s ForwardX11 doesn’t work with PulseAudio.
- The persistent pulse audio configuration is stored in `~/.config/pulse`
- PulseAudio has been configured to support access for all users of the system, not only the desktop user.
- The list of profiles can be determined with:

  ```
  pactl list | grep 'Part of profile'
  ```

  Alternatively, use `pavucontrol-qt` and select the profile in the tab `Configuration`

D.1.2. ALSA

Sound controls can be modified with `alsamixer`.

Playback

To play sound on the headphone jack, do:

```
aplay /usr/share/sounds/alsa/Front_Center.wav
```

Recording

To record from the microphone, do:

```
# Set microphone sensibility
amixer sset 'Mic Boost' 100
# Stop recording with Ctrl-C
arecord record.wav
# Play back with
aplay record.wav
```

Playback With HDMI

- Determine which sound device is used for HDMI. This setting is dependent on the connected monitors and sound cards.

  ```
  speaker-test -t sine -D 'hdmi:CARD=PCH,DEV=0' -c 2
  ```

  The list of supported hdmi PCM devices can be retrieved with:

  ```
  aplay -L
  hdmi:CARD=PCH,DEV=0
      HDA Intel PCH, HDMI 0
      HDMI Audio Output
  hdmi:CARD=PCH,DEV=1
      HDA Intel PCH, HDMI 1
      HDMI Audio Output
  ```

- aplay requires the sound files to be in the correct format for HDMI (e.g. `S16_LE`). To convert them on the development host, run:

  ```
  sox <input.wav> -b 16 -c 2 <output.wav>
  ```

- To make HDMI the default sound device, create a file `/root/.asoundrc` with this content:

  ```
  pcm.!default {
    type hw
    card 0
    device 3
  }
  ```

  Note that the device number is changed. This is because it is the raw device which list can be retrieved with:

  ```
  aplay -l
  **** List of PLAYBACK Hardware Devices ****
  card 0: PCH [HDA Intel PCH], device 3: HDMI 0 [HDMI 0]
    Subdevices: 1/1
    Subdevice #0: subdevice #0
  ```

Persistent Mixer Settings

The mixer settings can be saved to `/var/lib/alsa/asound.state` with `alsactl store` and restored with `alsactl restore`.

D.2. Audio (I2S)

I2S audio is supported on the SM2-MB-EP1 baseboard. This requires both hardware changes and software changes.

D.2.1. Kernel Command Line

Add `modprobe.blacklist=snd_hda_intel` to your kernel command line, e.g. on your bootmedia in `EFI/BOOT/grub.cfg`.

SM2-MB-EP1 hardware changes

- Set S2801-3 (HDA/I2S Select) to Off. The switch is right beside the battery and RS232 connector
- Remove R1705 and assemble it on R1707

D.2.2. BIOS configuration changes

Enter BIOS setup with key "del" after power-on or reset.

- Select Chipset/PCH-IO Configuration/HD Audio Configuration
  - Set "Audio DSP" to "Enabled"
  - Set "Audio DSP Compliance Mode" to "Non-UAA (IntelSST)"
  - Set "HD Audio Link" to "SSP (I2S)"
- Save changes and exit

D.2.3. Playback

Playback to the headphone is working out of scratch in ALSA and PulseAudio.

D.2.4. Recording From Microphone

Recording from microphone is enabled by default in ALSA and PulseAudio.
To switch in ALSA to microphone input, do:

```
amixer sset 'Capture Mux' 'MIC_IN'
```

D.2.5. Recording From Line-In

Recording from line-in is only supported from ALSA and not yet PulseAudio. To switch in ALSA to line-in input, do:

```
amixer sset 'Capture Mux' 'LINE_IN'
```

D.3. BIOS Flash

The BIOS flash is available read-only as a MTD device named "BIOS". See `/proc/mtd` for the actual device number.

D.4. I2C

The I2C bus is named `Synopsys DesignWare I2C adapter` and provided by the driver `i2c_designware`. Run `i2cdetect -l` to retrieve the actual bus number or use the I2C Bus Identification feature.

D.5. SMBus

The SMBus is named `SMBus I801 adapter` and provided by the driver `i2c_i801`. Run `i2cdetect -l` to retrieve the actual bus number or use the I2C Bus Identification feature.

D.6. SPI

SPI on the eSPI lines is supported when the CPU module is configured for SIO_SPI1 instead of eSPI. Then enter BIOS setup and enable "Chipset/PCH-IO Configuration/SerialIo Configuration/SPI1 Controller". The SPI master will be made available under `/sys/devices/pci0000:00/0000:00:1e.3/pxa2xx-spi.4/spi_master`.

D.7. SD/MMC

SD cards are accessible as usual under `/dev/mmcblk<N>`.

In case the signal-path length of the whole system (CPU, CPU-Module, Baseboard) is at or above the specification, reducing the signal frequency might be a workaround. This can be achieved with the kernel command line option `sdhci_pci.max_freq` and the maximum frequency in MHz. Possible values are 25, 50, 208.

For example: `sdhci_pci.max_freq=25`

enforces the driver to use an UHS-I card in SDR12 (25 MHz) mode instead of its maximum speed.

If the parameter is accepted, the driver will print a kernel message starting with: `SD quirks enforced:`

The values can be verified in the debugfs:

```
cat /sys/kernel/debug/mmc0/ios
clock:          25000000 Hz
actual clock:   25000000 Hz
vdd:            7 (1.65 - 1.95 V)
bus mode:       2 (push-pull)
chip select:    0 (don't care)
power mode:     2 (on)
bus width:      3 (8 bits)
timing spec:    0 (legacy)
signal voltage: 0 (3.30 V)
driver type:    0 (driver type B)
```

They are only valid if a file-system is mounted!

As SDR25 and DDR50 use both 50MHz, DDR50 can be explicitly disabled with:

```
sdhci_pci.disable_ddr50=1
```

D.8. UART

The UART ports are available via `/dev/ttyS<n>` device nodes. They are assigned on the SM2-EL module as:

Table D.2. – UART Ports

/dev/ttyLP<n>	SM2 Connector	MSC-SM2-MB-EP1	MSC-SM2S-MB-EP5
ttyS4	SER0	X2301	X1102 (UART0)
ttyS6	SER1	X2302	Feature Connector 3 (UART1)

`/dev/ttyS5` is not connected anywhere.

The UART port SER0 supports RTS/CTS signals.

E. MSC-IO

E.1. What Is MSC-IO?

MSC-IO provides access to CPU module information and non-plug-and-play hardware devices.

E.2. mscio-setup

`mscio-setup` is called on startup unless the kernel command line option `mscio_disable` is present. It parses the SMBIOS table similar to the tool `dmidecode` and stores the information in an easy to use file based format under `/var/run/mscio/` for non-root applications to access them.

`mscio-setup` also setups hardware devices which are not plug-and-play capable, e.g. the embedded controller on the SMBus. BIOS images for MSC products released after January 2018 add the capability that these devices will be loaded automatically without the need of mscio-setup.

E.3. mscio-drivers

For fully supporting the board, some drivers are necessary:

Table E.1. – mscio-drivers

Driver	Feature
`eapi_ec`	Multi-function master device driver for the embedded controller.
`eapi_ec_bl`	Driver for the backlight controller on the embedded controller.
`eapi_ec_hwm`	Driver for the hardware monitoring sensors (temperature, voltage, fan) of the embedded controller.
`eapi_ec_running_time`	Driver to retrieve the total up-time of the board.
`eapi_ec_wdt`	Driver for the system watchdog of the embedded controller.
`devreg`	Driver to register the eapi devices. Used on older BIOS images which don't provide the system information by ACPI.
`i2c_ids`	Maps the name of the I2C busses to their numbers. This simplifies accessing them as explained in I2C Bus Identification.
`user-gpios`	Maps the name of the user GPIOs on CPU module to their numbers. This simplifies accessing them as explained in User GPIOs.

E.3.1. Hardware monitoring sensors

To determine the hwmon device via which the sensors can be inspected and controlled:

```
ls -l /sys/class/hwmon/hwmon*
```

The correct device for the embedded controller's sensors is the one whose symbolic link contains `i2c-MEX0000:00/eapi_ec_hwm`. In the following explanation `hwmonX` will be used for that device.

The device contains several files for each sensor, each of these following the same naming pattern:

`<sensor type><number>_<function>`

`<sensor type>` is

- `fan` for fan sensors/controls,
- `temp` for temperature sensors and

- `in` for voltage sensors.

`<function>` is

- `eapi_id` for the sensor's EAPI ID,
- `label` for a human-readable sensor name (e.g. `CPU_TEMP`),
- `input` for the measured sensor value,
- `auto` (fan only) for turning on automatic control mode and
- `duty` (fan only) for controlling the fan speed manually.

To read a sensor value:

```
# fan rotations per minute
cat /sys/class/hwmon/hwmonX/fan1_input
# temperatue in degrees millicelsius
cat /sys/class/hwmon/hwmonX/temp1_input
# voltage in millivolt
cat /sys/class/hwmon/hwmonX/in0_input
```

To switch a fan to manual control, write values between 0 and 255 into `fanY_duty`:

```
# Turn fan off manually
echo 0 > /sys/class/hwmon/hwmonX/fan1_duty
# Set fan to 100% duty cycle
echo 255 > /sys/class/hwmon/hwmonX/fan1_duty
```

The values describe a range between 0% and 100% duty cycle. The fan will stay at that duty cycle until the duty cycle is again changed manually or the fan is switched to automatic control mode.

To switch on automatic control:

```
echo 1 > /sys/class/hwmon/hwmonX/fan1_auto
```

The fan will remain under automatic control until it is switched to manual control mode like described above.

E.3.2. Backlight

The backlight of the LVDS or eDP display can be controlled via the backlight device `/sys/class/backlight/eapi_ec_bl`.

The kernel documentation contains a detailed description of the Backlight API.

To change the brightness:

```
# Display brightness
cat /sys/class/backlight/eapi_ec_bl/actual_brightness
255

# Dim it
echo 128 >/sys/class/backlight/eapi_ec_bl/brightness
```

or:

```
mscio-cmd get backlights
eapi_ec_bl,acpi_bl,

mscio-cmd geti backlight_brightness eapi_ec_bl
255

mscio-cmd seti backlight_brightness eapi_ec_bl 128
```

This backlight must be enabled first by setting the BIOS option `Chipset/Flat Panel Configuration/Backlight Control/PWM Control` to `EC`.

E.3.3. Hardware Monitoring

The CPU module features various sensors. They are accessible through the standard linux hwmon API interface with a few extensions under `/sys/bus/platform/devices/eapi_ec_hwm/hwmon/` and `/sys/class/hwmon/`. Sensor values can be read from the files named `*_input`, human readable names are available in the matching `*_label` files, e.g. the core voltage from `in0_label` and `in0_input`. A unique identification which has the location encoded is available in `*_eapi_id`.

The ordering of the sensors is not fixed and might change after a reset.

The kernel documentation contains a detailed description of the HWMON API.

An easy way to list all sensor values is by running `mscio-example` which among others lists all sensor devices.

Another approach is to use the tool `mscio-cmd`.

To list all sensors, run:

```
mscio-cmd get sensors
temp1@acpitz,temp2@acpitz,Physical id 0,Core 0,Core 1,Core 2,Core 3,CPU_FAN,SYSTEM_FAN,
    CORE_VOLT,3_3V,BATTERY_VOLT,5V,5V_STANDBY,12V,CPU_TEMP,SYSTEM_TEMP,BOARD_TEMP,MEMORY_TEMP
    ,
```

To read one sensor, run:

```
mscio-cmd get sensor_value 5V
5010mV
```

E.3.4. Running Time

To retrieve the total time in minutes the CPU module has been powered on in its life-time, do:

```
cat /sys/bus/platform/devices/eapi_ec_running_time/minutes
1975
```

or:

```
mscio-cmd get total_running_time
1975min
```

E.3.5. Watchdog

The embedded controller features a watchdog accessible via `/sys/bus/platform/devices/eapi_ec_wdt/watchdog/watchdog*/`, e.g. `/sys/bus/platform/devices/eapi_ec_wdt/watchdog/watchdog1/`.

As the CPU or baseboard components might provide a watchdog as well the watchdog number might change on reboots.

To start the watchdog:

```
  # Trigger every second, reboot after no trigger within 10s
  watchdog -t 1 -T 10 -F /dev/watchdog1
  # press ctrl-z and wait 10s for a reboot.
```

or:

```
mscio-watchdog start --reset-timeout=10s --trigger-time=1s --watchdog=EAPI_EC_Watchdog
```

The device node can be easily retrieved with:

```
mscio-cmd get watchdogs
iTCO_wdt,EAPI_EC_Watchdog,

mscio-cmd geti watchdog_device EAPI_EC_Watchdog
/dev/watchdog1
```

The driver can be configured to be not stoppable via the kernel command line parameter `eapi_ec_wdt.nowayout=1`. It can also be configured to not require a magic character when closing the device `eapi_ec_wdt.magic_close=0`.

A reconfiguration of the watchdog will stop and then restart it.

E.3.6. User GPIOs

If the BIOS or bootloader provides a user GPIO mapping it can be located at `/sys/bus/acpi/devices/MEX0001:00/physical_node/gpios`. For each GPIO a file with its linux number exists, e.g. `GPI0` or `GPO3`. GPIOs named `GPO` should be used only for output, GPIOs named `GPI` only for inputs.

To read a GPIO with the linux API, do:

```
GPIO=$(cat /sys/bus/acpi/devices/MEX0001:00/physical_node/gpios/GPI0)
cd /sys/class/gpio
echo ${GPIO} >export
cd gpio${GPIO}
echo in >direction
cat value
```

Inputs can be used as interrupts when the file `edge` exists.

The kernel documentation contains a detailed description of the GPIO API.

An easy way to list all GPIOs is by running `mscio-example`:

```
GPIO
 Name: GPI0
 Number: 456
 CanInput: 1
 CanOutput: 0
 CanInterrupt: 1
 IsInput: 1
 Value: 1
```

Another approach is to use the tool `mscio-cmd`.

To list all GPIOs, run:

```
mscio-cmd get gpios
GPI0,GPI1,GPI2,GPI3,GPO0,GPO1,GPO2,GPO3,
```

To read the states of all input pins, do:

```
mscio-cmd get-gpio-values
GPI00=1
GPI01=0
GPIO02=0
GPIO03=1
```

If the output pins are connected to LEDs, a visual inspection of all output pins is possible using the following:

```
mscio-cmd gpio-blink
```

This turns off all LEDs, then separately lights every one in the order of their names. At the end, all LEDs light up together and are then turned off again.

To read the current state of a single pin, do:

```
mscio-cmd seti gpio_config GPI0 in
mscio-cmd geti gpio_value GPI0
1
```

After the GPIO has been configured as an input, it is possible to wait for interrupts.

```
mscio-cmd wait-for-gpio-interrupt --mode=rising --timeout=2s GPI0
GPIO level after interrupt: 1
```

Verify that an interrupt has been raised with:

```
cat /proc/interrupts|grep gpiolib
 217:          0          0          2          0  intel-gpio   96  gpiolib
```

To set the current state to high, do:

```
mscio-cmd seti gpio_config GPO0 high
# or
mscio-cmd seti gpio_config GPO0 1
```

Support for GPIOs must first be enabled in the BIOS.

E.3.7. NVRAM

The CPU modules typically have an EEPROM attached which has some space reserved for the user. As some of the space might be used by the system, it is recommended to access the EEPROM using `mscio-cmd` which ensures that only the user area is used.

At first determine the available space.

```
mscio-cmd geti nvram_size EAPI-0
64B
```

To read 64 bytes as a hex string:

```
mscio-cmd read EAPI-0 0B 64B
11cc22ddffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
```

Writing to the NVRAM is also possible using `mscio-cmd`. The following command sets byte 1 to 0xaa and byte 2 to 0x98:

```
mscio-cmd write EAPI-0 1B aa98
mscio-cmd read EAPI-0 0B 10B
11aa98ddffffffffffff
```

E.3.8. Board Information

The location of board information like serial numbers, hardware revision or the platform type depends on the actual hardware. It is therefore recommended to use the following tools.

To print everything known, do:

```
mscio-example
```

To just retrieve the serial number, do:

```
mscio-cmd get board_serial
1004301002
```

E.3.9. I2C Bus Identification

A common problem on some platforms, especially x86, is the random assignment of the I2C bus numbers, for example `/dev/i2c-0`. When multiple I2C controllers are available, the numbers are assigned in a first initialized order. With parallel initialization the order might change on kernel changes or even after reboots.

The current assignment can be analyzed with this command:

```
root@intel-corei7-64:~# i2cdetect -l
i2c-3   i2c             DPDDC-B                                 I2C adapter
i2c-1   i2c             i915 gmbus dpb                          I2C adapter
i2c-6   smbus           SMBus I801 adapter at f040              SMBus adapter
i2c-4   i2c             DPDDC-C                                 I2C adapter
i2c-2   i2c             i915 gmbus dpd                          I2C adapter
i2c-0   i2c             i915 gmbus dpc                          I2C adapter
i2c-5   i2c             Synopsys DesignWare I2C adapter         I2C adapter
```

Sometimes the bus number can be determined by grepping for the name, e.g. `SMBus I801 adapter at f040`. But this fails when multiple instances of this name are present.

Another way is to use the physical path as a base and looking for a directory with the name `i2c-*` below the PCI device of the controller, e.g. `/sys/bus/pci/devices/0000:00:15.0`.

Both approaches are error prone. Therefore a third option exists. On modern MSC boards the BIOS provides an ACPI device `MEX0002` or `MEX00002`. The driver `i2c_ids` creates mapping files that contain the I2C bus number which can be found in the directory. `/sys/bus/acpi/devices/MEX0002:00/physical_node/i2c_ids`.

```
root@intel-corei7-64:/sys/bus/acpi/devices/MEX0002:00/physical_node/i2c_ids# ls
smbus  user
root@intel-corei7-64:/sys/bus/acpi/devices/MEX0002:00/physical_node/i2c_ids# cat smbus
6
root@intel-corei7-64:/sys/bus/acpi/devices/MEX0002:00/physical_node/i2c_ids# cat user
5
```

The numbers can be used to access the related `/dev/i2c-*` device

MSC products use the name `smbus` for the dedicated SMBus of the board. Dito `user` is used for the I2C bus reserved for user applications. Some boards might even feature more mappings.

Furthermore the I2C tools of MSC-LDK have been extended to use these features.

`i2cdetect -l` now reports these mappings as `*link*`

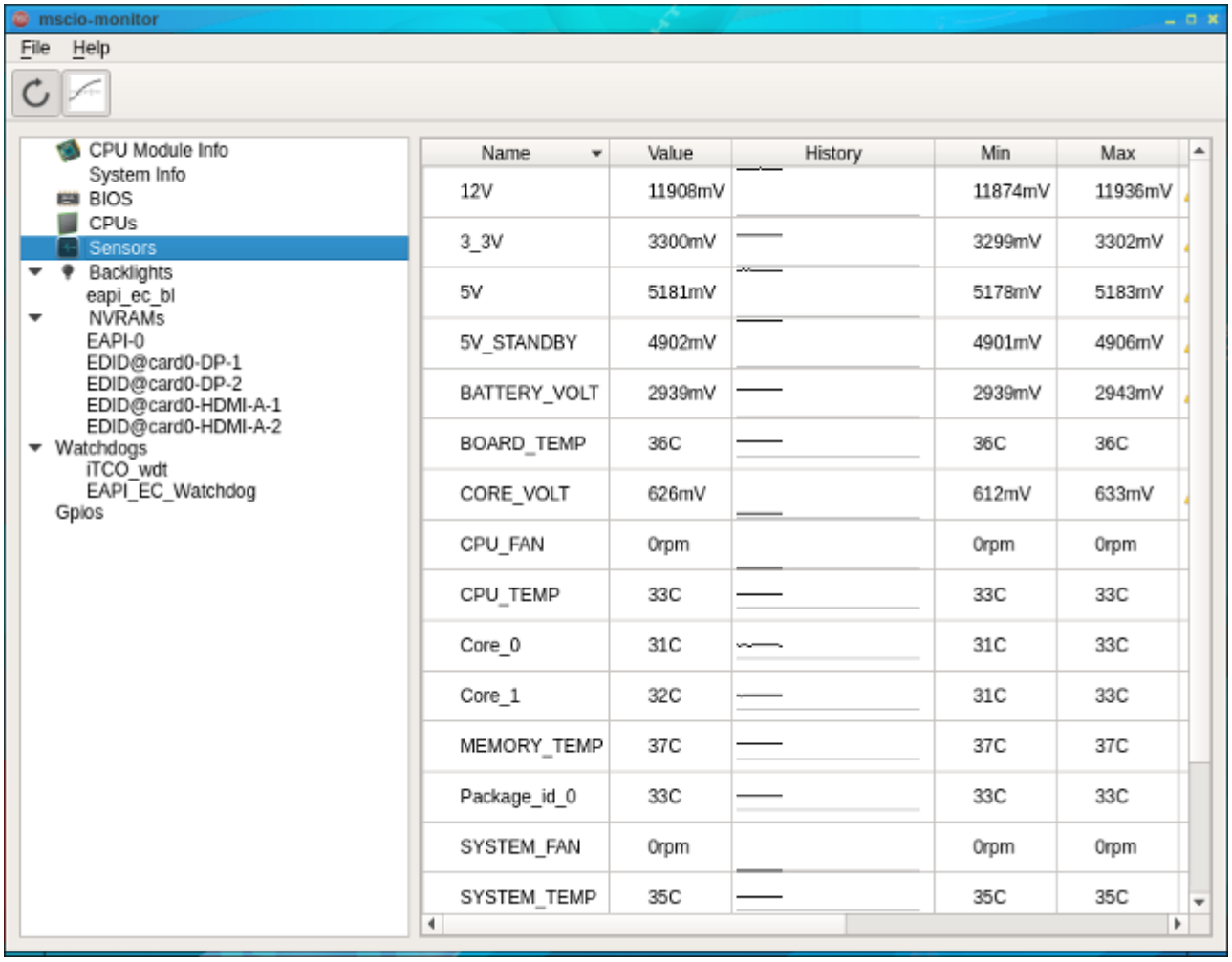
```
root@intel-corei7-64:~# i2cdetect -l
i2c-0   i2c             i915 gmbus dpc                          I2C adapter
i2c-5   i2c             Synopsys DesignWare I2C adapter         I2C adapter
i2c-5   *link*          user                                    ID mapped
i2c-6   *link*          smbus                                   ID mapped
```

If can be used as well as the bus parameter in some commands, e.g. `i2cdetect user` instead of `i2cdetect 0` or `i2cdump user 0x56` instead of `i2cdump 0 0x56`.

E.4. mscio-monitor

This is a Qt based front-end to MSC-IO included in the `msc-image-lxqt` and `msc-image-sato`.

Figure E.1. – mscio-monitor



E.5. User Library

Userspace application can access these devices with the raw kernel API or by using the package `mscio-lib`.

```
#include <iostream>
#include <msc/MscIo-3/MscIo.h>

using namespace Msc::MscIo;
using namespace std;

int main()
    {
    MscIo mscio;
    mscio.Init();

    for (const auto& s : mscio.Sensors())
        {
        cout << "Sensor" << endl
             << "  Name: " << s->Name() << endl
             << "  Value: " << s->ToString() << endl;
        }

    mscio.DeInit();

    return 0;
    }
```

File E.1 – example.cpp

It can be compiled with:

```
c++ -std=c++1z -o example example.cpp -l MscIo -lMscBoost
```

Output is:

```
Sensor
  Name: CPU_TEMP
  Value: 32C
Sensor
  Name: SYSTEM_TEMP
  Value: 35C
```

F. Troubleshooting

Table F.1. – Errors

Symptom	Solution
gitolite@msc-git02.msc-ge.com asks for password	Register as described here: Registration On The MSC Git Server
telnet login takes a few seconds after entering the password	A reverse DNS is performed to identify the login user. Either setup a local DNS server or change the `hosts` line in `/etc/nsswitch.conf` on the target to this: `hosts:  files`

G. Tips And Tricks

G.1. Using A Standard Browser

The image `msc-image-lxqt` includes the `otter-browser`. This is a small Qt5 based browser which is suitable for most web pages. If it is not sufficient other browsers like `google-chrome` or `firefox` can be installed instead via the layer `https://github.com/OSSystems/meta-browser`.

G.1.1. Using The google-chrome Browser

For example, to use `google-chrome` instead of `otter-browser`, do:

```
user@devhost:msc-ldk$ ./setup.py --bsp=01040 -layers-lxqt -layers-browser-chrome-instead-of-otter
```

Be aware that the compilation requires an additional 100GB and can take more than 2hours just for `chrome`.

G.2. Timekeeping

Having the exact time on a device is essential otherwise various internet protocols, e.g. HTTPS, are not working. Yet the RTC on the modules has a drift and should be therefore synchronized regularly. This chapter describes possible ways.

G.2.1. htpdate

If the device is behind a network proxy, the NTP protocol might be blocked by the firewall. There is a non-standard way to retrieve times using htpdate which is able to work with standard HTTP proxies.

To retrieve the current time from an HTTP server with debugging output (`-d`) and without a proxy use and set the system time:

```
htpdate -d www.google.com
```

If a HTTP proxy is configured, it must be provided on the command line:

```
htpdate -d -P ${http_proxy} www.google.com
```

Set the system time with (`-s`). For a better accuracy more and at best local web servers should be provided on the command line as well.

```
htpdate -s -P ${http_proxy} www.google.de www.kernel.org www.heise.de
```

To update the RTC, do:

```
hwclock --systohtc
```

H. Security

H.1. Introduction

Computer security is huge and important topic. Therefore Yocto offers some dedicated layers to

a) Analyze the created image and provide feedback about detected weak spots.

b) Increase the security of a running system.

H.2. Layer meta-security-isafw

The layer `meta-security-isafw` allows to enable the Image Security Analysis Framework (isafw) for your image builds. Further information about isafw can be found here: https://github.com/01org/isafw.

Adding the following line in `local.conf` enables a post processing step after the image creation: `INHERIT += "isafw"`

In that post processing step are several checks performed and the result is written to `build/01040*/tmp/log/isafw-report*/`:

CFA : Analyze executables on the image

RELRO : http://tk-blog.blogspot.de/2009/02/relro-not-so-well-known-memory.html

Canyry stack protection : https://lwn.net/Articles/584225/

Position Independent Executable : https://securityblog.redhat.com/2012/11/28/position-independent-executables-pie/

Memory Protection Extensions (MPX) : https://software.intel.com/sites/default/files/managed/9d/f6/Intel_MPX_EnablingGuide.pdf

CVE vulnerabilities : A list of not yet fixed CVE vulnerabilities.

FSA : SETUID, SETGID, World-writable files, World-writable dirs

KCA : Kernel settings

LA : Undesired recipe licenses

These reports can be used to improve your image step by step. Please note that security tools report a lot of possible problems and that almost no system fixes all of them. There is always a trade-off between security, usability and the amount of time that is invested to harden a system.

Using the layer `meta-security-isafw` is a good starting point to detect potential weak spots of the built Yocto images.

H.3. Setting Up A Project

`setup.py` will activate the layer`meta-security-isafw` when it is invoked with the additonal switch `--layers-security`. For example:

```
user@devhost:msc-ldk$ ./setup.py --bsp=01040 --layers-lxqt --layers-security
user@devhost:msc-ldk$ cd build/01040*-security
user@devhost:msc-ldk/build/01040*-security$ make msc-image-lxqt
```

I. Power Analysis

Various tools exist to analyse the power consumption and state changes of the CPU. They help to get an understanding when and why the kernel switches CPU core states and changes CPU core clock. With this information it is possible to adjust the system, e.g. with `taskset`, so more cores can run idle and the system use less power.

The most import CPU core states are:

Table I.1. – CPU core states

Mode	Description
C0	CPU fully turned on
C1	The main CPU clocks used for executing instructions are stopped, all others are still on
C3	All internal CPU clocks are stopped.
C6	All internal CPU clocks are stopped and internal voltage is reduced.

Depending on the CPU more modes might be available.

I.1. perf

This tool can be used to analyze task switches and monitor CPU state switches.

More information is available LWN perf sched and perf examples.

I.1.1. Analyzing CPU sleep states

To record the CPU core state transitions, record the execution of a command with:

```
root@intel-corei7-64:~# perf timechart record mscio-cmd geti sensor_value CORE_VOLT
942mV
[ perf record: Woken up 1 times to write data ]
[ perf record: Captured and wrote 0.053 MB perf.data (73 samples) ]
```

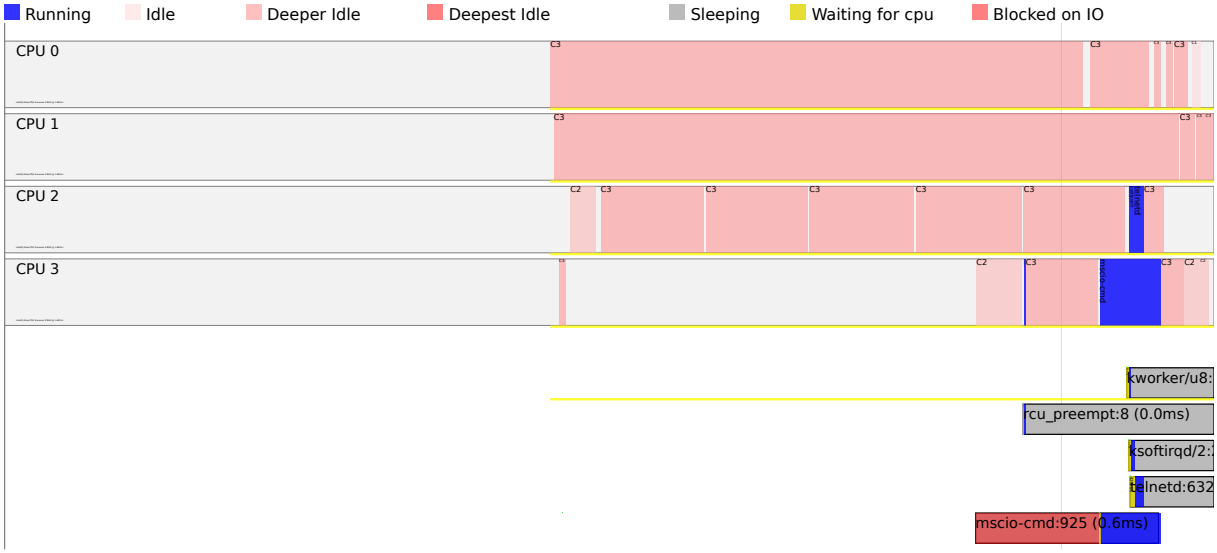
File I.1 – perf timechart record

Create the image file `output.svg` with:

```
root@intel-corei7-64:~# perf timechart
```

Then view it in `msc-image-lxqt` with `lximage-qt output.svg` or copy the file to the development host and run `inkscape output.svg`. Use of `inkscape` is recommended as it provides good zooming capability and doesn't cut wide text.

Figure I.1. – perf timechart output.svg



As can seen CPU 0 and 1 were mostly sleeping in C3 and one time in C1. CPU 2 was running a background telnet daemon and CPU3 was running the command `mscio-cmd`. This command was mostly waiting for some I/O (red) and finally some processing (blue).

I.1.2. Analyzing instruction and branches counter

With the `perf stat` command it is possible to retrieve the number of instructions and branches being executed for a command.

```
root@intel-corei7-64:~# perf stat sleep 1

Performance counter stats for 'sleep 1':

          0.727325      task-clock (msec)         #    0.001 CPUs utilized
                 1      context-switches          #    0.001 M/sec
                 0      cpu-migrations            #    0.000 K/sec
                53      page-faults               #    0.073 M/sec
           1132040      cycles                    #    1.556 GHz
            600009      instructions              #    0.53  insn per cycle
            123359      branches                  #  169.606 M/sec
              6806      branch-misses             #    5.52% of all branches

       1.001588350 seconds time elapsed
```

600.009 CPU instructions with 123359 conditional branches where needed to sleep for one second. Any children of the executed process are counted in.

I.2. cpupower

`cpupower` lists the percentages the system is in the various CPU core states.

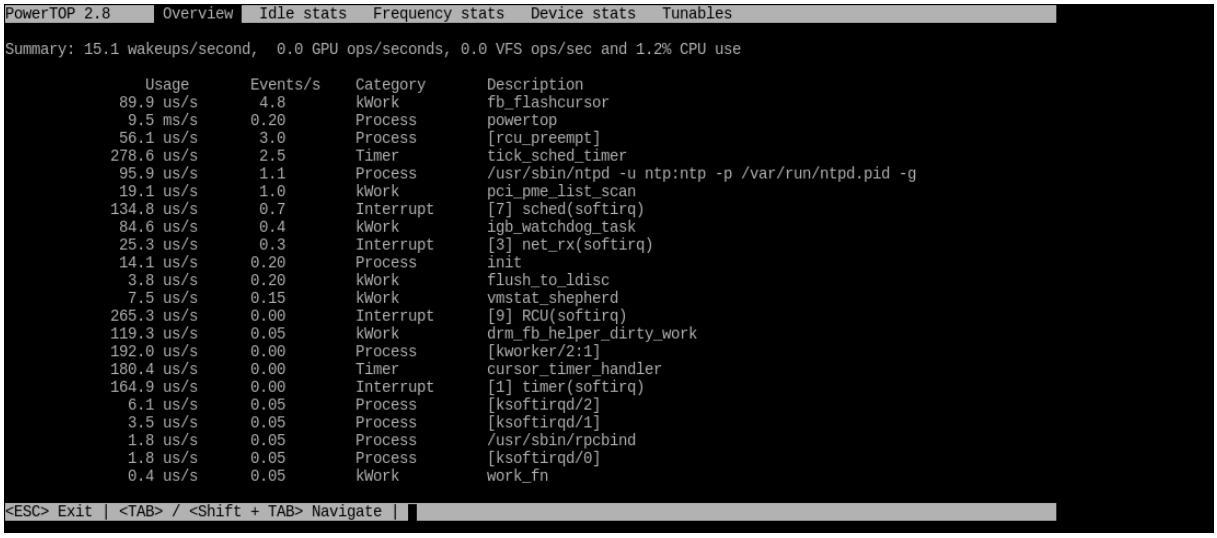
```
  root@intel-corei7-64:~# cpupower monitor
    |Nehalem                    || Mperf              || Idle_Stats
CPU | C3   | C6   | PC3  | PC6  || C0   | Cx   | Freq || POLL | C1   | C2   | C3
   0|  0.00| 99.62|  0.00|  0.00||  0.17| 99.83|  1792||  0.00|  0.17|  0.00| 99.68
   1|  0.00| 99.85|  0.00|  0.00||  0.14| 99.86|  1792||  0.00|  0.00|  0.00| 99.87
   2|  0.00| 99.98|  0.00|  0.00||  0.02| 99.98|  1823||  0.00|  0.00|  0.00| 99.99
   3|  0.00|  0.00|  0.00|  0.00|| 100.2| -0.16|  1792||  0.00|  0.00|  0.00|  0.00
```

The CPU is mostly in the C6 state while CPU 3 is busy serving one application.

I.3. powertop

`powertop` reports the activity on the system, e.g. events and processes actively using CPU, the CPU states being used, the CPU frequency distribution and the load of the physical devices. It also supports power tuning I/O devices. More documentation is available on its homepage.

Figure I.2. – powertop overview



J. Known Issues

J.1. USB drives

In rare cases individual USB flash drives are not registered properly.

K. Links

- MSC Technologies http://www.msc-technologies.eu
- Yocto project https://www.yoctoproject.org
- Yocto project documentation
  http://www.yoctoproject.org/docs/3.3/mega-manual/mega-manual.html
- LXQt http://lxqt.org
- Otter Browser https://otter-browser.org

L. License Overview

Table L.1. – list of licenses used in this build

Licenses		
(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial)	(GPL-2.0+ |LGPL-3.0)	(GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 |The-Qt-Company-Commercial)
(GPL-3.0 & The-Qt-Company-GPL-Exception-1.0)	(GPLv2+ |LGPLv3)	(MPL-2.0 & GPL-2.0+ & MIT)
(MPL-2.0 & LGPL-2.1+ & MIT)	(MPL-2.0 & MIT)	AFL-2.1
Apache-2.0	Artistic-1.0	BSD
BSD-1-Clause	BSD-2-Clause	BSD-3-Clause
BSD-4-Clause	BSL-1.0	CC-BY-SA-3.0
Firmware-Abilis	Firmware-GPLv2	Firmware-IntcSST2
Firmware-Lontium	Firmware-Marvell	Firmware-OLPC
Firmware-adsp_sst	Firmware-agere	Firmware-amd-ucode
Firmware-amdgpu	Firmware-amlogic_vdec	Firmware-ath9k-htc
Firmware-atheros_firmware	Firmware-atmel	Firmware-broadcom_bcm43xx
Firmware-ca0132	Firmware-cavium	Firmware-chelsio_firmware
Firmware-cw1200	Firmware-cypress	Firmware-dib0700
Firmware-e100	Firmware-ene_firmware	Firmware-fw_sst_0f28
Firmware-go7007	Firmware-hfi1_firmware	Firmware-i2400m
Firmware-i915	Firmware-ibt_firmware	Firmware-ice
Firmware-imx-sdma_firmware	Firmware-it913x	Firmware-iwlwifi_firmware
Firmware-kaweth	Firmware-moxa	Firmware-myri10ge_firmware
Firmware-netronome	Firmware-nvidia	Firmware-phanfw
Firmware-qat	Firmware-qcom	Firmware-qla1280
Firmware-qla2xxx	Firmware-qualcommAthos_ar3k	Firmware-qualcommAthos_ath10k
Firmware-r8a779x_usb3	Firmware-radeon	Firmware-ralink-firmware
Firmware-ralink_a_mediatek_company_firmware	Firmware-rtlwifi_firmware	Firmware-siano
Firmware-tda7706-firmware	Firmware-ti-connectivity	Firmware-ti-keystone
Firmware-ueagle-atm4-firmware	Firmware-via_vt6656	Firmware-wl1251
Firmware-xc4000	Firmware-xc5000	Firmware-xc5000c
FreeType	GFDL-1.2	GFDL-1.3
GPL-1.0+	GPL-2+	GPL-2.0
GPL-2.0+	GPL-2.0-only	GPL-2.0-or-later
GPL-2.0-with-OpenSSL-exception	GPL-3.0	GPL-3.0-only
GPL-3.0-with-GCC-exception	GPLv1	GPLv2
GPLv2+	GPLv2.0+	GPLv3
GPLv3+	ICU	ISC
Intel	Intel-Microcode-License	LGPL-2.0
LGPL-2.0+	LGPL-2.1	LGPL-2.1+
LGPL-2.1-or-later	LGPL-3.0	LGPLv2
LGPLv2+	LGPLv2.0+	LGPLv2.1
LGPLv2.1+	LGPLv3	LGPLv3+
Libpng	MIT	MIT-CMU
MIT-X	MIT-style	MPL-1.1
MPL-2.0	MPLv1.1	NTP
OFL-1.1	PD	PSF
PSFv2	Proprietary	Python-2.0
SGIv1	Sleepycat	The-Qt-Company-Commercial
WHENCE	Zlib	bzip2-1.0.4
bzip2-1.0.6	openssl	zsh

Table L.2. – License overview listing

Recipe	Version	Licenses
acl *acl*	2.2.53 *libacl*	GPLv2+
acpica *acpica*	20210105	Intel, BSD, GPLv2
acpid *acpid*	2.0.32	GPLv2+
adwaita-icon-theme *adwaita-icon-theme-symbolic*	3.34.3	LGPL-3.0, CC-BY-SA-3.0
alsa-lib *alsa-conf* *libatopology*	1.2.4 *alsa-lib*	LGPLv2.1, GPLv2+
alsa-plugins *alsa-plugins-pulseaudio-conf* *libasound-module-ctl-pulse*	1.2.2 *libasound-module-conf-pulse* *libasound-module-pcm-pulse*	LGPLv2.1, GPLv2+
alsa-state *alsa-state*	0.2.0 *alsa-states*	MIT
alsa-topology-conf *alsa-topology-conf*	1.2.4	BSD-3-Clause
alsa-ucm-conf *alsa-ucm-conf*	1.2.4	BSD-3-Clause
alsa-utils *alsa-utils* *alsa-utils-alsactl* *alsa-utils-alsamixer* *alsa-utils-alsaucm* *alsa-utils-aplay* *alsa-utils-aseqnet* *alsa-utils-midi*	1.2.4 *alsa-utils-aconnect* *alsa-utils-alsaloop* *alsa-utils-alsatplg* *alsa-utils-amixer* *alsa-utils-aseqdump* *alsa-utils-iecset* *alsa-utils-speakertest*	GPLv2
app-defaults *app-defaults*	1.0	MIT
at-spi2-atk *at-spi2-atk*	2.38.0	LGPL-2.1+
at-spi2-core *at-spi2-core*	2.38.0 *at-spi2-core-locale-en-gb*	LGPL-2.1+
atk *atk*	2.36.0 *atk-locale-en-gb*	GPLv2+, LGPLv2+
attr *libattr*	2.5.1	LGPLv2.1+
avahi *avahi-daemon* *libavahi-client* *libavahi-core*	0.8 *avahi-locale-en-gb* *libavahi-common*	LGPLv2.1+
base-files *base-files*	3.0.14	GPLv2
base-passwd *base-passwd*	3.5.29	GPLv2
bash *bash*	5.1	GPLv3+
bc *bc*	1.07.1	GPLv3+
binutils *libbfd*	2.36.1 *libopcodes*	GPLv3
bluez5 *bluez5* *bluez5-obex*	5.56 *bluez5-noinst-tools*	GPLv2+, LGPLv2.1+
bonnie++ *bonnie++*	1.03e *bonnie++-scripts*	GPLv2

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
boost *boost-filesystem* *boost-regex*	1.75.0 *boost-iostreams*	BSL-1.0, MIT, Python-2.0
bootlog *bootlog*	1.0	MIT
busybox *busybox* *busybox-syslog*	1.33.0 *busybox-hwclock* *busybox-udhcpc*	GPLv2, bzip2-1.0.4
bzip2 *libbz2*	1.0.8	bzip2-1.0.6
ca-certificates *ca-certificates*	20210119	GPL-2.0+, MPL-2.0
cairo *cairo*	1.16.0 *cairo-gobject*	MPL-1.1, LGPLv2.1
can-utils *can-utils*	2020.12.0	GPLv2, BSD-3-Clause
cifs-utils *cifs-utils*	6.4	GPLv3, LGPLv3
cmdline-keyboard *cmdline-keyboard*	1.0	MIT
cmdline-keyboard-xorg *cmdline-keyboard-xorg*	1.0	MIT
cmst *cmst*	2016.10.03	MIT
connman *connman*	1.39 *connman-client*	GPLv2
connman-info *connman-info*	1.0	GPLv2
consolekit *consolekit*	0.4.6	GPLv2+
continuous-test-system *continuous-test-system*	git	GPLv2
continuous-test-system-config *continuous-test-system-config*	1.0	MIT
coreutils *coreutils*	8.32 *coreutils-stdbuf*	GPLv3+
cpufrequtils *cpufrequtils*	008	GPLv2
cpupower *cpupower*	1.0	GPLv2
cryptsetup *cryptsetup*	2.3.5	GPL-2.0-with-OpenSSL-exception
cts-ui-web-compiled *cts-ui-web-compiled*	git	GPLv2
cups *cups-lib*	2.3.3 *cups-locale-en*	Apache-2.0
curl *libcurl*	7.75.0	MIT
db *db*	5.3.28	Sleepycat
dbus *dbus* *dbus-lib*	1.12.20 *dbus-common* *dbus-tools*	AFL-2.1, GPLv2+
dbus-glib *dbus-glib*	0.110	AFL-2.1, GPLv2+

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
desktop-file-utils *desktop-file-utils*	0.26	GPLv2+
dialog *dialog*	1.3-20210319	LGPL-2.1
dnf *dnf*	4.6.0 *dnf-locale-en-gb*	GPLv2
dosfstools *dosfstools*	4.2	GPLv3
e2fsprogs *e2fsprogs* *e2fsprogs-dumpe2fs* *e2fsprogs-mke2fs* *libcomerr* *libext2fs*	1.46.1 *e2fsprogs-badblocks* *e2fsprogs-e2fsck* *e2fsprogs-tune2fs* *libe2p* *libss*	GPLv2, LGPLv2, BSD, MIT
efi-var *efi-var*	git	GPLv2
eject *eject*	2.1.5	GPLv2
elfutils *elfutils* *libdw*	0.183 *libasm* *libelf*	GPLv3+
ell *ell*	0.38	LGPLv2.1
env-keep-gst *env-keep-gst*	1.0	MIT
ethtool *ethtool*	5.10	GPLv2+
eudev *eudev* *libudev*	3.2.10 *eudev-hwdb*	GPLv2.0+, LGPL-2.1+
evtest *evtest*	1.31	GPLv2
expat *expat*	2.2.10	MIT
ffmpeg *libavcodec* *libavformat* *libavutil* *libswscale*	4.3.2 *libavfilter* *libavresample* *libswresample*	LGPLv2.1+
file *file*	5.39	BSD-2-Clause
flac *libflac*	1.3.3	GFDL-1.2, GPLv2+, LGPLv2.1+, BSD
fontconfig *fontconfig*	2.13.1 *fontconfig-utils*	MIT-style, MIT, PD
formfactor *formfactor*	0.0	MIT
freetype *freetype*	2.10.4	FreeType, GPLv2+
fribidi *fribidi*	1.0.10	LGPLv2.1+
gcc-runtime *libatomic*	10.2.0 *libgomp*	GPL-3.0-with-GCC-exception

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
libstdc++		
gdbm	1.19	GPLv3
gdbm	*gdbm-compat*	
gdk-pixbuf	2.40.0	LGPLv2.1+
gdk-pixbuf	*gdk-pixbuf-loader-gif*	
gdk-pixbuf-loader-jpeg	*gdk-pixbuf-loader-png*	
gdk-pixbuf-loader-xpm	*gdk-pixbuf-locale-en-gb*	
git	2.31.1	GPLv2
git		
glib-2.0	2.66.7	LGPLv2.1+, BSD, PD
glib-2.0	*glib-2.0-locale-en-gb*	
glib-2.0-utils		
glib-networking	2.66.0	LGPLv2.1
glib-networking	*glib-networking-locale-en-gb*	
glibc	2.33	GPLv2, LGPLv2.1
glibc	*ldconfig*	
glibc-locale	2.33	GPLv2, LGPLv2.1
glibc-gconv	*glibc-gconv-iso8859-1*	
glibc-locale-en-gb	*locale-base-de-de*	
locale-base-en-gb	*locale-base-en-us*	
glmark2	2021.02	GPLv3+, SGIv1
glmark2		
gmmlib	21.1.1	MIT
gmmlib		
gmp	6.2.1	GPLv2+, LGPLv3+
gmp		
gnupg	2.2.27	GPLv3, LGPLv3
gnupg	*gnupg-gpg*	
gnutls	3.7.1	LGPLv2.1+
gnutls		
gobject-introspection	1.66.1	LGPLv2+, GPLv2+
gobject-introspection		
gottet	1.1.9	GPL-3.0
gottet	*gottet-locale-en*	
gpgme	1.15.1	GPLv2+, LGPLv2.1+
gpgme	*python3-gpg*	
grep	3.6	GPLv3
grep		
gsettings-desktop-schemas	3.38.0	LGPLv2.1
gsettings-desktop-schemas	*gsettings-desktop-schemas-locale-en-gb*	
gst-examples	1.18.4	LGPL-2.0+
gst-examples		
gstreamer1.0	1.18.4	LGPLv2+
gstreamer1.0	*gstreamer1.0-locale-en-gb*	
gstreamer1.0-libav	1.18.4	LGPLv2+
gstreamer1.0-libav		
gstreamer1.0-plugins-bad	1.18.4	GPLv2+, LGPLv2+, LGPLv2.1+
gstreamer1.0-plugins-bad-accurip	*gstreamer1.0-plugins-bad-adpcmdec*	
gstreamer1.0-plugins-bad-adpcmenc	*gstreamer1.0-plugins-bad-aiff*	
gstreamer1.0-plugins-bad-apps	*gstreamer1.0-plugins-bad-asfmux*	
gstreamer1.0-plugins-bad-audiobuffersplit	*gstreamer1.0-plugins-bad-audiofxbad*	
gstreamer1.0-plugins-bad-audiolatency	*gstreamer1.0-plugins-bad-audiomixmatrix*	
gstreamer1.0-plugins-bad-audiovisualizers	*gstreamer1.0-plugins-bad-autoconvert*	
gstreamer1.0-plugins-bad-bayer	*gstreamer1.0-plugins-bad-bluez*	
gstreamer1.0-plugins-bad-bz2	*gstreamer1.0-plugins-bad-camerabin*	
gstreamer1.0-plugins-bad-closedcaption	*gstreamer1.0-plugins-bad-coloreffects*	
gstreamer1.0-plugins-bad-curl	*gstreamer1.0-plugins-bad-dash*	

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
gstreamer1.0-plugins-bad-debugutilsbad	*gstreamer1.0-plugins-bad-decklink*	
gstreamer1.0-plugins-bad-dtls	*gstreamer1.0-plugins-bad-dvb*	
gstreamer1.0-plugins-bad-dvbsubenc	*gstreamer1.0-plugins-bad-dvbsuboverlay*	
gstreamer1.0-plugins-bad-dvdspu	*gstreamer1.0-plugins-bad-faceoverlay*	
gstreamer1.0-plugins-bad-fbdevsink	*gstreamer1.0-plugins-bad-festival*	
gstreamer1.0-plugins-bad-fieldanalysis	*gstreamer1.0-plugins-bad-freeverb*	
gstreamer1.0-plugins-bad-frei0r	*gstreamer1.0-plugins-bad-gaudieffects*	
gstreamer1.0-plugins-bad-gdp	*gstreamer1.0-plugins-bad-geometrictransform*	
gstreamer1.0-plugins-bad-hls	*gstreamer1.0-plugins-bad-id3tag*	
gstreamer1.0-plugins-bad-inter	*gstreamer1.0-plugins-bad-interlace*	
gstreamer1.0-plugins-bad-ipcpipeline	*gstreamer1.0-plugins-bad-ivfparse*	
gstreamer1.0-plugins-bad-ivtc	*gstreamer1.0-plugins-bad-jp2kdecimator*	
gstreamer1.0-plugins-bad-jpegformat	*gstreamer1.0-plugins-bad-legacyrawparse*	
gstreamer1.0-plugins-bad-locale-en-gb	*gstreamer1.0-plugins-bad-meta*	
gstreamer1.0-plugins-bad-midi	*gstreamer1.0-plugins-bad-mpegpsdemux*	
gstreamer1.0-plugins-bad-mpegpsmux	*gstreamer1.0-plugins-bad-mpegtsdemux*	
gstreamer1.0-plugins-bad-mpegtsmux	*gstreamer1.0-plugins-bad-mxf*	
gstreamer1.0-plugins-bad-netsim	*gstreamer1.0-plugins-bad-pcapparse*	
gstreamer1.0-plugins-bad-pnm	*gstreamer1.0-plugins-bad-proxy*	
gstreamer1.0-plugins-bad-removesilence	*gstreamer1.0-plugins-bad-rfbsrc*	
gstreamer1.0-plugins-bad-rist	*gstreamer1.0-plugins-bad-rsvg*	
gstreamer1.0-plugins-bad-rtmp2	*gstreamer1.0-plugins-bad-rtpmanagerbad*	
gstreamer1.0-plugins-bad-rtponvif	*gstreamer1.0-plugins-bad-sbc*	
gstreamer1.0-plugins-bad-sdpelem	*gstreamer1.0-plugins-bad-segmentclip*	
gstreamer1.0-plugins-bad-shm	*gstreamer1.0-plugins-bad-siren*	
gstreamer1.0-plugins-bad-smooth	*gstreamer1.0-plugins-bad-smoothstreaming*	
gstreamer1.0-plugins-bad-sndfile	*gstreamer1.0-plugins-bad-speed*	
gstreamer1.0-plugins-bad-subenc	*gstreamer1.0-plugins-bad-switchbin*	
gstreamer1.0-plugins-bad-timecode	*gstreamer1.0-plugins-bad-transcode*	
gstreamer1.0-plugins-bad-ttmlsubs	*gstreamer1.0-plugins-bad-uvch264*	
gstreamer1.0-plugins-bad-videofiltersbad	*gstreamer1.0-plugins-bad-videoframe-audiolevel*	
gstreamer1.0-plugins-bad-videoparsersbad	*gstreamer1.0-plugins-bad-videosignal*	
gstreamer1.0-plugins-bad-vmnc	*gstreamer1.0-plugins-bad-vulkan*	
gstreamer1.0-plugins-bad-waylandsink	*gstreamer1.0-plugins-bad-webp*	
gstreamer1.0-plugins-bad-y4mdec	*libgstadaptivedemux-1.0*	
libgstbadaudio-1.0	*libgstbasecamerabinsrc-1.0*	
libgstcodecparsers-1.0	*libgstcodecs-1.0*	
libgstinsertbin-1.0	*libgstisoff-1.0*	
libgstmpegts-1.0	*libgstphotography-1.0*	
libgstplayer-1.0	*libgstsctp-1.0*	
libgsttranscoder-1.0	*libgsturidownloader-1.0*	
libgstvulkan-1.0	*libgstwayland-1.0*	
libgstwebrtc-1.0		
gstreamer1.0-plugins-base	1.18.4	GPLv2+, LGPLv2+
gstreamer1.0-plugins-base-adder	*gstreamer1.0-plugins-base-alsa*	
gstreamer1.0-plugins-base-app	*gstreamer1.0-plugins-base-apps*	
gstreamer1.0-plugins-base-audioconvert	*gstreamer1.0-plugins-base-audiomixer*	
gstreamer1.0-plugins-base-audiorate	*gstreamer1.0-plugins-base-audioresample*	
gstreamer1.0-plugins-base-audiotestsrc	*gstreamer1.0-plugins-base-compositor*	
gstreamer1.0-plugins-base-encoding	*gstreamer1.0-plugins-base-gio*	
gstreamer1.0-plugins-base-locale-en-gb	*gstreamer1.0-plugins-base-meta*	
gstreamer1.0-plugins-base-ogg	*gstreamer1.0-plugins-base-opengl*	
gstreamer1.0-plugins-base-overlaycomposition	*gstreamer1.0-plugins-base-pango*	
gstreamer1.0-plugins-base-pbtypes	*gstreamer1.0-plugins-base-playback*	
gstreamer1.0-plugins-base-rawparse	*gstreamer1.0-plugins-base-subparse*	
gstreamer1.0-plugins-base-tcp	*gstreamer1.0-plugins-base-theora*	
gstreamer1.0-plugins-base-typefindfunctions	*gstreamer1.0-plugins-base-videoconvert*	
gstreamer1.0-plugins-base-videorate	*gstreamer1.0-plugins-base-videoscale*	
gstreamer1.0-plugins-base-videotestsrc	*gstreamer1.0-plugins-base-volume*	
gstreamer1.0-plugins-base-vorbis	*gstreamer1.0-plugins-base-ximagesink*	
gstreamer1.0-plugins-base-xvimagesink	*libgstallocators-1.0*	
libgstapp-1.0	*libgstaudio-1.0*	
libgstfft-1.0	*libgstgl-1.0*	
libgstpbutils-1.0	*libgstriff-1.0*	
libgstrtp-1.0	*libgstrtsp-1.0*	
libgstsdp-1.0	*libgsttag-1.0*	
libgstvideo-1.0		
gstreamer1.0-plugins-good	1.18.4	GPLv2+, LGPLv2.1+
gstreamer1.0-plugins-good	*gstreamer1.0-plugins-good-alaw*	

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
gstreamer1.0-plugins-good-alpha	*gstreamer1.0-plugins-good-alphacolor*	
gstreamer1.0-plugins-good-apetag	*gstreamer1.0-plugins-good-audiofx*	
gstreamer1.0-plugins-good-audioparsers	*gstreamer1.0-plugins-good-auparse*	
gstreamer1.0-plugins-good-autodetect	*gstreamer1.0-plugins-good-avi*	
gstreamer1.0-plugins-good-cairo	*gstreamer1.0-plugins-good-cutter*	
gstreamer1.0-plugins-good-debug	*gstreamer1.0-plugins-good-deinterlace*	
gstreamer1.0-plugins-good-dtmf	*gstreamer1.0-plugins-good-effectv*	
gstreamer1.0-plugins-good-equalizer	*gstreamer1.0-plugins-good-flac*	
gstreamer1.0-plugins-good-flv	*gstreamer1.0-plugins-good-flxdec*	
gstreamer1.0-plugins-good-gdkpixbuf	*gstreamer1.0-plugins-good-goom*	
gstreamer1.0-plugins-good-goom2k1	*gstreamer1.0-plugins-good-icydemux*	
gstreamer1.0-plugins-good-id3demux	*gstreamer1.0-plugins-good-imagefreeze*	
gstreamer1.0-plugins-good-interleave	*gstreamer1.0-plugins-good-isomp4*	
gstreamer1.0-plugins-good-jpeg	*gstreamer1.0-plugins-good-lame*	
gstreamer1.0-plugins-good-level	*gstreamer1.0-plugins-good-locale-en-gb*	
gstreamer1.0-plugins-good-matroska	*gstreamer1.0-plugins-good-meta*	
gstreamer1.0-plugins-good-monoscope	*gstreamer1.0-plugins-good-mpg123*	
gstreamer1.0-plugins-good-mulaw	*gstreamer1.0-plugins-good-multifile*	
gstreamer1.0-plugins-good-multipart	*gstreamer1.0-plugins-good-navigationtest*	
gstreamer1.0-plugins-good-ossaudio	*gstreamer1.0-plugins-good-png*	
gstreamer1.0-plugins-good-pulseaudio	*gstreamer1.0-plugins-good-qmlgl*	
gstreamer1.0-plugins-good-replaygain	*gstreamer1.0-plugins-good-rtp*	
gstreamer1.0-plugins-good-rtpmanager	*gstreamer1.0-plugins-good-rtsp*	
gstreamer1.0-plugins-good-shapewipe	*gstreamer1.0-plugins-good-smpte*	
gstreamer1.0-plugins-good-soup	*gstreamer1.0-plugins-good-spectrum*	
gstreamer1.0-plugins-good-speex	*gstreamer1.0-plugins-good-taglib*	
gstreamer1.0-plugins-good-udp	*gstreamer1.0-plugins-good-video4linux2*	
gstreamer1.0-plugins-good-videobox	*gstreamer1.0-plugins-good-videocrop*	
gstreamer1.0-plugins-good-videofilter	*gstreamer1.0-plugins-good-videomixer*	
gstreamer1.0-plugins-good-vpx	*gstreamer1.0-plugins-good-wavenc*	
gstreamer1.0-plugins-good-wavparse	*gstreamer1.0-plugins-good-ximagesrc*	
gstreamer1.0-plugins-good-y4menc		
gstreamer1.0-plugins-ugly	1.18.4	GPLv2+, LGPLv2.1+, LGPLv2+
gstreamer1.0-plugins-ugly-a52dec	*gstreamer1.0-plugins-ugly-asf*	
gstreamer1.0-plugins-ugly-dvdlpcmdec	*gstreamer1.0-plugins-ugly-dvdsub*	
gstreamer1.0-plugins-ugly-locale-en-gb	*gstreamer1.0-plugins-ugly-meta*	
gstreamer1.0-plugins-ugly-mpeg2dec	*gstreamer1.0-plugins-ugly-realmedia*	
gstreamer1.0-plugins-ugly-xingmux		
gstreamer1.0-vaapi	1.18.4	LGPLv2.1+
gstreamer1.0-vaapi		
gtk+3	3.24.25	LGPLv2, LGPLv2+, LGPLv2.1+
gtk+3	*gtk+3-locale-en*	
gtk+3-locale-en-gb		
gvfs	1.44.1	LGPLv2
gvfs	*gvfs-locale-en-gb*	
gvfsd-trash		
harfbuzz	2.7.4	MIT
harfbuzz		
hdparm	9.60	BSD
hdparm		
hicolor-icon-theme	0.17	GPLv2
hicolor-icon-theme		
hostapd	2.9	BSD-3-Clause
hostapd		
hsetroot	1.0.2	GPLv2
hsetroot		
htpdate	1.1.3	GPLv2
htpdate		
hunspell	1.7.0	GPLv2, LGPLv2.1
hunspell		

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
i2c-ids *i2c-ids*	1.0	MIT
i2c-tools *i2c-tools*	4.2	GPLv2+
i2c-write-read *i2c-write-read*	git	GPLv2
icu *libicudata* *libicuuc*	68.2 *libicui18n*	ICU
igt-gpu-tools *igt-gpu-tools*	1.25	MIT
imlib2 *imlib2*	1.4.6 *imlib2-loaders*	MIT, BSD
init-ifupdown *init-ifupdown*	1.0	GPLv2
init-system-helpers *init-system-helpers-service*	1.60	BSD-3-Clause, GPLv2
initscripts *initscripts*	1.0 *initscripts-functions*	GPLv2
intel-media-driver *intel-media-driver*	21.1.3	MIT, BSD-3-Clause
intel-mediasdk *intel-mediasdk*	21.1.3	MIT
intel-microcode *intel-microcode*	20210216	Intel-Microcode-License
intel-vaapi-driver *intel-vaapi-driver*	2.4.1	MIT
iperf3 *iperf3*	3.9	BSD-3-Clause
iproute2 *iproute2*	5.11.0 *iproute2-ip*	GPLv2+
iptables *iptables* *iptables-module-ip6t-dnat* *iptables-module-ip6t-dst* *iptables-module-ip6t-frag* *iptables-module-ip6t-hl* *iptables-module-ip6t-ipv6header* *iptables-module-ip6t-masquerade* *iptables-module-ip6t-netmap* *iptables-module-ip6t-reject* *iptables-module-ip6t-snat* *iptables-module-ip6t-srh* *iptables-module-ipt-clusterip* *iptables-module-ipt-ecn* *iptables-module-ipt-log* *iptables-module-ipt-netmap* *iptables-module-ipt-redirect* *iptables-module-ipt-snat* *iptables-module-ipt-ulog* *iptables-module-xt-audit* *iptables-module-xt-cgroup* *iptables-module-xt-classify* *iptables-module-xt-comment* *iptables-module-xt-connlimit* *iptables-module-xt-connsecmark* *iptables-module-xt-cpu* *iptables-module-xt-dccp* *iptables-module-xt-dscp* *iptables-module-xt-esp* *iptables-module-xt-helper* *iptables-module-xt-idletimer*	1.8.7 *iptables-module-ip6t-ah* *iptables-module-ip6t-dnpt* *iptables-module-ip6t-eui64* *iptables-module-ip6t-hbh* *iptables-module-ip6t-icmp6* *iptables-module-ip6t-log* *iptables-module-ip6t-mh* *iptables-module-ip6t-redirect* *iptables-module-ip6t-rt* *iptables-module-ip6t-snpt* *iptables-module-ipt-ah* *iptables-module-ipt-dnat* *iptables-module-ipt-icmp* *iptables-module-ipt-masquerade* *iptables-module-ipt-realm* *iptables-module-ipt-reject* *iptables-module-ipt-ttl* *iptables-module-xt-addrtype* *iptables-module-xt-bpf* *iptables-module-xt-checksum* *iptables-module-xt-cluster* *iptables-module-xt-connbytes* *iptables-module-xt-connmark* *iptables-module-xt-conntrack* *iptables-module-xt-ct* *iptables-module-xt-devgroup* *iptables-module-xt-ecn* *iptables-module-xt-hashlimit* *iptables-module-xt-hmark* *iptables-module-xt-ipcomp*	GPLv2+

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
iptables-module-xt-iprange	*iptables-module-xt-ipvs*	
iptables-module-xt-led	*iptables-module-xt-length*	
iptables-module-xt-limit	*iptables-module-xt-mac*	
iptables-module-xt-mark	*iptables-module-xt-multiport*	
iptables-module-xt-nfacct	*iptables-module-xt-nflog*	
iptables-module-xt-nfqueue	*iptables-module-xt-osf*	
iptables-module-xt-owner	*iptables-module-xt-physdev*	
iptables-module-xt-pkttype	*iptables-module-xt-policy*	
iptables-module-xt-quota	*iptables-module-xt-rateest*	
iptables-module-xt-recent	*iptables-module-xt-rpfilter*	
iptables-module-xt-sctp	*iptables-module-xt-secmark*	
iptables-module-xt-set	*iptables-module-xt-socket*	
iptables-module-xt-standard	*iptables-module-xt-statistic*	
iptables-module-xt-string	*iptables-module-xt-synproxy*	
iptables-module-xt-tcp	*iptables-module-xt-tcpmss*	
iptables-module-xt-tcpoptstrip	*iptables-module-xt-tee*	
iptables-module-xt-time	*iptables-module-xt-tos*	
iptables-module-xt-tproxy	*iptables-module-xt-trace*	
iptables-module-xt-u32	*iptables-module-xt-udp*	
iptables-modules		
iw	5.9	BSD-2-Clause
iw		
json-c	0.15	MIT
json-c		
json-glib	1.6.2	LGPLv2.1
json-glib	*json-glib-locale-en-gb*	
kbd	2.4.0	GPLv2+
kbd	*kbd-keymaps*	
kexec-tools	2.0.21	GPLv2
kexec		
keymaps	1.0	GPLv2
keymaps		
keyutils	1.6.1	LGPLv2.1+, GPLv2.0+
keyutils		
kidletime	5.83.0	GPL-2.0+, LGPL-2.0, MIT
kidletime		
kmod	28	GPL-2.0+, LGPL-2.1+
kmod	*libkmod*	
kwayland	5.83.0	BSD-3-Clause, LGPL-2.1, LGPL-2.1+, LGPL-3.0, MIT, MIT-CMU
kwayland		
kwindowsystem	5.83.0	LGPL-2.1, LGPL-2.1+, LGPL-3.0, MIT
kwindowsystem	*kwindowsystem-locale-en-gb*	
l3afpad	0.8.18.1.11	GPLv2+
l3afpad		
lame	3.100	LGPLv2+
libmp3lame		
lcms	2.12	MIT
lcms		
lib64-compatibility	1.0	MIT
lib64-compatibility		
liba52	0.7.4	GPLv2+

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
liba52		
libaio *libaio*	0.3.112	LGPLv2.1+
libassuan *libassuan*	2.5.4	LGPLv2.1+
libatasmart *libatasmart*	0.19	LGPLv2.1
libblockdev *libblockdev*	2.25	LGPLv2+
libbytesize *libbytesize*	2.4	LGPLv2+
libcap *libcap*	2.48	BSD, GPLv2
libcomps *libcomps*	0.1.15	GPLv2
libconfig *libconfig*	1.7.2	LGPLv2.1
libcroco *libcroco*	0.6.13	LGPLv2, LGPLv2.1
libdaemon *libdaemon*	0.14	LGPLv2.1+
libdbusmenu-qt5 *libdbusmenu-qt5*	0.9.3	GPLv2
libdmx *libdmx*	1.1.4	MIT
libdnf *libdnf*	0.58.0	LGPLv2.1+
libdrm *libdrm* *libdrm-nouveau*	2.4.104 *libdrm-intel* *libdrm-radeon*	MIT
libepoxy *libepoxy*	1.5.5	MIT
liberation-fonts *liberation-fonts*	2.00.1	OFL-1.1
libev *libev*	4.33	BSD-2-Clause, GPL-2.0+
libevdev *libevdev*	1.11.0	MIT
libexif *libexif*	0.6.22 *libexif-locale-en-gb*	LGPLv2.1
libffi *libffi*	3.3	MIT
libfm-extra *libfm-extra*	1.3.2	LGPLv2+
libfm-qt *libfm-qt*	0.17.1 *libfm-qt-locale-en-gb*	LGPL-2.1
libfontenc *libfontenc*	1.1.4	MIT
libgcc *libgcc*	10.2.0	GPL-3.0-with-GCC-exception
libgcrypt *libgcrypt*	1.9.2	LGPLv2.1+
libgpg-error *libgpg-error*	1.41	GPLv2+, LGPLv2.1+
libgphoto2	2.5.27	LGPLv2.1

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
libgphoto2 *libgphotoport*	*libgphoto2-camlibs*	
libgudev *libgudev*	234	LGPLv2.1
libical *libical*	3.0.9	LGPLv2.1, MPL-2.0
libice *libice*	1.0.10	MIT-style
libidn2 *libidn2*	2.3.0	(GPLv2+ | LGPLv3)
libinput *libinput*	1.16.4	MIT
libjitterentropy *libjitterentropy*	3.0.1	GPLv2+, BSD
libjpeg-turbo *libjpeg-turbo*	2.0.6	BSD-3-Clause
libksba *libksba*	1.5.0	GPLv2+, LGPLv3+
libkscreen *libkscreen*	5.22.2.1	GPL-2.0-only, GPL-2.0-or-later, GPL-3.0-only, LGPL-2.1-or-later
liblxqt *liblxqt*	0.17.0 *liblxqt-locale-en-gb*	LGPL-2.1
libmnl *libmnl*	1.0.4	LGPLv2.1+
libmodulemd *libmodulemd*	2.12.0	MIT
libmscboost *libmscboost*	git	LGPLv2.1
libmscboostpython *libmscboostpython*	git	LGPLv2.1
libmscdrm *libmscdrm*	git	LGPLv2.1
libnl *libnl*	3.5.0 *libnl-genl*	LGPLv2.1
libnotify *libnotify*	0.7.9	LGPLv2.1
libnsl2 *libnsl2*	1.3.0	LGPL-2.1
libnss-mdns *libnss-mdns*	0.14.1	LGPLv2.1+
libogg *libogg*	1.3.4	BSD-3-Clause
libpcap *libpcap*	1.10.0	BSD-3-Clause
libpciaccess *libpciaccess*	0.16	MIT, MIT-style
libpcre *libpcre*	8.44	BSD-3-Clause
libpcre2 *libpcre2*	10.36 *libpcre2-16*	BSD-3-Clause
libpng *libpng*	1.6.37	Libpng
libpsl	0.21.1	MIT

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
libpsl		
libqtxdg *libqtxdg*	3.7.1	LGPL-2.1
librepo *librepo*	1.13.0	LGPLv2.1
librsvg *librsvg*	2.40.21	LGPLv2+
libsamplerate0 *libsamplerate0*	0.1.9	BSD-2-Clause
libsdl *libsdl*	1.2.15	LGPLv2.1
libsecret *libsecret*	0.20.4 *libsecret-locale-en-gb*	LGPLv2.1
libsm *libsm*	1.2.3	MIT-style
libsndfile1 *libsndfile1*	1.0.28	LGPLv2.1
libsolv *libsolv*	0.7.17 *libsolvext*	BSD-3-Clause
libsoup-2.4 *libsoup-2.4*	2.72.0 *libsoup-2.4-locale-en-gb*	LGPLv2
libstatgrab *libstatgrab*	0.92	GPL-2.0+
libsysstat *libsysstat*	0.4.5	LGPL-2.1
libtheora *libtheora*	1.1.1	BSD-3-Clause
libtirpc *libtirpc*	1.3.1	BSD-3-Clause
libtool *libltdl*	2.4.6	GPLv2, LGPLv2.1
libunistring *libunistring*	0.9.10	LGPLv3+, GPLv2
libunwind *libunwind*	1.5.0	MIT
libusb1 *libusb1*	1.0.23	LGPLv2.1+
libva-intel *libva-intel* *libva-intel-x11*	2.12.0 *libva-intel-wayland*	MIT
libva-intel-utils *libva-intel-utils*	2.11.1	MIT
libvorbis *libvorbis*	1.3.7	BSD-3-Clause
libvpx *libvpx*	1.8.2	BSD-3-Clause
libwebp *libwebp*	1.2.0	BSD-3-Clause
libx11 *libx11* *libx11-xcb*	1.7.0 *libx11-locale*	MIT, MIT-style, BSD
libxau *libxau*	1.0.9	MIT-style
libxaw *libxaw7*	1.0.14	MIT-X
libxcb	1.14	MIT

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
libxcb *libxcb-damage* *libxcb-dri3* *libxcb-present* *libxcb-render* *libxcb-shape* *libxcb-sync* *libxcb-xinerama* *libxcb-xkb*	*libxcb-composite* *libxcb-dri2* *libxcb-glx* *libxcb-randr* *libxcb-res* *libxcb-shm* *libxcb-xfixes* *libxcb-xinput*	
libxcomposite *libxcomposite*	0.4.5	MIT-style
libxcrypt *libxcrypt*	4.4.18	LGPLv2.1
libxcursor *libxcursor*	1.2.0	MIT-style
libxdamage *libxdamage*	1.1.5	MIT
libxdmcp *libxdmcp*	1.1.3	MIT-style
libxext *libxext*	1.3.4	MIT-style
libxfce4ui *libxfce4ui*	4.16.0 *libxfce4ui-locale-en-gb*	GPLv2
libxfce4util *libxfce4util*	4.16.0 *libxfce4util-locale-en-gb*	GPLv2
libxfixes *libxfixes*	5.0.3	MIT-style
libxfont2 *libxfont2*	2.0.4	MIT, MIT-style, BSD
libxft *libxft*	2.3.3	MIT
libxi *libxi*	1.7.10	MIT, MIT-style
libxinerama *libxinerama*	1.1.4	MIT
libxkbcommon *libxkbcommon*	1.0.3	MIT, MIT-style
libxkbfile *libxkbfile*	1.1.0	MIT-style
libxml2 *libxml2*	2.9.10 *libxml2-utils*	MIT
libxmu *libxmu*	1.1.3 *libxmuu*	MIT, MIT-style
libxpm *libxpm*	3.5.13	MIT
libxrandr *libxrandr*	1.5.2	MIT-style
libxrender *libxrender*	0.9.10	MIT-style
libxscrnsaver *libxscrnsaver*	1.2.3	MIT
libxshmfence *libxshmfence*	1.3	MIT-style
libxslt *libxslt*	1.1.34	MIT
libxt *libxt*	1.2.1	MIT, MIT-style

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
libxtst *libxtst*	1.2.3	MIT-style
libxv *libxv*	1.0.11	MIT-style
libxvmc *libxvmc*	1.0.12	MIT
libxxf86vm *libxxf86vm*	1.1.4	MIT
libyaml *libyaml*	0.2.5	MIT
linux-audio-firmware *linux-audio-firmware*	1.0	Proprietary
linux-firmware	20210511	Firmware-Abilis, Firmware-agere, Firmware-amdgpu, Firmware-amd-ucode, Firmware-amlogic_vdec, Firmware-atmel, Firmware-ca0132, Firmware-cavium, Firmware-chelsio_firmware, Firmware-cw1200, Firmware-dib0700, Firmware-e100, Firmware-ene_firmware, Firmware-fw_sst_0f28, Firmware-go7007, Firmware-hfi1_firmware, Firmware-i2400m, Firmware-ibt_firmware, Firmware-it913x, Firmware-IntcSST2, Firmware-kaweth, Firmware-moxa, Firmware-myri10ge_firmware, Firmware-nvidia, Firmware-OLPC, Firmware-ath9k-htc, Firmware-phanfw, Firmware-qat, Firmware-qcom, Firmware-qla1280, Firmware-qla2xxx, Firmware-r8a779x_usb3, Firmware-radeon, Firmware-ralink_a_mediatek_company_firmware, Firmware-ralink-firmware, Firmware-imx-sdma_firmware, Firmware-siano, Firmware-tda7706-firmware, Firmware-ti-connectivity, Firmware-ti-keystone, Firmware-ueagle-atm4-firmware, Firmware-wl1251, Firmware-xc4000, Firmware-xc5000, Firmware-xc5000c, WHENCE
linux-firmware	*linux-firmware-adsp-sst*	
linux-firmware-adsp-sst-license	*linux-firmware-amlogic-vdec*	
linux-firmware-amlogic-vdec-license	*linux-firmware-ar3k*	
linux-firmware-ar3k-license	*linux-firmware-ar9170*	
linux-firmware-ath10k	*linux-firmware-ath10k-license*	
linux-firmware-ath11k	*linux-firmware-ath6k*	

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
linux-firmware-ath9k	*linux-firmware-atheros-license*	
linux-firmware-bcm-0bb4-0306	*linux-firmware-bcm43143*	
linux-firmware-bcm43236b	*linux-firmware-bcm43241b0*	
linux-firmware-bcm43241b4	*linux-firmware-bcm43241b5*	
linux-firmware-bcm43242a	*linux-firmware-bcm4329*	
linux-firmware-bcm4329-fullmac	*linux-firmware-bcm4330*	
linux-firmware-bcm4334	*linux-firmware-bcm43340*	
linux-firmware-bcm4335	*linux-firmware-bcm43362*	
linux-firmware-bcm4339	*linux-firmware-bcm43430*	
linux-firmware-bcm43430a0	*linux-firmware-bcm43455*	
linux-firmware-bcm4350	*linux-firmware-bcm4350c2*	
linux-firmware-bcm4354	*linux-firmware-bcm4356*	
linux-firmware-bcm4356-pcie	*linux-firmware-bcm43569*	
linux-firmware-bcm43570	*linux-firmware-bcm4358*	
linux-firmware-bcm43602	*linux-firmware-bcm4366b*	
linux-firmware-bcm4366c	*linux-firmware-bcm4371*	
linux-firmware-bcm4373	*linux-firmware-bcm43xx*	
linux-firmware-bcm43xx-hdr	*linux-firmware-bnx2-mips*	
linux-firmware-broadcom-license	*linux-firmware-carl9170*	
linux-firmware-cypress-license	*linux-firmware-gplv2-license*	
linux-firmware-i915	*linux-firmware-i915-license*	
linux-firmware-ibt	*linux-firmware-ibt-11-5*	
linux-firmware-ibt-12-16	*linux-firmware-ibt-17*	
linux-firmware-ibt-20	*linux-firmware-ibt-hw-37-7*	
linux-firmware-ibt-hw-37-8	*linux-firmware-ibt-license*	
linux-firmware-ibt-misc	*linux-firmware-ice*	
linux-firmware-ice-license	*linux-firmware-imx-sdma-imx6q*	
linux-firmware-imx-sdma-imx7d	*linux-firmware-imx-sdma-license*	
linux-firmware-iwlwifi	*linux-firmware-iwlwifi-135-6*	
linux-firmware-iwlwifi-3160-10	*linux-firmware-iwlwifi-3160-12*	
linux-firmware-iwlwifi-3160-13	*linux-firmware-iwlwifi-3160-16*	
linux-firmware-iwlwifi-3160-17	*linux-firmware-iwlwifi-3160-7*	
linux-firmware-iwlwifi-3160-8	*linux-firmware-iwlwifi-3160-9*	
linux-firmware-iwlwifi-6000-4	*linux-firmware-iwlwifi-6000g2a-5*	
linux-firmware-iwlwifi-6000g2a-6	*linux-firmware-iwlwifi-6000g2b-5*	
linux-firmware-iwlwifi-6000g2b-6	*linux-firmware-iwlwifi-6050-4*	
linux-firmware-iwlwifi-6050-5	*linux-firmware-iwlwifi-7260*	
linux-firmware-iwlwifi-7265	*linux-firmware-iwlwifi-7265d*	
linux-firmware-iwlwifi-8000c	*linux-firmware-iwlwifi-8265*	
linux-firmware-iwlwifi-9000	*linux-firmware-iwlwifi-license*	
linux-firmware-iwlwifi-misc	*linux-firmware-license*	
linux-firmware-liquidio	*linux-firmware-lontium-license*	
linux-firmware-lt9611uxc	*linux-firmware-marvell-license*	
linux-firmware-mt7601u	*linux-firmware-mt7601u-license*	
linux-firmware-netronome	*linux-firmware-netronome-license*	
linux-firmware-nvidia-gpu	*linux-firmware-nvidia-license*	
linux-firmware-nvidia-tegra	*linux-firmware-nvidia-tegra-k1*	
linux-firmware-pcie8897	*linux-firmware-pcie8997*	
linux-firmware-qat	*linux-firmware-qat-license*	
linux-firmware-qca	*linux-firmware-qcom-adreno-a3xx*	
linux-firmware-qcom-adreno-a530	*linux-firmware-qcom-adreno-a630*	
linux-firmware-qcom-license	*linux-firmware-qcom-sdm845-audio*	
linux-firmware-qcom-sdm845-compute	*linux-firmware-qcom-sdm845-modem*	
linux-firmware-qcom-venus-1.8	*linux-firmware-qcom-venus-4.2*	
linux-firmware-qcom-venus-5.2	*linux-firmware-qcom-venus-5.4*	
linux-firmware-radeon	*linux-firmware-radeon-license*	
linux-firmware-ralink	*linux-firmware-ralink-license*	
linux-firmware-rtl-license	*linux-firmware-rtl8168*	
linux-firmware-rtl8188	*linux-firmware-rtl8192ce*	
linux-firmware-rtl8192cu	*linux-firmware-rtl8192su*	
linux-firmware-rtl8723	*linux-firmware-rtl8821*	
linux-firmware-sd8686	*linux-firmware-sd8688*	
linux-firmware-sd8787	*linux-firmware-sd8797*	
linux-firmware-sd8801	*linux-firmware-sd8887*	
linux-firmware-sd8897	*linux-firmware-sd8997*	
linux-firmware-ti-connectivity-license	*linux-firmware-usb8997*	
linux-firmware-vt6656	*linux-firmware-vt6656-license*	
linux-firmware-whence-license	*linux-firmware-wl12xx*	
linux-firmware-wl18xx	*linux-firmware-wlcommon*	
linux-tools-testusb	1.0	GPL-2.0

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
linux-tools-testusb		
linux-yocto-custom	5.14	GPLv2
kernel	*kernel-base*	
kernel-image	*kernel-image-bzimage*	
kernel-module-8021q-5.14.0-yocto-standard-custom	*kernel-module-8250-exar-5.14.0-yocto-standard-custom*	
kernel-module-acp-audio-dma-5.14.0-yocto-standard-custom	*kernel-module-acpi-mdio-5.14.0-yocto-standard-custom*	
kernel-module-acpi-thermal-rel-5.14.0-yocto-standard-custom	*kernel-module-af-key-5.14.0-yocto-standard-custom*	
kernel-module-asix-5.14.0-yocto-standard-custom	*kernel-module-at24-5.14.0-yocto-standard-custom*	
kernel-module-at25-5.14.0-yocto-standard-custom	*kernel-module-ath-5.14.0-yocto-standard-custom*	
kernel-module-ath5k-5.14.0-yocto-standard-custom	*kernel-module-ath9k-5.14.0-yocto-standard-custom*	
kernel-module-ath9k-common-5.14.0-yocto-standard-custom	*kernel-module-ath9k-hw-5.14.0-yocto-standard-custom*	
kernel-module-atkbd-5.14.0-yocto-standard-custom	*kernel-module-ax88179-178a-5.14.0-yocto-standard-custom*	
kernel-module-ax88796b-5.14.0-yocto-standard-custom	*kernel-module-baytrail-edac-5.14.0-yocto-standard-custom*	
kernel-module-binfmt-misc-5.14.0-yocto-standard-custom	*kernel-module-blowfish-common-5.14.0-yocto-standard-custom*	
kernel-module-blowfish-generic-5.14.0-yocto-standard-custom	*kernel-module-bnep-5.14.0-yocto-standard-custom*	
kernel-module-bonding-5.14.0-yocto-standard-custom	*kernel-module-bq25890-charger-5.14.0-yocto-standard-custom*	
kernel-module-btbcm-5.14.0-yocto-standard-custom	*kernel-module-btintel-5.14.0-yocto-standard-custom*	
kernel-module-btmrvl-5.14.0-yocto-standard-custom	*kernel-module-btmrvl-sdio-5.14.0-yocto-standard-custom*	
kernel-module-btrtl-5.14.0-yocto-standard-custom	*kernel-module-btusb-5.14.0-yocto-standard-custom*	
kernel-module-camellia-generic-5.14.0-yocto-standard-custom	*kernel-module-can-5.14.0-yocto-standard-custom*	
kernel-module-can-bcm-5.14.0-yocto-standard-custom	*kernel-module-can-dev-5.14.0-yocto-standard-custom*	
kernel-module-can-gw-5.14.0-yocto-standard-custom	*kernel-module-can-raw-5.14.0-yocto-standard-custom*	
kernel-module-cast-common-5.14.0-yocto-standard-custom	*kernel-module-cast5-generic-5.14.0-yocto-standard-custom*	
kernel-module-cast6-generic-5.14.0-yocto-standard-custom	*kernel-module-cdc-acm-5.14.0-yocto-standard-custom*	
kernel-module-cdc-eem-5.14.0-yocto-standard-custom	*kernel-module-cdc-ether-5.14.0-yocto-standard-custom*	
kernel-module-cdc-ncm-5.14.0-yocto-standard-custom	*kernel-module-cdc-subset-5.14.0-yocto-standard-custom*	
kernel-module-cdc-wdm-5.14.0-yocto-standard-custom	*kernel-module-cfg80211-5.14.0-yocto-standard-custom*	
kernel-module-ch7006-5.14.0-yocto-standard-custom	*kernel-module-chipreg-5.14.0-yocto-standard-custom*	
kernel-module-cifs-5.14.0-yocto-standard-custom	*kernel-module-cmdlinepart-5.14.0-yocto-standard-custom*	
kernel-module-configfs-5.14.0-yocto-standard-custom	*kernel-module-cordic-5.14.0-yocto-standard-custom*	
kernel-module-coretemp-5.14.0-yocto-standard-custom	*kernel-module-crc-ccitt-5.14.0-yocto-standard-custom*	
kernel-module-crc-itu-t-5.14.0-yocto-standard-custom	*kernel-module-cryptoloop-5.14.0-yocto-standard-custom*	
kernel-module-cuse-5.14.0-yocto-standard-custom	*kernel-module-diag-5.14.0-yocto-standard-custom*	
kernel-module-dm9601-5.14.0-yocto-standard-custom	*kernel-module-dme1737-5.14.0-yocto-standard-custom*	
kernel-module-dp83867-5.14.0-yocto-standard-custom	*kernel-module-drm-ttm-helper-5.14.0-yocto-standard-custom*	
kernel-module-dummy-5.14.0-yocto-standard-custom	*kernel-module-dummy-hcd-5.14.0-yocto-standard-custom*	
kernel-module-dw-dmac-5.14.0-yocto-standard-custom	*kernel-module-dwc2-5.14.0-yocto-standard-custom*	
kernel-module-dwc2-pci-5.14.0-yocto-standard-custom	*kernel-module-dwc3-5.14.0-yocto-standard-custom*	
kernel-module-dwc3-haps-5.14.0-yocto-standard-custom	*kernel-module-dwc3-pci-5.14.0-yocto-standard-custom*	
kernel-module-dwmac-generic-5.14.0-yocto-standard-custom	*kernel-module-dwmac-intel-5.14.0-yocto-standard-custom*	
kernel-module-e100-5.14.0-yocto-standard-custom	*kernel-module-e1000-5.14.0-yocto-standard-custom*	
kernel-module-e1000e-5.14.0-yocto-standard-custom	*kernel-module-e752x-edac-5.14.0-yocto-standard-custom*	
kernel-module-efivars-5.14.0-yocto-standard-custom	*kernel-module-fixed-phy-5.14.0-yocto-standard-custom*	
kernel-module-ftdi-sio-5.14.0-yocto-standard-custom	*kernel-module-fuse-5.14.0-yocto-standard-custom*	
kernel-module-fwnode-mdio-5.14.0-yocto-standard-custom	*kernel-module-g-acm-ms-5.14.0-yocto-standard-custom*	
kernel-module-g-audio-5.14.0-yocto-standard-custom	*kernel-module-g-cdc-5.14.0-yocto-standard-custom*	
kernel-module-g-dbgp-5.14.0-yocto-standard-custom	*kernel-module-g-ether-5.14.0-yocto-standard-custom*	
kernel-module-g-ffs-5.14.0-yocto-standard-custom	*kernel-module-g-hid-5.14.0-yocto-standard-custom*	
kernel-module-g-mass-storage-5.14.0-yocto-standard-custom	*kernel-module-g-midi-5.14.0-yocto-standard-custom*	
kernel-module-g-multi-5.14.0-yocto-standard-custom	*kernel-module-g-ncm-5.14.0-yocto-standard-custom*	
kernel-module-g-printer-5.14.0-yocto-standard-custom	*kernel-module-g-serial-5.14.0-yocto-standard-custom*	
kernel-module-g-zero-5.14.0-yocto-standard-custom	*kernel-module-gadgetfs-5.14.0-yocto-standard-custom*	
kernel-module-gpio-exar-5.14.0-yocto-standard-custom	*kernel-module-gpio-keys-5.14.0-yocto-standard-custom*	
kernel-module-gpio-pca953x-5.14.0-yocto-standard-custom	*kernel-module-gpio-wcove-5.14.0-yocto-standard-custom*	
kernel-module-hci-uart-5.14.0-yocto-standard-custom	*kernel-module-hid-sensor-hub-5.14.0-yocto-standard-custom*	
kernel-module-hidp-5.14.0-yocto-standard-custom	*kernel-module-hwmon-vid-5.14.0-yocto-standard-custom*	
kernel-module-i2c-designware-pci-5.14.0-yocto-standard-custom	*kernel-module-i2c-i801-5.14.0-yocto-standard-custom*	
kernel-module-i2c-ismt-5.14.0-yocto-standard-custom	*kernel-module-i2c-smbus-5.14.0-yocto-standard-custom*	
kernel-module-i3000-edac-5.14.0-yocto-standard-custom	*kernel-module-i3200-edac-5.14.0-yocto-standard-custom*	
kernel-module-i5000-edac-5.14.0-yocto-standard-custom	*kernel-module-i5100-edac-5.14.0-yocto-standard-custom*	
kernel-module-i5400-edac-5.14.0-yocto-standard-custom	*kernel-module-i7300-edac-5.14.0-yocto-standard-custom*	
kernel-module-i7core-edac-5.14.0-yocto-standard-custom	*kernel-module-i8042-5.14.0-yocto-standard-custom*	
kernel-module-i82975x-edac-5.14.0-yocto-standard-custom	*kernel-module-i915-5.14.0-yocto-standard-custom*	
kernel-module-igb-5.14.0-yocto-standard-custom	*kernel-module-igbvf-5.14.0-yocto-standard-custom*	
kernel-module-int3400-thermal-5.14.0-yocto-standard-custom	*kernel-module-int3401-thermal-5.14.0-yocto-standard-custom*	
kernel-module-int3402-thermal-5.14.0-yocto-standard-custom	*kernel-module-int3403-thermal-5.14.0-yocto-standard-custom*	
kernel-module-int340x-thermal-zone-5.14.0-yocto-standard-custom	*kernel-module-intel-bxt-pmic-thermal-5.14.0-yocto-standard-custom*	
kernel-module-intel-lpss-5.14.0-yocto-standard-custom	*kernel-module-intel-lpss-acpi-5.14.0-yocto-standard-custom*	
kernel-module-intel-lpss-pci-5.14.0-yocto-standard-custom	*kernel-module-intel-pch-thermal-5.14.0-yocto-standard-custom*	
kernel-module-intel-pmc-bxt-5.14.0-yocto-standard-custom	*kernel-module-intel-powerclamp-5.14.0-yocto-standard-custom*	

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
kernel-module-intel-rapl-common-5.14.0-yocto-standard-custom		*kernel-module-intel-rapl-msr-5.14.0-yocto-standard-custom*
kernel-module-intel-soc-dts-iosf-5.14.0-yocto-standard-custom		*kernel-module-intel-soc-dts-thermal-5.14.0-yocto-standard-custom*
kernel-module-intel-soc-pmic-bxtwc-5.14.0-yocto-standard-custom		*kernel-module-intel-spi-5.14.0-yocto-standard-custom*
kernel-module-intel-spi-pci-5.14.0-yocto-standard-custom		*kernel-module-intel-spi-platform-5.14.0-yocto-standard-custom*
kernel-module-intel-telemetry-core-5.14.0-yocto-standard-custom		*kernel-module-intel-telemetry-debugfs-5.14.0-yocto-standard-custom*
kernel-module-intel-telemetry-pltdrv-5.14.0-yocto-standard-custom		*kernel-module-ip-tunnel-5.14.0-yocto-standard-custom*
kernel-module-ipmi-devintf-5.14.0-yocto-standard-custom		*kernel-module-ipmi-msghandler-5.14.0-yocto-standard-custom*
kernel-module-ipmi-poweroff-5.14.0-yocto-standard-custom		*kernel-module-ipmi-si-5.14.0-yocto-standard-custom*
kernel-module-ipmi-watchdog-5.14.0-yocto-standard-custom		*kernel-module-isst-if-common-5.14.0-yocto-standard-custom*
kernel-module-isst-if-mbox-msr-5.14.0-yocto-standard-custom		*kernel-module-isst-if-mbox-pci-5.14.0-yocto-standard-custom*
kernel-module-isst-if-mmio-5.14.0-yocto-standard-custom		*kernel-module-itco-wdt-5.14.0-yocto-standard-custom*
kernel-module-iwl3945-5.14.0-yocto-standard-custom		*kernel-module-iwl4965-5.14.0-yocto-standard-custom*
kernel-module-iwldvm-5.14.0-yocto-standard-custom		*kernel-module-iwlegacy-5.14.0-yocto-standard-custom*
kernel-module-iwlmvm-5.14.0-yocto-standard-custom		*kernel-module-iwlwifi-5.14.0-yocto-standard-custom*
kernel-module-ixgbe-5.14.0-yocto-standard-custom		*kernel-module-jme-5.14.0-yocto-standard-custom*
kernel-module-kaweth-5.14.0-yocto-standard-custom		*kernel-module-ledtrig-audio-5.14.0-yocto-standard-custom*
kernel-module-libarc4-5.14.0-yocto-standard-custom		*kernel-module-libcomposite-5.14.0-yocto-standard-custom*
kernel-module-libphy-5.14.0-yocto-standard-custom		*kernel-module-libps2-5.14.0-yocto-standard-custom*
kernel-module-lm75-5.14.0-yocto-standard-custom		*kernel-module-lp-5.14.0-yocto-standard-custom*
kernel-module-lpc-ich-5.14.0-yocto-standard-custom		*kernel-module-m-can-5.14.0-yocto-standard-custom*
kernel-module-m-can-pci-5.14.0-yocto-standard-custom		*kernel-module-mac80211-5.14.0-yocto-standard-custom*
kernel-module-mce-inject-5.14.0-yocto-standard-custom		*kernel-module-mcs7830-5.14.0-yocto-standard-custom*
kernel-module-mdio-5.14.0-yocto-standard-custom		*kernel-module-mdio-devres-5.14.0-yocto-standard-custom*
kernel-module-mei-5.14.0-yocto-standard-custom		*kernel-module-mei-me-5.14.0-yocto-standard-custom*
kernel-module-mei-txe-5.14.0-yocto-standard-custom		*kernel-module-mei-wdt-5.14.0-yocto-standard-custom*
kernel-module-michael-mic-5.14.0-yocto-standard-custom		*kernel-module-mii-5.14.0-yocto-standard-custom*
kernel-module-mt7601u-5.14.0-yocto-standard-custom		*kernel-module-mtd-5.14.0-yocto-standard-custom*
kernel-module-mwifiex-5.14.0-yocto-standard-custom		*kernel-module-mwifiex-sdio-5.14.0-yocto-standard-custom*
kernel-module-mxm-wmi-5.14.0-yocto-standard-custom		*kernel-module-nbd-5.14.0-yocto-standard-custom*
kernel-module-net1080-5.14.0-yocto-standard-custom		*kernel-module-net2280-5.14.0-yocto-standard-custom*
kernel-module-nls-cp850-5.14.0-yocto-standard-custom		*kernel-module-nls-utf8-5.14.0-yocto-standard-custom*
kernel-module-nouveau-5.14.0-yocto-standard-custom		*kernel-module-parport-5.14.0-yocto-standard-custom*
kernel-module-parport-pc-5.14.0-yocto-standard-custom		*kernel-module-pcs-xpcs-5.14.0-yocto-standard-custom*
kernel-module-pcspkr-5.14.0-yocto-standard-custom		*kernel-module-peak-usb-5.14.0-yocto-standard-custom*
kernel-module-pegasus-5.14.0-yocto-standard-custom		*kernel-module-phylink-5.14.0-yocto-standard-custom*
kernel-module-pinctrl-lynxpoint-5.14.0-yocto-standard-custom		*kernel-module-pinctrl-sunrisepoint-5.14.0-yocto-standard-custom*
kernel-module-pktgen-5.14.0-yocto-standard-custom		*kernel-module-pl2303-5.14.0-yocto-standard-custom*
kernel-module-pnd2-edac-5.14.0-yocto-standard-custom		*kernel-module-portmux-intel-drcfg-5.14.0-yocto-standard-custom*
kernel-module-ppdev-5.14.0-yocto-standard-custom		*kernel-module-processor-thermal-device-5.14.0-yocto-standard-custom*
kernel-module-processor-thermal-device-pci-5.14.0-yocto-standard-custom		*kernel-module-processor-thermal-device-pci-legacy-5.14.0-yocto-standard-custo*
kernel-module-processor-thermal-mbox-5.14.0-yocto-standard-custom		*kernel-module-processor-thermal-rapl-5.14.0-yocto-standard-custom*
kernel-module-processor-thermal-rfim-5.14.0-yocto-standard-custom		*kernel-module-psmouse-5.14.0-yocto-standard-custom*
kernel-module-pwm-lpss-5.14.0-yocto-standard-custom		*kernel-module-pwm-lpss-pci-5.14.0-yocto-standard-custom*
kernel-module-pwm-lpss-platform-5.14.0-yocto-standard-custom		*kernel-module-r8152-5.14.0-yocto-standard-custom*
kernel-module-radeon-5.14.0-yocto-standard-custom		*kernel-module-rfcomm-5.14.0-yocto-standard-custom*
kernel-module-rndis-host-5.14.0-yocto-standard-custom		*kernel-module-roles-5.14.0-yocto-standard-custom*
kernel-module-rtc-rv8803-5.14.0-yocto-standard-custom		*kernel-module-rtl8150-5.14.0-yocto-standard-custom*
kernel-module-selftests-5.14.0-yocto-standard-custom		*kernel-module-serio-5.14.0-yocto-standard-custom*
kernel-module-serpent-generic-5.14.0-yocto-standard-custom		*kernel-module-sil164-5.14.0-yocto-standard-custom*
kernel-module-skx-edac-5.14.0-yocto-standard-custom		*kernel-module-slhc-5.14.0-yocto-standard-custom*
kernel-module-slip-5.14.0-yocto-standard-custom		*kernel-module-smsc-5.14.0-yocto-standard-custom*
kernel-module-smsc75xx-5.14.0-yocto-standard-custom		*kernel-module-smsc95xx-5.14.0-yocto-standard-custom*
kernel-module-snd-ac97-codec-5.14.0-yocto-standard-custom		*kernel-module-snd-ak4113-5.14.0-yocto-standard-custom*
kernel-module-snd-ak4114-5.14.0-yocto-standard-custom		*kernel-module-snd-ak4xxx-adda-5.14.0-yocto-standard-custom*
kernel-module-snd-aloop-5.14.0-yocto-standard-custom		*kernel-module-snd-atiixp-5.14.0-yocto-standard-custom*
kernel-module-snd-atiixp-modem-5.14.0-yocto-standard-custom		*kernel-module-snd-ca0106-5.14.0-yocto-standard-custom*
kernel-module-snd-cmipci-5.14.0-yocto-standard-custom		*kernel-module-snd-cs8427-5.14.0-yocto-standard-custom*
kernel-module-snd-ctl-led-5.14.0-yocto-standard-custom		*kernel-module-snd-ctxfi-5.14.0-yocto-standard-custom*
kernel-module-snd-emu10k1-5.14.0-yocto-standard-custom		*kernel-module-snd-emu10k1-synth-5.14.0-yocto-standard-custom*
kernel-module-snd-emu10k1x-5.14.0-yocto-standard-custom		*kernel-module-snd-emux-synth-5.14.0-yocto-standard-custom*
kernel-module-snd-hda-codec-5.14.0-yocto-standard-custom		*kernel-module-snd-hda-codec-cirrus-5.14.0-yocto-standard-custom*
kernel-module-snd-hda-codec-generic-5.14.0-yocto-standard-custom		*kernel-module-snd-hda-codec-hdmi-5.14.0-yocto-standard-custom*
kernel-module-snd-hda-codec-idt-5.14.0-yocto-standard-custom		*kernel-module-snd-hda-codec-realtek-5.14.0-yocto-standard-custom*
kernel-module-snd-hda-codec-via-5.14.0-yocto-standard-custom		*kernel-module-snd-hda-core-5.14.0-yocto-standard-custom*
kernel-module-snd-hda-ext-core-5.14.0-yocto-standard-custom		*kernel-module-snd-hda-intel-5.14.0-yocto-standard-custom*
kernel-module-snd-hdsp-5.14.0-yocto-standard-custom		*kernel-module-snd-hdspm-5.14.0-yocto-standard-custom*
kernel-module-snd-hrtimer-5.14.0-yocto-standard-custom		*kernel-module-snd-hwdep-5.14.0-yocto-standard-custom*
kernel-module-snd-i2c-5.14.0-yocto-standard-custom		*kernel-module-snd-ice1712-5.14.0-yocto-standard-custom*
kernel-module-snd-ice1724-5.14.0-yocto-standard-custom		*kernel-module-snd-ice17xx-ak4xxx-5.14.0-yocto-standard-custom*
kernel-module-snd-mpu401-uart-5.14.0-yocto-standard-custom		*kernel-module-snd-opl3-lib-5.14.0-yocto-standard-custom*

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
kernel-module-snd-opl3-synth-5.14.0-yocto-standard-custom		*kernel-module-snd-oxygen-5.14.0-yocto-standard-custom*
kernel-module-snd-oxygen-lib-5.14.0-yocto-standard-custom		*kernel-module-snd-pt2258-5.14.0-yocto-standard-custom*
kernel-module-snd-rawmidi-5.14.0-yocto-standard-custom		*kernel-module-snd-rme96-5.14.0-yocto-standard-custom*
kernel-module-snd-rme9652-5.14.0-yocto-standard-custom		*kernel-module-snd-seq-5.14.0-yocto-standard-custom*
kernel-module-snd-seq-device-5.14.0-yocto-standard-custom		*kernel-module-snd-seq-midi-5.14.0-yocto-standard-custom*
kernel-module-snd-seq-midi-emul-5.14.0-yocto-standard-custom		*kernel-module-snd-seq-midi-event-5.14.0-yocto-standard-custom*
kernel-module-snd-seq-virmidi-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-ac97-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-ak4104-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-ak4554-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-ak4613-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-ak4642-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-ak5386-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-alc5623-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs35l32-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs35l33-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs4265-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs4270-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs4271-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs4271-i2c-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs4271-spi-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs42l51-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs42l51-i2c-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs42l52-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs42l56-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs42l73-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs42xx8-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs42xx8-i2c-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-cs4349-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-cs53l30-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-ehl-sgtl5000-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-hdac-hda-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-max98090-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-max98357a-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-pcm1681-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-pcm179x-codec-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-pcm179x-i2c-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-pcm179x-spi-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-pcm3168a-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-pcm3168a-i2c-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-pcm3168a-spi-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-pcm512x-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-pcm512x-i2c-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-pcm512x-spi-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-rl6231-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-rt5616-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-rt5631-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-rt5640-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-rt5645-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-rt5651-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-rt5670-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-sgtl5000-x86-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-simple-card-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-simple-card-utils-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-spdif-rx-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-spdif-tx-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-ssm4567-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-sst-bytcr-rt5640-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-sst-bytcr-rt5651-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-sst-cht-bsw-max98090-ti-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-sst-cht-bsw-rt5645-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-sst-cht-bsw-rt5672-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-tlv320aic31xx-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-tpa6130a2-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-ts3a227e-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8510-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8523-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8580-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8711-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8728-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8731-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8737-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8741-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8750-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8753-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8770-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8776-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8804-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8804-i2c-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8804-spi-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8903-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8960-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8962-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8974-5.14.0-yocto-standard-custom*
kernel-module-snd-soc-wm8978-5.14.0-yocto-standard-custom		*kernel-module-snd-soc-wm8985-5.14.0-yocto-standard-custom*
kernel-module-snd-sof-5.14.0-yocto-standard-custom		*kernel-module-snd-sof-intel-hda-5.14.0-yocto-standard-custom*
kernel-module-snd-sof-intel-hda-common-5.14.0-yocto-standard-custom		*kernel-module-snd-sof-pci-5.14.0-yocto-standard-custom*
kernel-module-snd-sof-pci-intel-tgl-5.14.0-yocto-standard-custom		*kernel-module-snd-sof-xtensa-dsp-5.14.0-yocto-standard-custom*
kernel-module-snd-util-mem-5.14.0-yocto-standard-custom		*kernel-module-snd-virtuoso-5.14.0-yocto-standard-custom*
kernel-module-soc-button-array-5.14.0-yocto-standard-custom		*kernel-module-softdog-5.14.0-yocto-standard-custom*
kernel-module-spi-nor-5.14.0-yocto-standard-custom		*kernel-module-spidev-5.14.0-yocto-standard-custom*
kernel-module-stmmac-5.14.0-yocto-standard-custom		*kernel-module-stmmac-pci-5.14.0-yocto-standard-custom*
kernel-module-stmmac-platform-5.14.0-yocto-standard-custom		*kernel-module-tcpm-5.14.0-yocto-standard-custom*
kernel-module-tcrypt-5.14.0-yocto-standard-custom		*kernel-module-tipc-5.14.0-yocto-standard-custom*
kernel-module-tpm-5.14.0-yocto-standard-custom		*kernel-module-tpm-atmel-5.14.0-yocto-standard-custom*
kernel-module-tpm-crb-5.14.0-yocto-standard-custom		*kernel-module-tpm-infineon-5.14.0-yocto-standard-custom*
kernel-module-tpm-nsc-5.14.0-yocto-standard-custom		*kernel-module-tpm-tis-5.14.0-yocto-standard-custom*
kernel-module-tpm-tis-core-5.14.0-yocto-standard-custom		*kernel-module-tpm-vtpm-proxy-5.14.0-yocto-standard-custom*
kernel-module-ttm-5.14.0-yocto-standard-custom		*kernel-module-tun-5.14.0-yocto-standard-custom*
kernel-module-twofish-common-5.14.0-yocto-standard-custom		*kernel-module-twofish-generic-5.14.0-yocto-standard-custom*
kernel-module-typec-5.14.0-yocto-standard-custom		*kernel-module-typec-wcove-5.14.0-yocto-standard-custom*
kernel-module-u-audio-5.14.0-yocto-standard-custom		*kernel-module-u-ether-5.14.0-yocto-standard-custom*
kernel-module-u-serial-5.14.0-yocto-standard-custom		*kernel-module-udp-tunnel-5.14.0-yocto-standard-custom*
kernel-module-usb-f-acm-5.14.0-yocto-standard-custom		*kernel-module-usb-f-ecm-5.14.0-yocto-standard-custom*
kernel-module-usb-f-ecm-subset-5.14.0-yocto-standard-custom		*kernel-module-usb-f-fs-5.14.0-yocto-standard-custom*
kernel-module-usb-f-hid-5.14.0-yocto-standard-custom		*kernel-module-usb-f-mass-storage-5.14.0-yocto-standard-custom*
kernel-module-usb-f-midi-5.14.0-yocto-standard-custom		*kernel-module-usb-f-ncm-5.14.0-yocto-standard-custom*
kernel-module-usb-f-obex-5.14.0-yocto-standard-custom		*kernel-module-usb-f-printer-5.14.0-yocto-standard-custom*
kernel-module-usb-f-rndis-5.14.0-yocto-standard-custom		*kernel-module-usb-f-serial-5.14.0-yocto-standard-custom*

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
kernel-module-usb-f-ss-lb-5.14.0-yocto-standard-custom	*kernel-module-usb-f-uac2-5.14.0-yocto-standard-custom*	
kernel-module-usblp-5.14.0-yocto-standard-custom	*kernel-module-usbnet-5.14.0-yocto-standard-custom*	
kernel-module-usbserial-5.14.0-yocto-standard-custom	*kernel-module-usbtest-5.14.0-yocto-standard-custom*	
kernel-module-usbtouchscreen-5.14.0-yocto-standard-custom	*kernel-module-uvesafb-5.14.0-yocto-standard-custom*	
kernel-module-video-5.14.0-yocto-standard-custom	*kernel-module-virtio-input-5.14.0-yocto-standard-custom*	
kernel-module-w83627ehf-5.14.0-yocto-standard-custom	*kernel-module-w83627hf-5.14.0-yocto-standard-custom*	
kernel-module-wmi-5.14.0-yocto-standard-custom	*kernel-module-wmi-bmof-5.14.0-yocto-standard-custom*	
kernel-module-wp512-5.14.0-yocto-standard-custom	*kernel-module-x38-edac-5.14.0-yocto-standard-custom*	
kernel-module-x86-pkg-temp-thermal-5.14.0-yocto-standard-custom	*kernel-module-xcbc-5.14.0-yocto-standard-custom*	
kernel-module-xfrm-algo-5.14.0-yocto-standard-custom	*kernel-module-xfrm-user-5.14.0-yocto-standard-custom*	
kernel-module-xhci-plat-hcd-5.14.0-yocto-standard-custom	*kernel-module-zaurus-5.14.0-yocto-standard-custom*	
kernel-modules		
lmsensors	3.6.0	GPLv2+, LGPLv2.1+
lmsensors-libsensors		
lmsensors-config	1.0	MIT-X
lmsensors-config-libsensors		
lrzsz	0.12.20	GPLv2+
lrzsz		
lvm2	2.03.11	GPLv2, LGPLv2.1
libdevmapper	*lvm2*	
lvm2-scripts	*lvm2-udevrules*	
lximage-qt	0.17.0	GPLv2
lximage-qt	*lximage-qt-locale-en-gb*	
lxmenu-data	0.1.5	LGPL-2.1
lxmenu-data		
lxqt-about	0.17.0	LGPL-2.1
lxqt-about	*lxqt-about-locale-en-gb*	
lxqt-admin	0.17.0	LGPL-2.1
lxqt-admin	*lxqt-admin-locale-en-gb*	
lxqt-archiver	0.4.0	GPLv2
lxqt-archiver	*lxqt-archiver-locale-en-gb*	
lxqt-build-tools	0.9.0	BSD-3-Clause
lxqt-build-tools		
lxqt-config	0.17.1	LGPL-2.1
lxqt-config	*lxqt-config-locale-en-gb*	
lxqt-globalkeys	0.17.0	LGPL-2.1
lxqt-globalkeys	*lxqt-globalkeys-locale-en-gb*	
lxqt-notificationd	0.17.0	LGPL-2.1
lxqt-notificationd	*lxqt-notificationd-locale-en-gb*	
lxqt-openssh-askpass	0.17.0	LGPL-2.1
lxqt-openssh-askpass	*lxqt-openssh-askpass-locale-en-gb*	
lxqt-panel	0.17.1	LGPL-2.1
lxqt-panel	*lxqt-panel-locale-en-gb*	
lxqt-policykit	0.17.0	LGPL-2.1
lxqt-policykit		
lxqt-powermanagement	0.17.1	LGPL-2.1
lxqt-powermanagement		
lxqt-qtplugin	0.17.0	LGPL-2.1
lxqt-qtplugin		
lxqt-runner	0.17.0	LGPL-2.1
lxqt-runner		
lxqt-session	0.17.1	LGPL-2.1
lxqt-session		
lxqt-sudo	0.17.0	LGPL-2.1
lxqt-sudo		
lxqt-themes	0.17.0	LGPL-2.1
lxqt-themes		
lxqt-world	1.0	MIT

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
lxqt-world		
mcelog *mcelog*	175	GPLv2
mem-edit *mem-edit*	git	GPLv2
menu-cache *menu-cache*	1.1.0	LGPLv2.1+
mesa *libegl-mesa* *libgl-mesa* *libgles1-mesa* *mesa-megadriver*	21.0.3 *libgbm* *libglapi* *libgles2-mesa* *mesa-vulkan-drivers*	MIT
mesa-demos *mesa-demos*	8.4.0	MIT, PD
mini-x-session *mini-x-session*	0.1	GPLv2
minicom *minicom*	2.7.1	GPLv2+
mmc-utils *mmc-utils*	0.1	GPLv2
mobile-broadband-provider-info *mobile-broadband-provider-info*	20201225	PD
modutils-initscripts *modutils-initscripts*	1.0	PD
mozjs *libmozjs*	60.9.0	MPL-2.0
mpeg2dec *libmpeg2*	0.5.1	GPLv2+
mpfr *mpfr*	4.1.0	LGPLv3+
mpg123 *mpg123*	1.26.4	LGPLv2.1
msc-bug-report *msc-bug-report*	1.0	GPLv2
msc-companion-tools *msc-companion-tools*	git	GPLv2
msc-completion *msc-completion*	1.0	MIT
msc-cpufreq *msc-cpufreq*	1.0	GPLv2
msc-init-script-early *msc-init-script-early*	1.0	MIT
msc-init-script-late *msc-init-script-late*	1.0	MIT
msc-ldk-benchmark *msc-ldk-benchmark*	git	GPLv2
msc-ldk-exhibition-safe *msc-ldk-exhibition-safe*	git	GPLv2
msc-ldk-verification *msc-ldk-verification*	git	GPLv2
msc-ldk-verification-apps *msc-ldk-verification-apps*	git	GPLv2
msc-linux-scripts *msc-linux-scripts*	git	GPLv2
msc-lxqt-config *msc-lxqt-config*	1.0	MIT

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
msc-screensaver *msc-screensaver*	git	GPLv2
msc-set-x-screen-resolution *msc-set-x-screen-resolution*	1.0	MIT
msc-wallpaper *msc-wallpaper*	git	GPLv2
mscio-cmd *mscio-cmd*	git	LGPLv2.1
mscio-drivers *kernel-module-devreg-5.14.0-yocto-standard-custom* *kernel-module-eapi-ec-bl-5.14.0-yocto-standard-custom* *kernel-module-eapi-ec-running-time-5.14.0-yocto-standard-custom* *kernel-module-eapi-nvram-5.14.0-yocto-standard-custom* *kernel-module-leds-gpio-acpi-5.14.0-yocto-standard-custom* *mscio-drivers*	git *kernel-module-eapi-ec-5.14.0-yocto-standard-custom* *kernel-module-eapi-ec-hwm-5.14.0-yocto-standard-custom* *kernel-module-eapi-ec-wdt-5.14.0-yocto-standard-custom* *kernel-module-i2c-ids-acpi-5.14.0-yocto-standard-custom* *kernel-module-user-gpios-5.14.0-yocto-standard-custom*	GPLv2
mscio-lib *mscio-lib*	git	LGPLv2.1
mscio-monitor *mscio-monitor*	git	GPLv2
mscio-setup *mscio-setup*	git	LGPLv2.1
mtd-utils *mtd-utils*	2.1.2	GPLv2+
mtdev *mtdev*	1.1.6	MIT
muparser *muparser*	2.3.2	BSD-2-Clause
nano *nano*	5.6	GPLv3
ncurses *ncurses-libformw* *ncurses-libncurses* *ncurses-libpanelw* *ncurses-terminfo-base*	6.2 *ncurses-libmenuw* *ncurses-libncursesw* *ncurses-libtinfo*	MIT
ndctl *ndctl*	v69	GPLv2+
neard *neard*	0.16	GPLv2
netbase *netbase*	6.2	GPLv2
netpipe *netpipe*	3.7.2	GPLv1
nettle *nettle*	3.7.2	LGPLv3+, GPLv2+
nfs-utils *nfs-utils-client*	2.5.3 *nfs-utils-mount*	MIT, GPLv2+, BSD
npth *npth*	1.6	LGPLv2+
nspr *nspr*	4.29	GPL-2.0, MPL-2.0, LGPL-2.1
nss *nss*	3.64	(MPL-2.0 & LGPL-2.1+ & MIT), (MPL-2.0 & GPL-2.0+ & MIT), (MPL-2.0 & MIT)
ntp *ntp* *ntpdate*	4.2.8p15 *ntp-tickadj*	NTP

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
obconf-qt	0.16.1	LGPL-2.1
obconf-qt		
ofono	1.31	GPLv2
ofono		
openbox	3.6.1	GPLv2+
openbox	*openbox-config*	
openbox-core	*openbox-theme-artwiz-boxed*	
openbox-theme-bear2	*openbox-theme-clearlooks*	
openbox-theme-clearlooks-3.4	*openbox-theme-clearlooks-olive*	
openbox-theme-mikachu	*openbox-theme-natura*	
openbox-theme-onyx	*openbox-theme-onyx-citrus*	
openbox-theme-orang	*openbox-theme-syscrash*	
openjpeg	2.4.0	BSD-2-Clause
openjpeg		
openssh	8.5p1	BSD-2-Clause, BSD-3-Clause, BSD-4-Clause, BSD, ISC, MIT
openssh	*openssh-keygen*	
openssh-scp	*openssh-ssh*	
openssh-sshd		
openssl	1.1.1k	openssl
libcrypto	*libssl*	
openssl	*openssl-bin*	
openssl-conf		
opkg-utils	0.4.3	GPLv2+
update-alternatives-opkg		
orc	0.4.32	BSD-2-Clause, BSD-3-Clause
liborc-0.4		
otter-browser	1.0.02	GPLv3
otter-browser		
oxygen-icons5	5.83.0	LGPL-3.0
oxygen-icons5		
packagegroup-base	1.0	MIT
packagegroup-base	*packagegroup-base-3g*	
packagegroup-base-acpi	*packagegroup-base-alsa*	
packagegroup-base-bluetooth	*packagegroup-base-ext2*	
packagegroup-base-extended	*packagegroup-base-ipv6*	
packagegroup-base-keyboard	*packagegroup-base-nfc*	
packagegroup-base-nfs	*packagegroup-base-pci*	
packagegroup-base-usbgadget	*packagegroup-base-usbhost*	
packagegroup-base-wifi	*packagegroup-base-zeroconf*	
packagegroup-distro-base	*packagegroup-machine-base*	
packagegroup-core-boot	1.0	MIT
packagegroup-core-boot		
packagegroup-core-ssh-openssh	1.0	MIT
packagegroup-core-ssh-openssh		
packagegroup-core-x11	1.0	MIT
packagegroup-core-x11-utils		
packagegroup-core-x11-base	1.0	MIT
packagegroup-core-x11-base		
packagegroup-core-x11-xserver	1.0	MIT
packagegroup-core-x11-xserver		
packagegroup-lxqt-base	1.0	MIT
packagegroup-lxqt-base		
packagegroup-msc-ldk-core	1.0	MIT
packagegroup-msc-ldk-core		

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
packagegroup-msc-lxqt	1.0	MIT
packagegroup-msc-lxqt	*packagegroup-msc-lxqt-apps*	
packagegroup-msc-lxqt-base	*packagegroup-msc-lxqt-games*	
packagegroup-mscio	1.0	MIT
packagegroup-mscio		
packagegroup-tools-in-x	1.0	MIT
packagegroup-tools-in-x		
pango	1.48.2	LGPLv2.0+
pango		
parole	4.16.0	GPLv2
parole	*parole-locale-en-gb*	
parted	3.4	GPLv3+
parted		
pavucontrol-qt	0.17.0	GPLv2
pavucontrol-qt		
pci2uio	git	GPLv2
kernel-module-pci2uio-5.14.0-yocto-standard-custom	*pci2uio*	
pciutils	3.7.0	GPLv2+
libpci	*pciutils*	
pciutils-ids		
pcmanfm-qt	0.17.0	GPLv2
pcmanfm-qt	*pcmanfm-qt-locale-en-gb*	
perf	1.0	GPLv2
perf		
perl	5.32.1	Artistic-1.0, GPL-1.0+
perl	*perl-module-config-heavy*	
picom	8.2	MPL-2.0, MIT
picom		
pinentry	1.1.1	GPLv2
pinentry		
pixman	0.40.0	MIT, MIT-style, PD
pixman		
pm-utils	1.4.1	GPLv2
pm-utils		
polkit	0.116	LGPLv2+
polkit		
polkit-group-rule-datetime	1.0	MIT
polkit-group-rule-datetime		
polkit-qt-1	0.113.0	LGPL-2.1
polkit-qt-1		
popt	1.18	MIT
popt		
powertop	2.13	GPLv2
powertop	*powertop-locale-en-gb*	
powertop-locale-en-us		
procps	3.3.17	GPLv2+, LGPLv2+
procps	*procps-lib*	
procps-ps	*procps-sysctl*	
proxy-config	1.0	MIT
proxy-config		
pulseaudio	14.2	LGPLv2.1+, MIT, BSD-3-Clause
libpulse	*libpulse-mainloop-glib*	
libpulse-simple	*libpulsecommon*	
libpulsecore	*pulseaudio-lib-alsa-util*	
pulseaudio-lib-protocol-native	*pulseaudio-module-alsa-card*	
pulseaudio-module-alsa-sink	*pulseaudio-module-alsa-source*	

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
pulseaudio-module-always-sink	*pulseaudio-module-augment-properties*	
pulseaudio-module-card-restore	*pulseaudio-module-console-kit*	
pulseaudio-module-default-device-restore	*pulseaudio-module-detect*	
pulseaudio-module-device-manager	*pulseaudio-module-device-restore*	
pulseaudio-module-filter-apply	*pulseaudio-module-filter-heuristics*	
pulseaudio-module-intended-roles	*pulseaudio-module-native-protocol-unix*	
pulseaudio-module-null-sink	*pulseaudio-module-position-event-sounds*	
pulseaudio-module-rescue-streams	*pulseaudio-module-role-cork*	
pulseaudio-module-stream-restore	*pulseaudio-module-suspend-on-idle*	
pulseaudio-module-switch-on-port-available	*pulseaudio-module-udev-detect*	
pulseaudio-module-x11-cork-request	*pulseaudio-module-x11-publish*	
pulseaudio-module-x11-xsmp	*pulseaudio-server*	
python3	3.9.5	PSFv2
libpython3	*python3-2to3*	
python3-asyncio	*python3-audio*	
python3-codecs	*python3-compile*	
python3-compression	*python3-core*	
python3-crypt	*python3-ctypes*	
python3-curses	*python3-datetime*	
python3-db	*python3-debugger*	
python3-difflib	*python3-distutils*	
python3-doctest	*python3-email*	
python3-fcntl	*python3-html*	
python3-idle	*python3-image*	
python3-io	*python3-json*	
python3-logging	*python3-mailbox*	
python3-math	*python3-mime*	
python3-misc	*python3-mmap*	
python3-modules	*python3-multiprocessing*	
python3-netclient	*python3-netserver*	
python3-numbers	*python3-pickle*	
python3-pkgutil	*python3-plistlib*	
python3-pprint	*python3-profile*	
python3-pydoc	*python3-resource*	
python3-shell	*python3-smtpd*	
python3-sqlite3	*python3-stringold*	
python3-syslog	*python3-terminal*	
python3-threading	*python3-tkinter*	
python3-unittest	*python3-unixadmin*	
python3-venv	*python3-xml*	
python3-xmlrpc		
python3-async	0.6.2	BSD-3-Clause
python3-async		
python3-dbus	1.2.16	MIT
python3-dbus		
python3-ewmh	0.1.5	LGPL-3.0
python3-ewmh		
python3-git	3.1.14	BSD-3-Clause
python3-git		
python3-gitdb	4.0.5	BSD-3-Clause
python3-gitdb		
python3-iniparse	0.4	MIT, PSF
python3-iniparse		
python3-pillow	8.2.0	MIT
python3-pillow		
python3-pycairo	1.20.0	LGPLv2.1, MPLv1.1
python3-pycairo		
python3-pygobject	3.38.0	LGPLv2.1
python3-pygobject		
python3-six	1.15.0	MIT
python3-six		
python3-smmap	4.0.0	BSD-3-Clause
python3-smmap		

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
python3-xlib *python3-xlib*	0.25	LGPL-2.1
pyxdg *pyxdg*	0.26	LGPLv2
qps *qps*	2.3.0	GPLv2
qtbase *qtbase* *qtbase-qmlplugins*	5.15.2 *qtbase-plugins*	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtdeclarative *qtdeclarative* *qtdeclarative-qmlplugins*	5.15.2 *qtdeclarative-plugins*	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qterminal *qterminal*	0.17.0	GPLv2
qtermwidget *qtermwidget*	0.17.0	GPLv2
qtimageformats *qtimageformats-plugins*	5.15.2 	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtlocation *qtlocation* *qtlocation-qmlplugins*	5.15.2 *qtlocation-plugins*	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), Apache-2.0, MIT, openssl, BSL-1.0, GFDL-1.3, BSD
qtmultimedia *qtmultimedia* *qtmultimedia-qmlplugins*	5.15.2 *qtmultimedia-plugins*	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
qtsensors	5.15.2	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtsensors *qtsensors-qmlplugins*	*qtsensors-plugins*	
qtsvg	5.15.2	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtsvg *qtsvg-qmlplugins*	*qtsvg-plugins*	
qtwayland	5.15.2	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtwayland *qtwayland-qmlplugins*	*qtwayland-plugins*	
qtwebchannel	5.15.2	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtwebchannel *qtwebchannel-qmlplugins*	*qtwebchannel-plugins*	
qtwebkit	5.15.2	BSD, LGPLv2+, GPL-2.0
qtwebkit *qtwebkit-qmlplugins*	*qtwebkit-plugins*	
qtx11extras	5.15.2	(GPL-2.0+ | LGPL-3.0), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0), GFDL-1.3, BSD, The-Qt-Company-Commercial
qtx11extras *qtx11extras-qmlplugins*	*qtx11extras-plugins*	
qtxmlpatterns	5.15.2	(GPL-2.0+ | LGPL-3.0 | The-Qt-Company-Commercial), (GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 | The-Qt-Company-Commercial), GFDL-1.3, BSD
qtxmlpatterns *qtxmlpatterns-qmlplugins*	*qtxmlpatterns-plugins*	
readline *readline*	8.1	GPLv3+
rgb *rgb*	1.0.6	MIT-X
rng-tools *rng-tools*	6.11	GPLv2

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
rpcbind *rpcbind*	1.2.5	BSD-3-Clause
rpm *python3-rpm* *rpm-build*	4.16.1.3 *rpm* *rpm-sign*	GPL-2.0
run-postinsts *run-postinsts*	1.0	MIT
rxvt-unicode *rxvt-unicode*	9.22	GPLv3
sbc *sbc*	1.5	LGPLv2.1+
sdbus-c++ *sdbus-c++*	1.1.0	LGPLv2.1
shadow *shadow*	4.8.1 *shadow-base*	BSD, Artistic-1.0
shadow-securetty *shadow-securetty*	4.6	MIT
shared-mime-info *shared-mime-info* *shared-mime-info-locale-en-gb*	2.1 *shared-mime-info-data*	GPLv2
slang *slang*	2.3.2	GPLv2
smartmontools *smartmontools-ctl*	6.3	GPLv2
solid *solid*	5.83.0 *solid-locale-en-gb*	BSD-3-Clause, LGPL-2.0, LGPL-2.1, LGPL-2.1+, LGPL-3.0
speex *speex*	1.2.0	BSD-3-Clause
speexdsp *speexdsp*	1.2.0	BSD-3-Clause
spi-register *spi-register*	git	GPLv2
sqlite3 *libsqlite3*	3.35.0	PD
strace *strace*	5.11	LGPL-2.1+, GPL-2+
stress-ng *stress-ng*	0.12.05	GPLv2
sudo *sudo* *sudo-sudo*	1.9.6p1 *sudo-lib*	ISC, BSD, Zlib
superiotool *superiotool*	1.0	GPLv2
sysfsutils *libsysfs*	2.1.0	LGPLv2.1
system-test-controller-config *system-test-controller-config*	1.0	MIT
system-test-controller-v2 *system-test-controller-v2*	git	GPLv2
systemd-boot *systemd-boot*	247.6	GPLv2, LGPLv2.1
systemd-bootconf *systemd-bootconf*	1.00	MIT

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
sysvinit	2.99	GPLv2+
sysvinit	*sysvinit-pidof*	
sysvinit-inittab	2.88dsf	GPLv2
sysvinit-inittab		
taglib	1.12	LGPLv2.1, MPL-1.1
taglib	*taglib-c*	
tcp-wrappers	7.6	BSD-1-Clause
libwrap		
tcpdump	4.99.0	BSD-3-Clause
tcpdump		
tiff	4.2.0	BSD-2-Clause
tiff		
tiny-shell	git	GPLv2
tiny-shell		
turbostat	3.4	GPLv2
turbostat		
tzdata	2021a	PD, BSD, BSD-3-Clause
tzdata	*tzdata-africa*	
tzdata-americas	*tzdata-antarctica*	
tzdata-arctic	*tzdata-asia*	
tzdata-atlantic	*tzdata-australia*	
tzdata-core	*tzdata-europe*	
tzdata-misc	*tzdata-pacific*	
tzdata-posix	*tzdata-right*	
udev-extraconf	1.1	MIT
udev-extraconf		
udisks2	2.9.2	GPLv2+, LGPLv2+
udisks2	*udisks2-libs*	
udisks2-locale-en-gb		
update-flash	git	GPLv2
update-flash		
update-rc.d	0.8	GPLv2+
update-rc.d		
usbutils	013	GPLv2+
usbutils		
useradd-msc	1.0	MIT
useradd-msc		
util-linux	2.36.2	GPLv2+, LGPLv2.1+, BSD-3-Clause, BSD-4-Clause
util-linux-blkid	*util-linux-dmesg*	
util-linux-findmnt	*util-linux-fstrim*	
util-linux-libblkid	*util-linux-libmount*	
util-linux-libsmartcols	*util-linux-lsblk*	
util-linux-lscpu	*util-linux-mcookie*	
util-linux-mount	*util-linux-sulogin*	
util-linux-swapoff	*util-linux-swapon*	
util-linux-swaponoff	*util-linux-umount*	
util-linux-libuuid	2.36.2	BSD-3-Clause
util-linux-libuuid		
volume-key	0.3.12	GPLv2
volume-key	*volume-key-locale-en-gb*	
vulkan-loader	1.2.170.0	Apache-2.0
vulkan-loader		
wayland	1.19.0	MIT
wayland		

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
wireless-regdb *wireless-regdb-static*	2021.04.21	ISC
wpa-supplicant *wpa-supplicant* *wpa-supplicant-passphrase*	2.9 *wpa-supplicant-cli*	BSD-3-Clause
xauth *xauth*	1.1	MIT-X
xcb-util *xcb-util*	0.4.0	MIT
xcb-util-image *xcb-util-image*	0.4.0	MIT
xcb-util-keysyms *xcb-util-keysyms*	0.4.0	MIT
xcb-util-renderutil *xcb-util-renderutil*	0.3.9	MIT
xcb-util-wm *xcb-util-wm*	0.4.1	MIT
xdg-user-dirs *xdg-user-dirs*	0.17	GPLv2
xdpyinfo *xdpyinfo*	1.3.2	MIT-X
xf86-input-libinput *xf86-input-libinput*	0.30.0	MIT-X
xf86-video-ast *xf86-video-ast*	1.1.5	MIT-X
xf86-video-fbdev *xf86-video-fbdev*	0.5.0	MIT-X
xf86-video-intel *xf86-video-intel*	2.99.917	MIT-X
xf86-video-vesa *xf86-video-vesa*	2.5.0	MIT-X
xfconf *xfconf*	4.16.0 *xfconf-locale-en-gb*	GPLv2
xhost *xhost*	1.0.8	MIT-X
xinit *xinit*	1.4.1	MIT-X
xinput *xinput*	1.6.3	MIT-X
xinput-calibrator *xinput-calibrator*	0.7.5	MIT-X
xkbcomp *xkbcomp*	1.4.4	MIT-X
xkeyboard-config *xkeyboard-config*	2.32 *xkeyboard-config-locale-en-gb*	MIT, MIT-style
xmodmap *xmodmap*	1.0.10	MIT
xrandr *xrandr*	1.5.1	MIT
xserver-nodm-init *xserver-nodm-init*	3.0	GPLv2
xserver-xf86-config *xserver-xf86-config*	0.1	MIT-X
xserver-xorg *xf86-video-modesetting* *xserver-xorg-extension-glx*	1.20.10 *xserver-xorg* *xserver-xorg-module-libint10*	MIT-X
xset	1.2.4	MIT

Table L.2. – License overview listing (continued)

Recipe	Version	Licenses
xset		
xssstate *xssstate*	1.1	MIT
xterm *xterm*	367	MIT-X
xvinfo *xvinfo*	1.1.4	MIT-X
xz *liblzma*	5.2.5	PD
zip *zip*	3.0	BSD-3-Clause
zlib-intel *zlib-intel*	1.2.11.1.jtkv6.3	Zlib
zsh *zsh*	5.4.2	zsh
zsh-config *zsh-config*	1.0	MIT