Intelligent Solutions

msc technologies
an Avnet Company

Engineering Leadership

MSC-LDK Manual
BSP C984-baytrail V1.5.0

2019-12-12

Copyright Notice

Trademarks

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

1. History **9**
- 1.1. Document . . . 9
- 1.2. MSC-LDK . . . 10
- 1.3. BSP C984-baytrail . . . 12

2. Introduction **13**
- 2.1. Scope . . . 13
- 2.2. Out Of Scope . . . 13
- 2.3. Features . . . 13
- 2.4. Conventions . . . 13
- 2.5. Product Support . . . 14

3. Getting Started **15**
- 3.1. Requirements . . . 15
- 3.2. Registration On The MSC Git Server . . . 15
  - 3.2.1. Creating An SSH Key . . . 15
  - 3.2.2. Configuring For HTTP proxies . . . 16

4. Using MSC-LDK **17**
- 4.1. Yocto/MSC-LDK Terms . . . 17
- 4.2. Setup Optional Docker Container . . . 17
- 4.3. Installation Of MSC-LDK . . . 18
- 4.4. Directory Layout . . . 18
  - 4.4.1. MSC-LDK Layers . . . 19
- 4.5. Installation Of MSC-LDK BSPs . . . 21
  - 4.5.1. 64bit Target Linux . . . 21
- 4.6. BSP Aliases . . . 21
- 4.7. Reduced Cleanup Time . . . 21
- 4.8. Building Images . . . 21
- 4.9. Traceable And Reproducable Images . . . 22
- 4.10. Building Documentation . . . 23
- 4.11. Image Types . . . 23
  - 4.11.1. Headless . . . 23
  - 4.11.2. LXQt . . . 23
  - 4.11.3. LXQt SDK . . . 23
  - 4.11.4. Sato . . . 24
  - 4.11.5. Sato SDK . . . 24
- 4.12. Deploying Images To The Hardware . . . 24
  - 4.12.1. USB . . . 24
  - 4.12.2. SD Card . . . 25
  - 4.12.3. SATA . . . 25
  - 4.12.4. Other . . . 25
- 4.13. Login . . . 25
- 4.14. Enabled Services . . . 26
- 4.15. Enhancing The Images . . . 26
  - 4.15.1. Adding Other Packages . . . 26
  - 4.15.2. Adding Own Layers . . . 26
  - 4.15.3. Network Configuration . . . 27
  - 4.15.4. Time Zone Setup . . . 27
- 4.16. Using LXQt . . . 28
- 4.17. ConnMan Configuration . . . 28
  - 4.17.1. Proxy Configuration For The otter-browser . . . 31
- 4.18. Using Sato . . . 32

4.19. Using The SDK Images . . . 32
4.19.1. Setting Up The Development Host . . . 32
4.19.2. Setting Up The Target . . . 32
4.19.3. Building hello_world Manually . . . 33
4.19.4. Building hello_world With CMake . . . 33
4.19.5. Building Qt Based hello_world (Only In msc-image-lxqt-sdk) . . . 33
4.20. German Keyboard Layouts . . . 34
4.21. Bug Reporting . . . 34
4.22. Hotfixes And Updating MSC-LDK . . . 36

5. Package List **37**
5.1. eapi-so . . . 37
5.2. mscio . . . 37
5.3. system-test-controller . . . 37

A. Glossary **38**

B. Bootloader Configuration **39**
B.1. RAMDISK . . . 39

C. X11/SATO Configuration **40**
C.1. Screens . . . 40

D. Graphic Demos **41**
D.1. Graphic Driver Status . . . 41
D.1.1. igt-gpu-tools . . . 41
D.2. Video Acceleration . . . 41
D.2.1. GStreamer . . . 41

E. Supported Periphery **43**
E.1. eMMC . . . 44
E.2. GPIOs . . . 44
E.3. Audio . . . 44
E.3.1. PulseAudio . . . 44
E.3.2. ALSA . . . 45
E.4. I2C . . . 46
E.5. USB Device Support . . . 46
E.5.1. Setting Up USB Device Functionality . . . 46
E.5.2. Testing Connection . . . 46
E.5.3. Setting Up Serial Gadget . . . 46
E.5.4. Setting Up Ethernet Gadget . . . 46
E.5.5. Setting Up Mass/File Storage Gadget . . . 47
E.6. UART . . . 47
E.7. SD/MMC . . . 47
E.8. SPI . . . 48
E.9. Powermanagement . . . 48
E.9.1. Supported Capabilities . . . 48
E.9.2. Suspend To RAM / S3 . . . 48
E.9.3. Wake-up Sources . . . 48
E.10. Graphic Output . . . 49
E.10.1. Dual Head Support . . . 49
E.11. ACPI . . . 50

F. MSC-IO **51**
F.1. What Is MSC-IO? . . . 51
F.2. mscio-setup . . . 51
F.3. mscio-drivers . . . 51
F.3.1. Backlight . . . 51
F.3.2. Hardware Monitoring . . . 52
F.3.3. Running Time . . . 52
F.3.4. Watchdog . . . 52
F.3.5. User GPIOs . . . 53
F.3.6. NVRAM . . . 54

F.3.7. Board Information . . . 54
F.3.8. I2C Bus Identification . . . 55
F.4. mscio-monitor . . . 55
F.5. User Library . . . 56

G. Troubleshooting **58**

H. Tips And Tricks **59**
H.1. Using A Standard Browser . . . 59
H.1.1. Using The google-chrome Browser . . . 59
H.2. Timekeeping . . . 59
H.2.1. htpdate . . . 59

I. Security **61**
I.1. Introduction . . . 61
I.2. Layer meta-security-isafw . . . 61
I.3. Setting Up A Project . . . 61

J. Real-time **62**
J.1. What Is Real-time . . . 62
J.2. Setup A BSP For Real-time Support . . . 62
J.3. Analysis Of Real-time Behavior . . . 62
J.3.1. Yocto Tracing And Profiling . . . 62
J.3.2. Application List With Real-Time Priorities . . . 66
J.4. Latency . . . 66
J.4.1. cyclictest . . . 67
J.4.2. Hardware Latency . . . 67
J.4.3. Reducing Latency . . . 67
J.4.4. Summary . . . 69

K. Power Analysis **70**
K.1. perf . . . 70
K.1.1. Analyzing CPU sleep states . . . 70
K.1.2. Analyzing instruction and branches counter . . . 71
K.2. cpupower . . . 71
K.3. powertop . . . 71

L. Known Issues **73**
L.1. LVDS Displays - REQ-00616 . . . 73
L.2. kexec . . . 73
L.3. Installation Of 64bit Images With .hddimg - REQ-00645 . . . 73
L.4. HDMI Audio . . . 73
L.5. Standby And LVDS Display - REQ-00720 . . . 73
L.6. Standby And USB Device Support - REQ-00723 . . . 73
L.7. USB Peripheral Drivers Must Be Reloaded Before Switching Gadget Driver - REQ-00890 73

M. Links **74**

N. License Overview **75**

List of Figures

4.1. MSC-LDK Layers . . . 19
4.2. LXQt - Mainwindow . . . 28
4.3. Start ConnMan Property Editor . . . 28
4.4. View ConnMan Properties . . . 29
4.5. Manual IP Assignment . . . 29
4.6. Otter Browser - Proxy Setup . . . 31
4.7. Sato - Mainwindow . . . 32
4.8. Sato - Groups . . . 32
4.9. Sato - Selecting Running Apps . . . 32
4.10. Bug Reporting - Start . . . 34
4.11. Bug Reporting - Start . . . 35
4.12. Bug Reporting - View . . . 35
4.13. Bug Reporting - Create Zip . . . 35
4.14. Bug Reporting - Save Zip . . . 36

F.1. mscio-monitor . . . 56

J.1. kernelshark - systemcalls . . . 63
J.2. kernelshark - events . . . 64
J.3. kernelshark - IRQs . . . 65

K.1. perf timechart output.svg . . . 71
K.2. powertop overview . . . 72

List of Tables

4.1. MSC-LDK Directory Layout . . . 18
4.1. MSC-LDK Directory Layout (continued) . . . 19
4.2. MSC-LDK Layer Description . . . 19
4.2. MSC-LDK Layer Description (continued) . . . 20
4.3. User Accounts . . . 26

D.1. Intel IGT GPU Tools . . . 41

E.1. Supported periphery . . . 43
E.2. GPIOs . . . 44

F.1. mscio-drivers . . . 51

G.1. Errors . . . 58

K.1. CPU core states . . . 70

N.1. list of licenses used in this build . . . 75
N.2. License overview listing . . . 76
N.2. License overview listing (continued) . . . 77
N.2. License overview listing (continued) . . . 78
N.2. License overview listing (continued) . . . 79
N.2. License overview listing (continued) . . . 80
N.2. License overview listing (continued) . . . 81
N.2. License overview listing (continued) . . . 82
N.2. License overview listing (continued) . . . 83
N.2. License overview listing (continued) . . . 84
N.2. License overview listing (continued) . . . 85
N.2. License overview listing (continued) . . . 86
N.2. License overview listing (continued) . . . 87
N.2. License overview listing (continued) . . . 88
N.2. License overview listing (continued) . . . 89
N.2. License overview listing (continued) . . . 90
N.2. License overview listing (continued) . . . 91
N.2. License overview listing (continued) . . . 92
N.2. License overview listing (continued) . . . 93
N.2. License overview listing (continued) . . . 94
N.2. License overview listing (continued) . . . 95
N.2. License overview listing (continued) . . . 96
N.2. License overview listing (continued) . . . 97
N.2. License overview listing (continued) . . . 98
N.2. License overview listing (continued) . . . 99
N.2. License overview listing (continued) . . . 100
N.2. License overview listing (continued) . . . 101
N.2. License overview listing (continued) . . . 102

Listings

msc-ldk-doc/bsp/yocto-output.txt 22
4.1. /etc/exports 32
4.2. hello_world.c 33
4.3. CMakeLists.txt 33
4.4. hello_world.cpp 33
4.5. hello_world.pro 34

B.1. EFI/BOOT/grub.cfg 39

F.1. example.cpp 56

K.1. perf timechart record 70

1. History

1.1. Document

Revision	Changes
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” • Added section “Hotfixed and Updating MSC-LDK” • Updated section “Installation of MSC-LDK”
1.0.0	• Added section “Using LXQt” • Added section “Using the SDK images” • Added section “Bug Reporting”

1.2. MSC-LDK

Revision	Changes
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
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.

Revision	Changes
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
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 C984-baytrail

Revision	Changes
1.5.0	• Updated to kernel 4.19 • Using igt-gpu-tools instead of intel-gpu-tools
1.4.0	• Updated to kernel 4.14
1.3.0	• Updated to kernel 4.12 • Added kernel option `sdhci_pci.disable_ddr50` • Removed kernel option `acpi_enforce_resources=lax`
1.2.0	• Updated to kernel 4.8 • Graphic driver i915 is loaded as a module • SD/MMC frequency can be limited
1.0.0	• Updated to kernel 3.19
0.4.0	• Added EDAC driver • I2C speed reduced to 100kHz (from 400kHz) • Updated to kernel 3.17
0.3.0	• Updated Known Issues • Updated to kernel 3.14 • Some kernel configuration tweaks.
0.2.0	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 2.7 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 16.04 (LTS), but other distributions up to Ubuntu 19.04 may also work.
- 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 v1.5.0
```

```
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
```

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 v1.5.0
```

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/C984*`	BSP build directory.
`doc`	This contains all the generated Yocto documentation.

Table 4.1. – MSC-LDK Directory Layout (continued)

Directory	Contents
	(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/C984/*.git`	BSP specific layers. (will be created in chapter 4.5)
`sources/meta-freescale.git/`	Freescale specific layers, only on Freescale 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

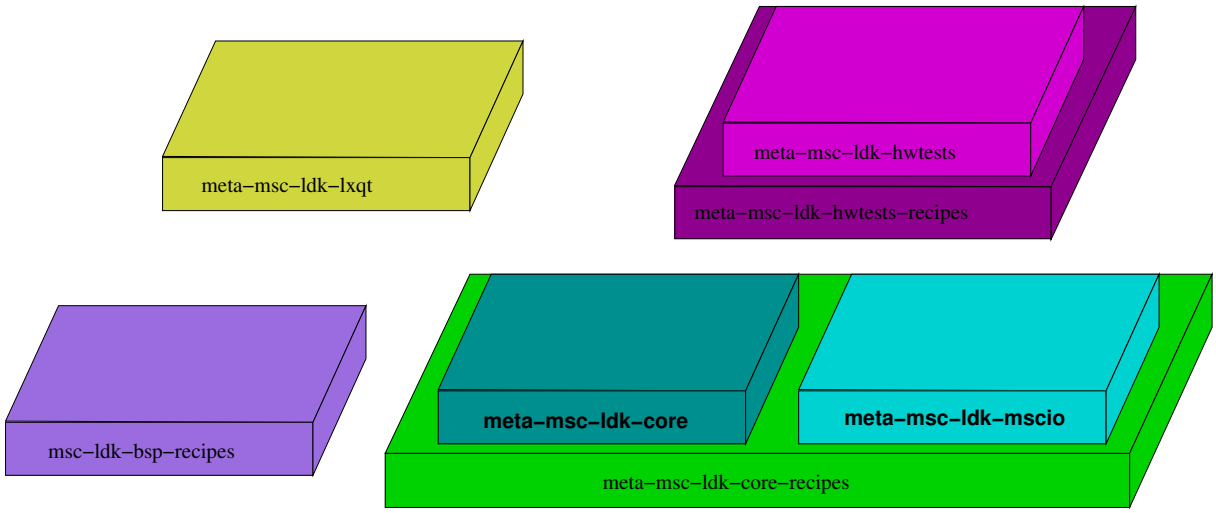


Table 4.2. – MSC-LDK Layer Description

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`.

Table 4.2. – MSC-LDK Layer Description (continued)

Layer	Description
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=C984
```

4.6. BSP Aliases

`setup.py` supports one of the following aliases for the option `--bsp=intelcombined,C984,Baytrail,Q7-BT,C6C-BT`

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=C984
user@devhost:msc-ldk$ cd build/C984
user@devhost:C984$ 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
```

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         = "1.28.0"
BUILD_SYS          = "x86_64-linux"
NATIVELSBSTRING    = "CentOS-Linux-7.1.1503"
TARGET_SYS         = "x86_64-poky-linux"
MACHINE            = "intel-corei7-64-cavs-hda"
DISTRO             = "poky"
DISTRO_VERSION     = "2.0.1"
TUNE_FEATURES      = "m64 corei7"
TARGET_FPU         = ""
meta-yocto
meta-yocto-bsp     = "jethro-msc:c94fbbfb8c10f77022e18e492b41d6ab809012d0"
meta               = "master:bac6d827c15519becb945941374b45d69b8b23a5"
meta               = "jethro-msc:c94fbbfb8c10f77022e18e492b41d6ab809012d0"
meta-oe
meta-networking
meta-python        = "jethro-msc:e0a0e2168013305f48564ee4b7c24238eefbf2b8"
meta-msc-ldk-core-recipes.git = "master:578ccad7af23219d5873a6d32b50806475ecb233"
meta-msc-ldk-core.git = "master:409f83c4027ca9a21994326334a4bbe4e7a9b211"
meta-msc-ldk-mscio.git = "master:ede1abf912247df8e1f853453fc46208b41dd05b"
meta-intel-leafhill
meta-intel-middleware = "jethro-msc:825b8edb2077037e43073543a9adac9739f05414"
meta-networking    = "jethro-msc:e0a0e2168013305f48564ee4b7c24238eefbf2b8"
meta-msc-ldk-hwtests.git = "master:a9b42de3a42322f001ddc02bb17a090b27202aed"
meta-msc-ldk-hwtests-recipes.git = "master:15318)
    baee7ba08262447e9463d459ae2ce1c6c01"}
```

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/C984-baytrail-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/C984*
user@devhost:msc-ldk/build/C984*$ 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=C984 --layers-lxqt
user@devhost:msc-ldk$ cd build/C984*-lxqt
user@devhost:msc-ldk/build/C984*-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/C984*-lxqt
user@devhost:msc-ldkbuild/C984*-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/C984*
user@devhost:msc-ldk/build/C984*$ 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/C984*
user@devhost:msc-ldk/build/C984*$ 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) or `sato` (standard Yocto GUI).

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. Adding Other Packages

Further packages can be included in the images by adding these lines to:
`build/C984*/conf/local.conf`:

```
IMAGE_INSTALL += " \
   my-foo-package \
"
```

Further information can be found here:
http://www.yoctoproject.org/docs/2.7/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/C984*/conf/local.conf`. A non-existing file is silently ignored.

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

user@devhost:msc-ldk$ ./setup.py --bsp=C984 --local-conf-append=conf.append
```

4.15.2. Adding Own Layers

The images can be further enhanced or configured by adding own layers. Extend the file
`build/C984*/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/2.7/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=C984 --add-layer=https://github.com/OSSystems/meta-browser
```

4.15.3. 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/C984*/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 = ""
```

4.15.4. 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/C984*/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`.

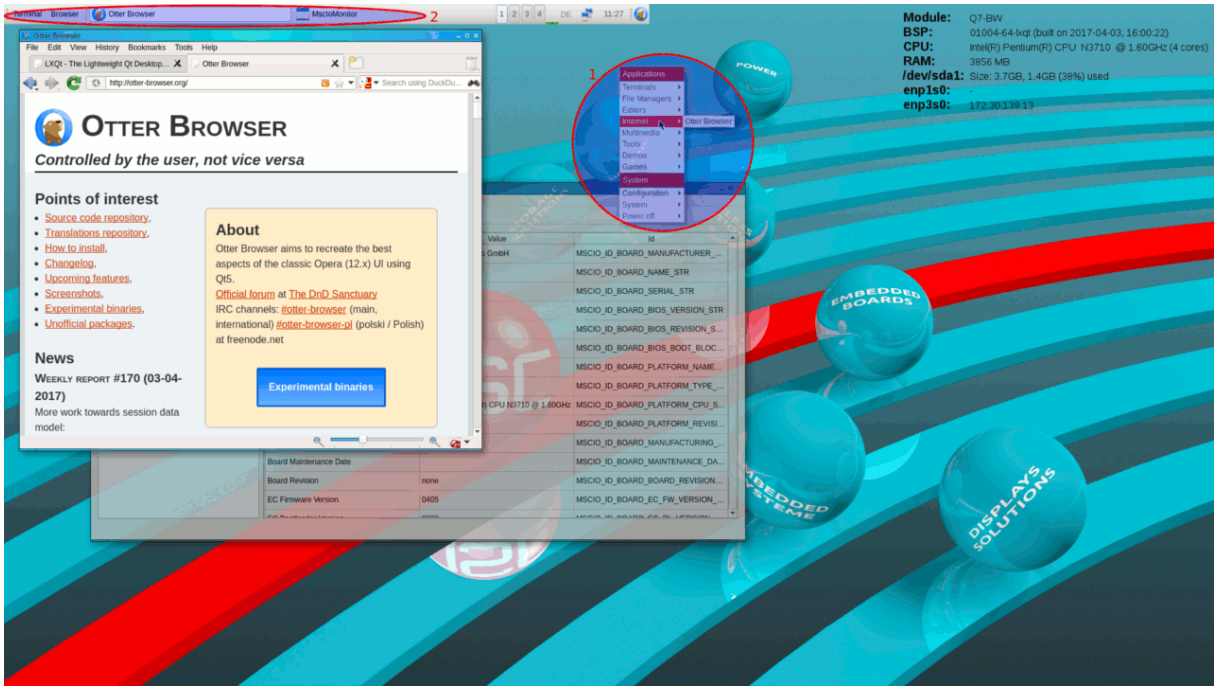
It is also possible to change the time zone of a running system to e.g. New York, using the following shell commands:

```
# 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.

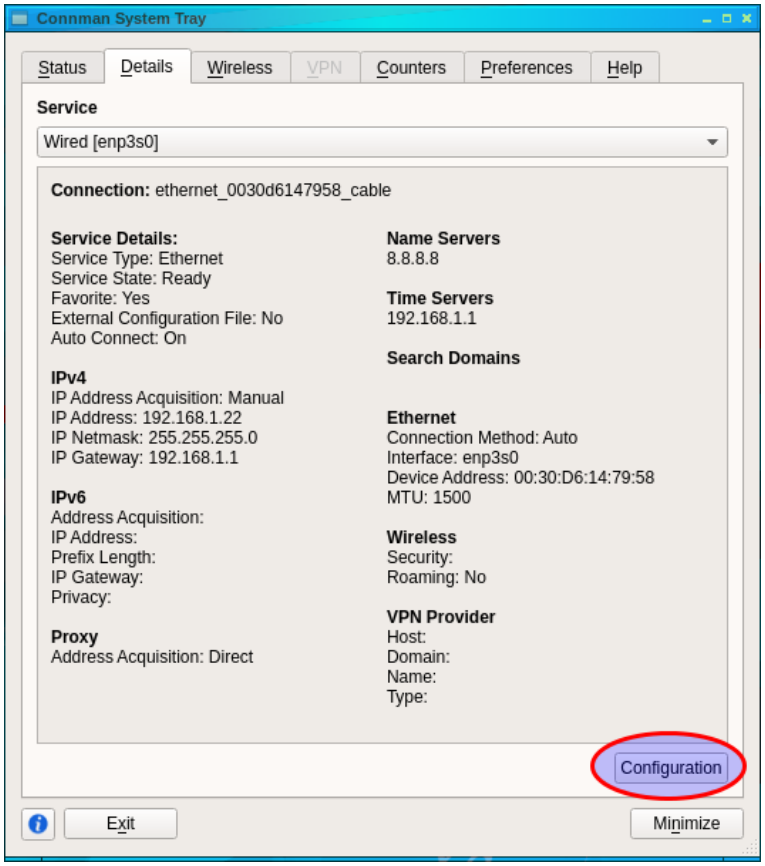
4.17. ConnMan Configuration

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

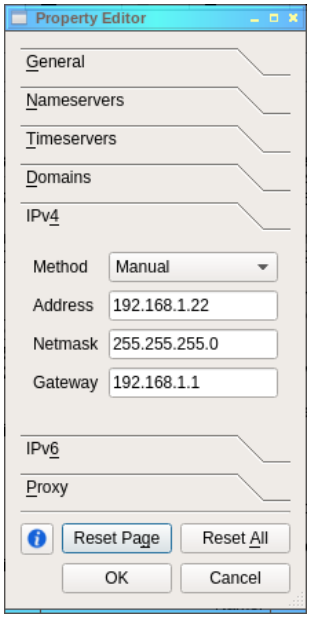
There is a detailed overview for each connection when the Details tab is selected:

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



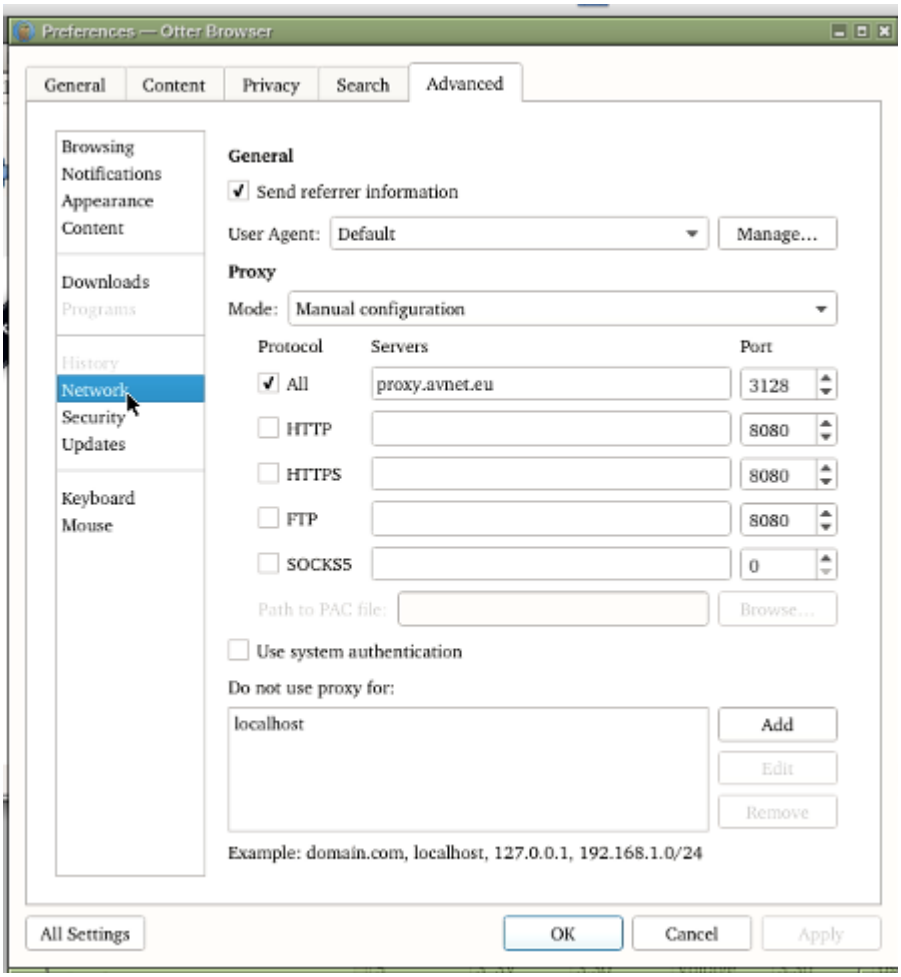
The tool `connman-info` can be used to show retrieve network interface details:

```
# 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 4.15.3).

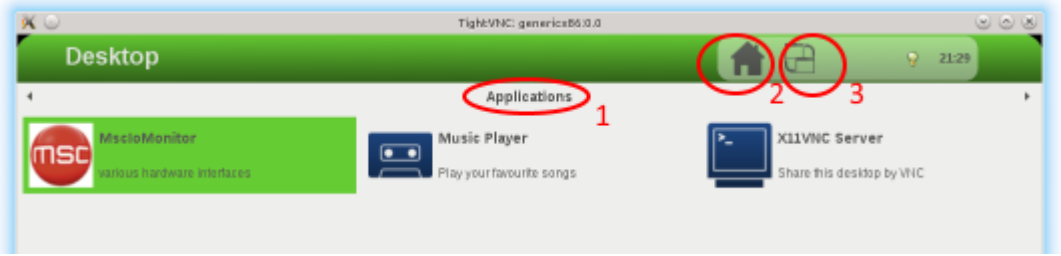
Figure 4.6. – Otter Browser - Proxy Setup



4.18. Using Sato

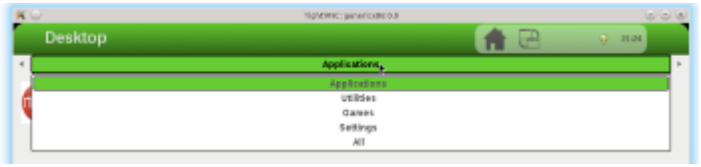
Sato is a window manager in which applications run full-screen, optimized for use with touch-displays.

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`.

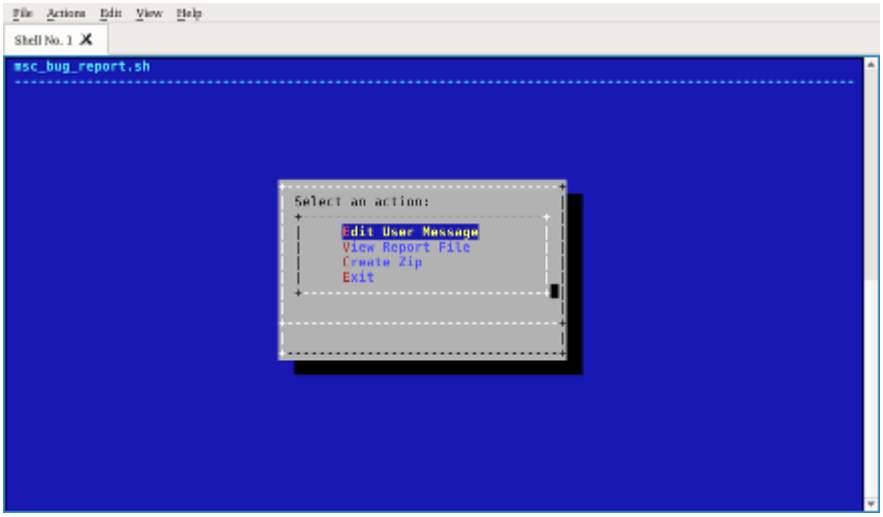
To enable german keyboard layout, add the kernel command line option: `keyboard=de-latin1-nodeadkeys`

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.

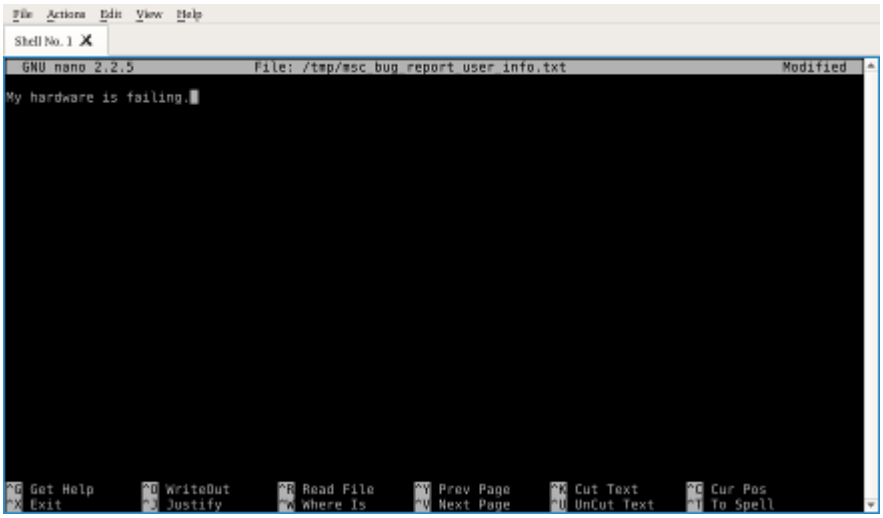
- Run `msc_bug_report.sh`.

Figure 4.10. – Bug Reporting - Start



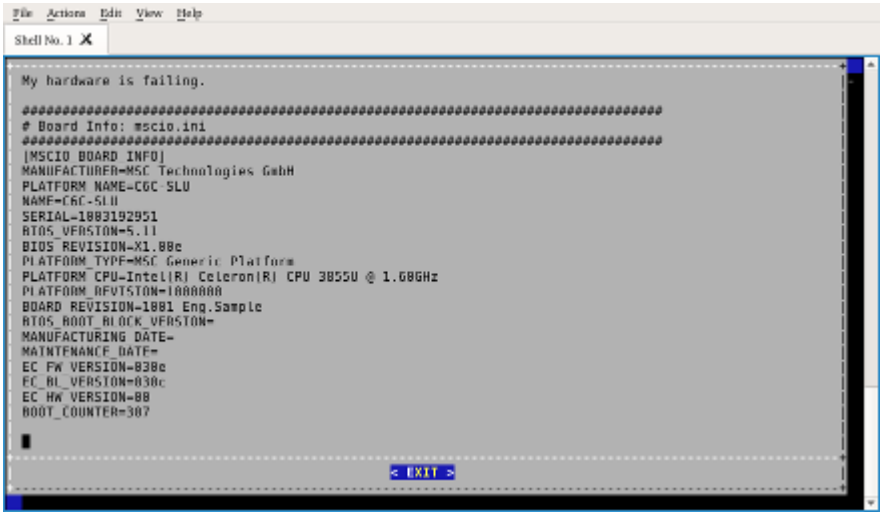
- Select “Edit User Message”.
- Enter bug report message and press Ctrl-O and Ctrl-X.

Figure 4.11. – Bug Reporting - Start



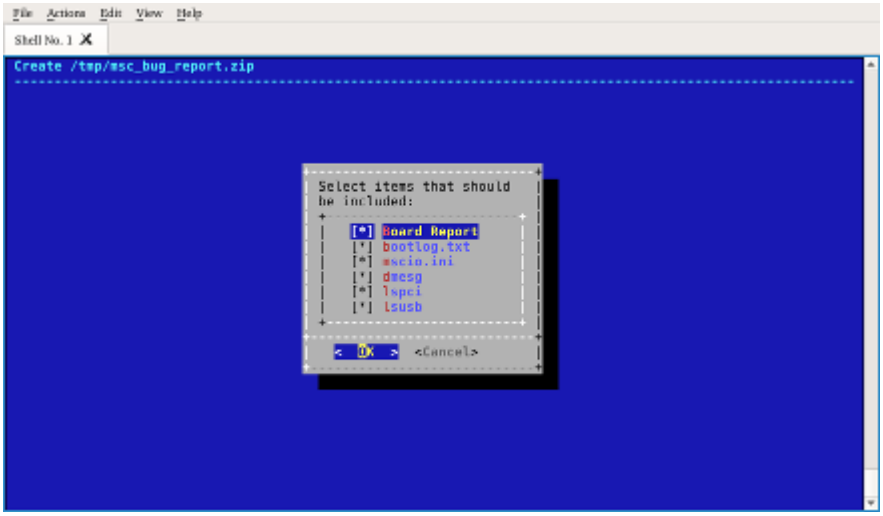
- Optionally you can then view the message with the board report (hardware information).

Figure 4.12. – Bug Reporting - View



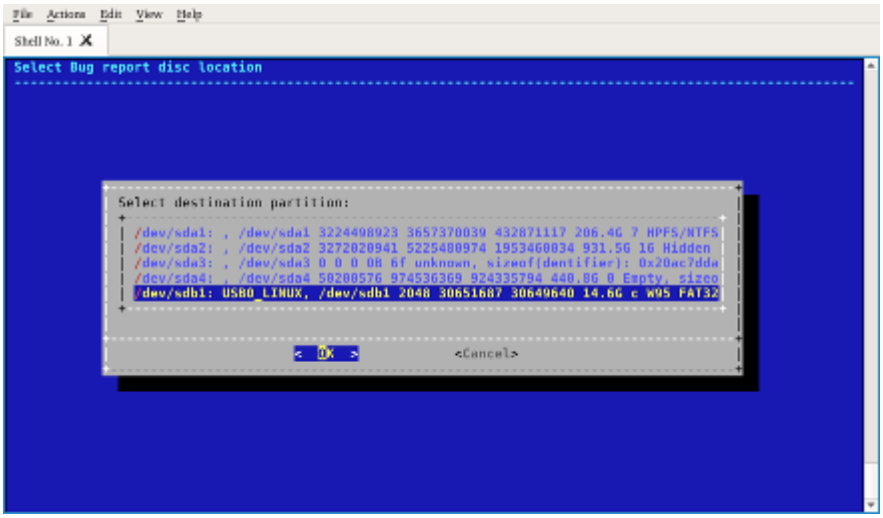
- Press “Create a zip file” and select the components you want to send (e.g. bootlog, mscio.ini, last kernel logs (dmesg) or the installed hardware).

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 v1.5.0`) 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. 17, 21

EAPI Embedded Application Programmer Interface. 37

EFI Extensible Firmware Interface. 39

MSC-LDK MSC Linux Development Kit. 13, 17

RPMB Replay Protected Memory Block. 44

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
```

```
serial --unit=0 --speed=115200 --word=8 --parity=no --stop=1
default=msc-image-hwtests-valleyisland-32
timeout=2

menuentry 'msc-image-base-valleyisland-32'{
          linux /images/bzImage-valleyisland-32.bin LABEL=base loglevel=3 vga=1 vmalloc=256M
              ramdisk_size=196608 root=/dev/ram
          initrd /images/msc-image-base-valleyisland-32.cpio.gz
}

menuentry 'msc-image-sato-valleyisland-32'{
          linux /images/bzImage-valleyisland-32.bin LABEL=sato loglevel=3 vga=1 vmalloc=256M
              ramdisk_size=196608 root=/dev/ram
          initrd /images/msc-image-sato-valleyisland-32.cpio.gz
}

menuentry 'msc-image-hwtests-valleyisland-32'{
          linux /images/hwtests/bzImage-valleyisland-32.bin LABEL=hwtests loglevel=3 vga=1
              vmalloc=256M ramdisk_size=196608 root=/dev/ram modprobe.blacklist=igb,ahci,
              snd_hda_intel,mmc_block mscio_disable=y
          initrd /images/hwtests/msc-image-hwtests-valleyisland-32.cpio.gz
}

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

menuentry 'msc-image-sato-valleyisland-64'{
          linux /images/bzImage-valleyisland-64.bin LABEL=sato loglevel=3 vga=1 vmalloc=256M
              ramdisk_size=196608 root=/dev/ram
          initrd /images/msc-image-sato-valleyisland-64.cpio.gz
}

menuentry 'msc-image-hwtests-valleyisland-64'{
          linux /images/hwtests/bzImage-valleyisland-64.bin LABEL=hwtests loglevel=3 vga=1
              vmalloc=256M ramdisk_size=196608 root=/dev/ram modprobe.blacklist=igb,ahci,
              snd_hda_intel,mmc_block mscio_disable=y
          initrd /images/hwtests/msc-image-hwtests-valleyisland-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. Graphic Demos

When launching applications from a telnet/ssh login, the following command must be executed first:
`export DISPLAY=:0.0`

D.1. Graphic Driver Status

D.1.1. igt-gpu-tools

This is a collection of performance, information and debugging tools for intel cards.

Table D.1. – Intel IGT GPU Tools

Application	Description
`intel_gpu_top`	Displays the GPU frequency, power usage and subsystem load.

D.2. Video Acceleration

D.2.1. GStreamer

With GStreamer it is possible to playback H.264/MP4 videos accelerated by the video hardware.

- Automatic selection of decoder and playback devices:

```
gst-launch-1.0 playbin uri=file:///video.mp4
```

- When audio is not available (message " The autoaudiosink element is not working"), disable audio manually:

```
gst-launch-1.0 playbin uri=file:///video.mp4 audio-sink=fakesink
```

- Verbose output:

```
gst-launch-1.0 -v playbin uri=file:///video.mp4 audio-sink=fakesink
```

- Manually selection of the video pipeline:

```
gst-launch-1.0 filesrc location=/video.mp4  ! qtdemux ! vaapidecodebin ! vaapisink ↩
    fullscreen=true
```

- To list all available sources:

```
gst-inspect-1.0 | grep src
```

- To list all available sinks/playback devices:

```
gst-inspect-1.0 | grep sink
```

- To list all available decoder:

```
gst-inspect-1.0 | grep decoder
```

- To list all available encoder:

```
gst-inspect-1.0 | grep encoder
```

The load can be displayed by running `intel_gpu_top`.

The measured CPU load should increase only by 10% while the render can increase by 40%.

E. Supported Periphery

Table E.1. – Supported periphery

Periphery	Status	Comment
ACPI	implemented	see ACPI
Audio	implemented	see Audio
Backlight	implemented	see Backlight
Beeper	implemented	
Ethernet	implemented	
eMMC	implemented	`/dev/mmcblk1`, see eMMC
GPIOs	implemented	see GPIOs
Graphic output	implemented	see Dual Head Support
Video encoding/decoding (H.264)	implemented	see Video Acceleration
Hardware sensors	implemented	
I2C	implemented	see I2C
LPC	implemented	use `superiotool`.
NVRAM	implemented	see NVRAM
PCIe	implemented	
RTC	implemented	
SATA	implemented	
SD/MMC	implemented	see SD/MMC
SMBus	implemented	see I2C Bus Identification
SPI	implemented	see SPI
TPM	implemented	
UART	implemented	see UART
USB-2.0	implemented	
USB-2.0 device	implemented	see USB Device Support
USB-3.0	implemented	Within BIOS Option `Chipset/South Bridge/USB Configuration`, `USB 2.0 (EHCI) Support` must be disabled and `XHCI Mode` be enabled. This is not the default.
USB-3.0 device	implemented	
Watchdog	implemented	see Watchdog

E.1. eMMC

eMMC is available as an MMC device .

eMMC chips differ to MMC cards that they provide additional boot partitions and an RPMB. Detection of the RPMB causes a significant boot delay of 10 seconds as support for it is not fully integrated in Linux.

As RPMB is typically unused, the C984 BSP has detection of the RPMB disabled by default. To enable it, either provide a kernel command line option `mmc_block.mmcblk.disable_rpmb=0` or a module option for modprobe `mmcblk.disable_rpmb=0`. This will cause these boot errors.

```
mmcblk1rpmb: error -110 transferring data, sector 8064, nr 8, cmd response 0x900, card status
     0xb00
[   26.791047] mmcblk1rpmb: retrying using single block read
[   26.793191] mmcblk1rpmb: timed out sending r/w cmd command, card status 0x400900

udevd[74]: worker [77] timeout, kill it
udevd[74]: seq 2179 '/devices/pci0000:00/0000:00:17.0/mmc_host/mmc1/mmc1:0001/block/mmcblk1/
    mmcblk1rpmb' killed
```

E.2. GPIOs

GPIOs are available. Add the `Base GPIO#` to the hardware GPIO number. E.g. for `GPIO_S0_SC_55` add 55 to 154 to get the GPIO number 209. Then access this GPIO via `/sys/class/gpio`.

Table E.2. – GPIOs

Group	Base GPIO#	gpiochip label
S5	82	byt_gpio.2
SC	154	byt_gpio.0

E.3. Audio

E.3.1. 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`

E.3.2. 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 (rear) microphone, do:

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

Playback With HDMI

HDMI is not enabled by default. To enable it:

- 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 -c 2
  ```

- In some cases, sound might not be audible. In these cases, run:

  ```
  amixer sset 'IEC958' off; amixer sset 'IEC958' on
  ```

  or in alsamixer toggle both controls `S/PDIF` with the button ’m’ twice.
- 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
  }
  ```

Persistent Mixer Settings

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

E.4. I2C

To use the I2C bus, the BIOS Option `Advanced/Module-specific Initialization/User I2C Support` must be set to `Controller-based`, otherwise I2C support is broken. This setting is not the default.

Then the I2C bus is available as the bus named `i2c-designware-pci`, configured to 100kHz (standard speed). The actual bus number can be different depending on the baseboard, therefore run `i2cdetect -l` to list all busses and determine the I2C bus number. Or use the I2C Bus Identification feature.

E.5. USB Device Support

USB device support means the board appears as an USB device and can be connected to another host. This is achieved by loading various gadget drivers. All these tests assume that the hardware is connect to a linux host (which may be the board itself).

E.5.1. Setting Up USB Device Functionality

USB Device functionality is provided by some hardware variants on port 1. It is not a full USB OTG support. The automatic detection whether the USB port 1 is used as host or device is done by BIOS on reboot depending on the USB connection.

For these examples, it is recommended to configure the USB port1 for device. This can be done in the BIOS setup under `Advanced/Module-specific Initialization/USB Port 1 Mode`.

The BIOS Version X1.00e and before further require to disable the reset line ULPI_RESET# within Linux. This can be done before loading the gadget drivers with this command:
`mem_edit -a 0xfed0e110 -w 0x2003cc81 -a 0xfed0e118 -w 0x3`

E.5.2. Testing Connection

To load the driver, do:

```
modprobe g_zero
```

A new USB device `0525:a4a0 Netchip Technology, Inc. Linux-USB "Gadget Zero"` will appear.

To disable the device, run:

```
rmmod g_zero
```

E.5.3. Setting Up Serial Gadget

To load the driver, do:

```
modprobe g_serial
```

A new USB device `0525:a4a7 Netchip Technology, Inc. Linux-USB Serial Gadget (CDC ACM mode)` will appear.

On the host side, open a reader via `stty -F /dev/ttyACM0 raw; dd if=/dev/ttyACM0`. On the device, send some text via `hostname >/dev/ttyGS0`.

To disable the device, run:

```
rmmod g_serial
```

E.5.4. Setting Up Ethernet Gadget

To load the driver, do:

```
modprobe g_ether
```

A new USB device `Netchip Technology, Inc.  Linux-USB Ethernet/RNDIS Gadget` will appear.

On the host side, assign an IP address with `ifconfig usb0 192.168.160.2`. On the device, assign an IP address with `ifconfig usb0 192.168.160.3`. The host can now ping the device with `ping 192.168.160.3`.

To disable the device, run:

```
rmmod g_ether
```

E.5.5. Setting Up Mass/File Storage Gadget

To setup a simulated mass storage device, do:

```
dd if=/dev/zero of=/tmp/image.bin bs=1K count=128
mkfs.vfat /tmp/image.bin
mount -o loop /tmp/image.bin /mnt
cp /etc/hostname /mnt
umount /mnt
modprobe g_mass_storage file=/tmp/image.bin
```

A new USB device `Netchip Technology, Inc.  Linux-USB Storage Gadget` will appear.

It can be mounted and the contents modified.

To disable the device, run:

```
rmmod g_mass_storage
```

E.6. UART

The kernel reserves 4 serial port devices (`/dev/ttyS0` ... `/dev/ttyS3`) even when the PCU UART is disabled in the BIOS (default) and the baseboard has no superio connected. Accessing any of these devices will result in I/O errors.

Use the kernel log to determine how many UART devices are really existing, e.g. with:

```
grep ttyS /var/log/bootlog.txt
```

But the kernel can be configured to reserve less UART ports with the kernel command line option `8250.nr_uarts=<n>`, e.g. `8250.nr_uarts=1`

A login shell is created on `/dev/ttyS0` running with 115200 baud, 8 databits, no parity.

*****

E.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 50,100,208.

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:            21 (3.3 ~ 3.4 V)
bus mode:       2 (push-pull)
chip select:    0 (don't care)
power mode:     2 (on)
bus width:      2 (4 bits)
timing spec:    3 (sd uhs SDR12)
signal voltage: 1 (1.80 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
```

E.8. SPI

SPI is available as `/dev/spidev0.0` limited to 50 MHz and configured for SPI Mode 0.

The baseboard signal `BIOS_DISABLE#` must be kept high for the SPI bus to be available.

E.9. Powermanagement

E.9.1. Supported Capabilities

To determine whether suspend to ram/S3 is supported, run:

```
pm-is-supported --suspend && echo "S3 is supported" || echo "S3 is not supported"
```

To determine whether hibernation/suspend to disk/S4 is supported, run:

```
pm-is-supported --hibernate && echo "S4 is supported" || echo "S4 is not supported"
```

E.9.2. Suspend To RAM / S3

To suspend to RAM, run `pm-suspend`. The RAM will be placed into self-refresh mode and the CPU be disabled until Wake-up Sources trigger the wake-up. The power supply must still be enabled.

E.9.3. Wake-up Sources

Power-Button

The power button is always enabled as a wake-up source.

RTC

To wake-up in `n` seconds, enter the absolute system time in seconds when to wake-up in the file wakealarm. To wake-up in `+n` seconds, get the current system time and increment it.

This code sets the wake-up alarm in 15 seconds from now. If it was successfull, the system is put in standby. Note that going to standby takes some time, but we must be asleep for the wake-up alarm to take effect.

```
expr $(date '+%s') + 15 > /sys/class/rtc/rtc0/wakealarm && pm-suspend
```

Ethernet (Wake On LAN)

To wake on LAN because of magic packets, configure the ethernet controller to accept magic packets (this is the default):

```
ethtool -s eth0 wol g
```

To disable wake on LAN, run:

```
ethtool -s eth0 wol d
```

To remotely wake-up the module, send a magic packet from the development host to it:

```
sudo etherwake -i <ethernet interface the module is connected to> <MAC address>
```

for example:

```
sudo etherwake -i eth1 00:30:d6:10:36:c7
```

USB

Resuming from USB is enabled by default. But as it requires that the USB devices are being powered in standby, this functionality depends on the baseboard. On some baseboards (e.g. for Q7), resuming from USB is therefore not possible.

```
pm-suspend
```

Now press a key on the USB keyboard to resume operation.

E.10. Graphic Output

E.10.1. Dual Head Support

The BSP supports output on multiple displays. The default mode is to clone the display on HDMI/Displayport (named `DP-1`) and LVDS (named `DP-2`).

- To disable a display, run:

  ```
  xrandr --output <display> --off
  ```

  e.g. to disable the HDMI/Displayport display, run:

  ```
  xrandr --output DP-1 --off
  ```

- To enable a display, run:

  ```
  xrandr --output <display> --auto
  ```

- To extend the display with LVDS left of the HDMI, run:

  ```
  xrandr --output DP-2 --left-of DP-1
  ```

- To clone both displays, run:

  ```
  xrandr --output DP-2 --same-as DP-1
  ```

- To change the brightness of the LVDS display to 50%, run:

  ```
  xrandr --output DP-2 --brightness 0.5
  ```

LVDS (DP-2) is always enabled even if no hardware is connected.

E.11. ACPI

ACPI events for buttons (Sleep, Power, LID) are supported but without any default action configured.

By running `acpi_listen`, any buttons pressed will be displayed on the console.

To configure an action, e.g. for the LID button, add a file `/etc/acpi/events/lid-button`.

```
event=button/lid
action=echo "%e" >/dev/console
```

Then run `killall -HUP acpid` to make acpid read the new configuration file. The next time the LID button is pressed or released, a message will be printed on the system console.

F. MSC-IO

F.1. What Is MSC-IO?

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

F.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.

F.3. mscio-drivers

For fully supporting the board, some drivers are necessary:

Table F.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.

F.3.1. 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`.

F.3.2. 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
```

F.3.3. 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
```

F.3.4. 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.

F.3.5. 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.

F.3.6. 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
11cc22ddffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffffff
```

F.3.7. 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
```

F.3.8. 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*`

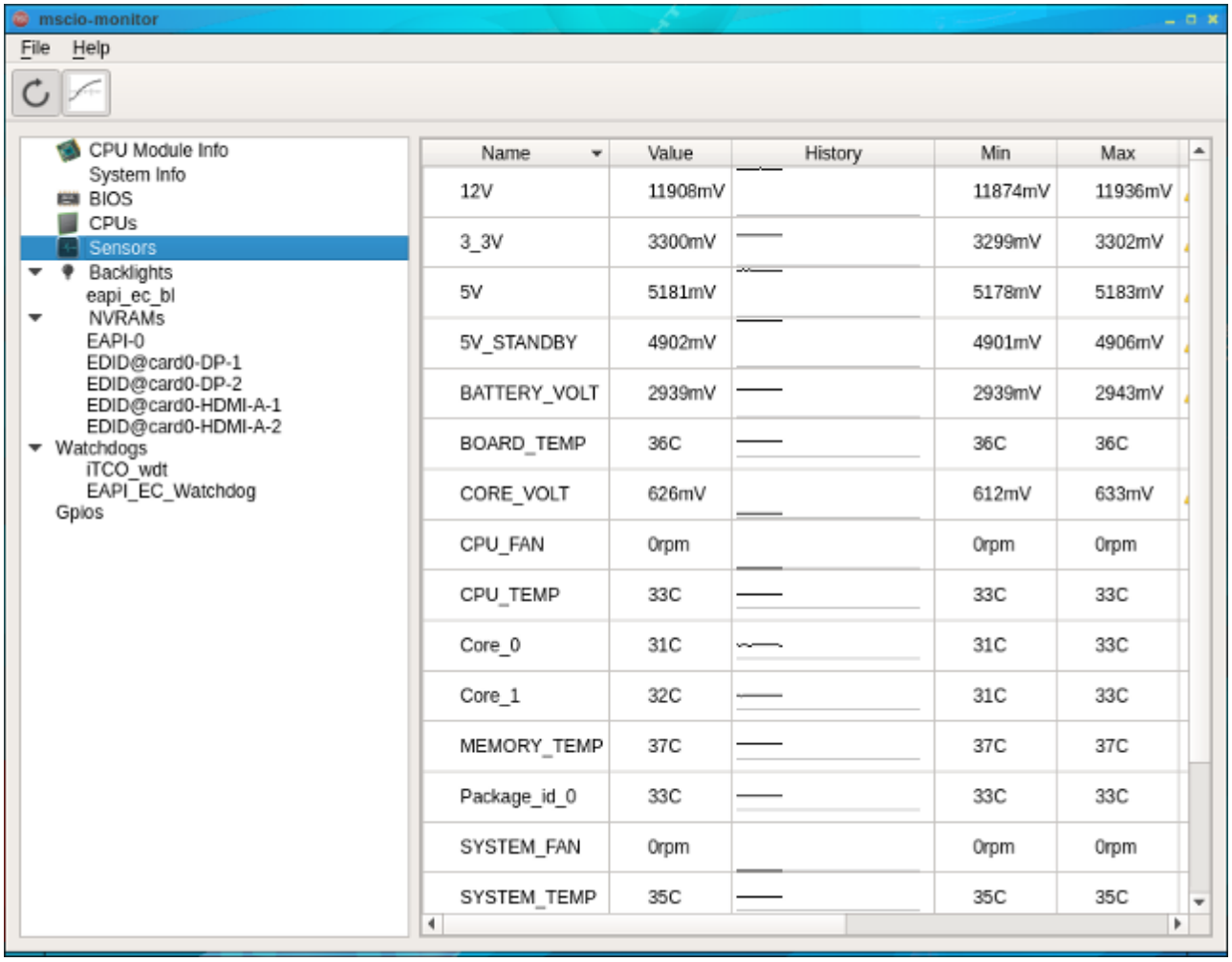
```
root@intel-corei7-64:~# i2cdetect -l
i2c-0   i2c             i915 gmbus dpc                          I2C adapter
i2c-5   i2c             Synopsys DesignWare I2C adapter         I2C adapter
i2c-5   *link*          user                                    ID mapped
i2c-6   *link*          smbus    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

F.4. mscio-monitor

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

Figure F.1. – mscio-monitor



F.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 F.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
```

G. Troubleshooting

Table G.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`

H. Tips And Tricks

H.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`.

H.1.1. Using The google-chrome Browser

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

```
user@devhost:msc-ldk$ cat << _EOF >chrome.append
IMAGE_INSTALL += " \
   google-chrome \
   "
_EOF

user@devhost:msc-ldk$ ./setup.py --bsp=C984 \
   --local-conf-append=chrome.append \
   --add-layer=https://github.com/OSSystems/meta-browser
```

This will install google-chrome additional to the otter-browser. Replacing `otter-browser` can be achieved via a special layer without the need for `chrome.append`

```
user@devhost:msc-ldk$ ./setup.py --bsp=C984 \
   --add-layer=https://github.com/OSSystems/meta-browser \
   --add-layer=ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/meta-chrome-instead-of-otter
```

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

H.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.

H.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
```

I. Security

I.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.

I.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/C984*/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.

I.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=C984 --layers-lxqt --layers-security
user@devhost:msc-ldk$ cd build/C984*-security
user@devhost:msc-ldk/build/C984*-security$ make msc-image-lxqt
```

J. Real-time

J.1. What Is Real-time

Real-time is the capability of an operating system to react on a request within a defined time. There are various categories.

- Hard Real-time - The operating system ensures that it can react to any event within a maximum time.
- Soft Real-time - The operating system is capable to react on most events within a maximum time. Some sporadic events might take longer, e.g. error handling or too many parallel interrupts requests.

Modern standard kernels are considered soft real-time. They reduce the overall latency and improve preemption. To further boost preemption Yocto includes the PREEMPT_RT patches for it's standard kernels.

J.2. Setup A BSP For Real-time Support

To build an image with real-time support two requirements must be met:

1. The kernel must have included and enabled the real-time patches.
2. User-space tools for real-time configuration and analysis should be included in the image.

This can be achieved by setting up the BSP as:

```
user@devhost:msc-ldk$ ./setup.py --bsp=C984 --layers-lxqt --layers-rt
user@devhost:msc-ldk$ cd build/C984*-lxqt-rt
user@devhost:msc-ldk/build/C984*-lxqt-rt$ make msc-image-lxqt
```

To check that the running system has real-time support enabled:

- `/proc/version` must contain the string `PREEMPT RT`, e.g.
  ```
  Linux version 4.8.3-rt2-yocto-preempt-rt (buildserver@destsm3ux05bs01.emea.avnet.com) (gcc
  version 6.2.0 (GCC) ) #3 SMP PREEMPT RT Mon Apr 3 10:37:02 CEST 2017
  ```
- The kernel must have full preemption enabled:
  ```
  gunzip -c /proc/config.gz | grep CONFIG_PREEMPT_RT_FULL=y
  CONFIG_PREEMPT_RT_FULL=y
  ```
- The file `/usr/bin/cyclictest` must be present.

J.3. Analysis Of Real-time Behavior

J.3.1. Yocto Tracing And Profiling

Detailed tracing of kernel activity can be enabled using trace-cmd.

Requirements

Tracing requires the tool kernelshark.

This can be installed on ubuntu with:

```
sudo apt-get install trace-cmd kernelshark
```

Tracing System Calls

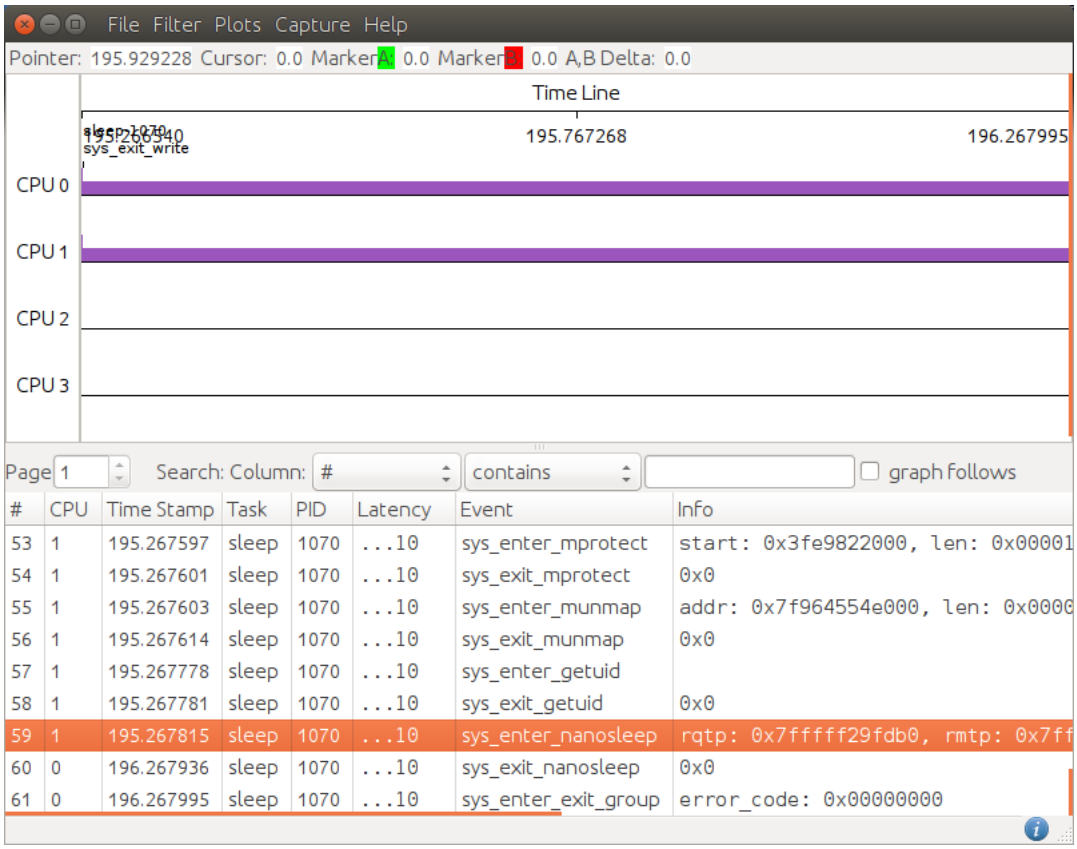
To trace the system calls of `sleep`, do:

```
root@intel-corei7-64:~# trace-cmd record -o trace.dat -e "syscalls:*" -F sleep 1
```

The results can then be viewed on the host after copying trace.dat to the host or sharing it by an NFS mounted directory with:

```
$ kernelshark trace.dat
```

Figure J.1. – kernelshark - systemcalls



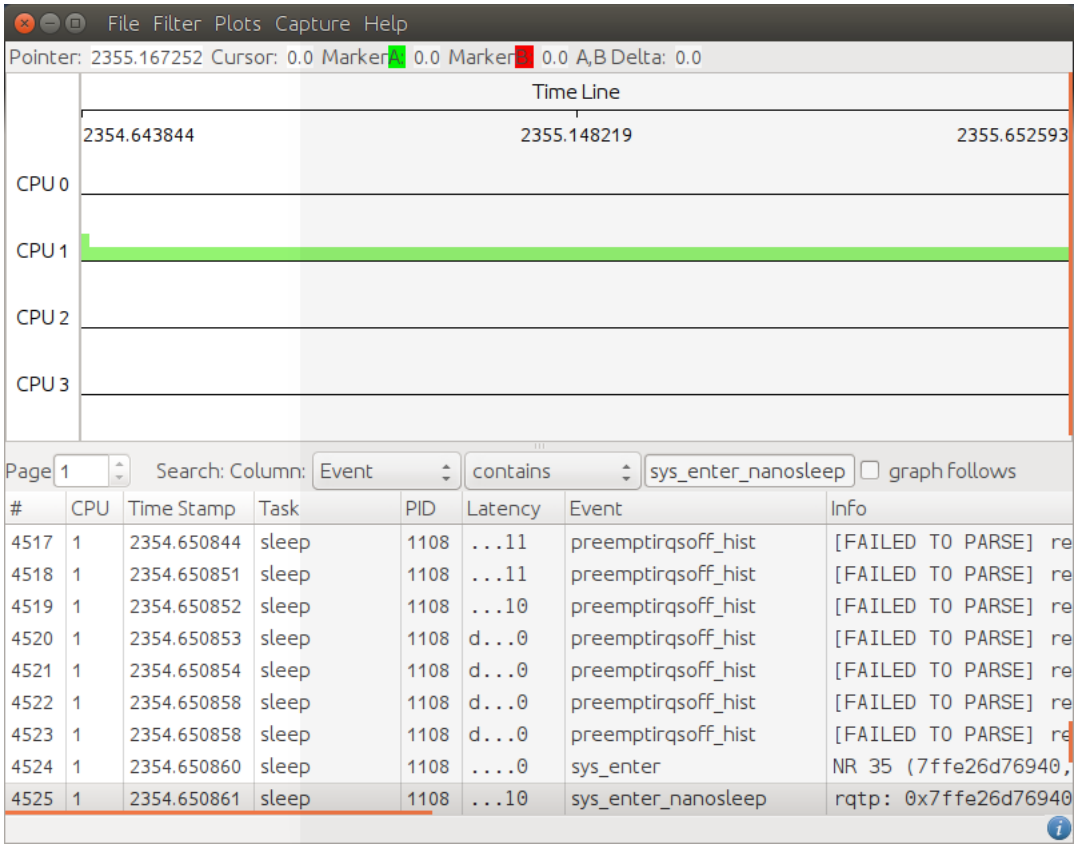
Tracing Kernel Events

To have a more detailed view of the system, do:

```
root@intel-corei7-64:~# trace-cmd record -o trace.dat -e "*" -F sleep 1
```

Then search in kernelshark for `sys_enter_nanosleep` and then scroll the following events until `sys_exit_nanosleep` appears.

Figure J.2. – kernelshark - events



A function stack trace can be added to the report. This might require an increase of the trace buffer size.

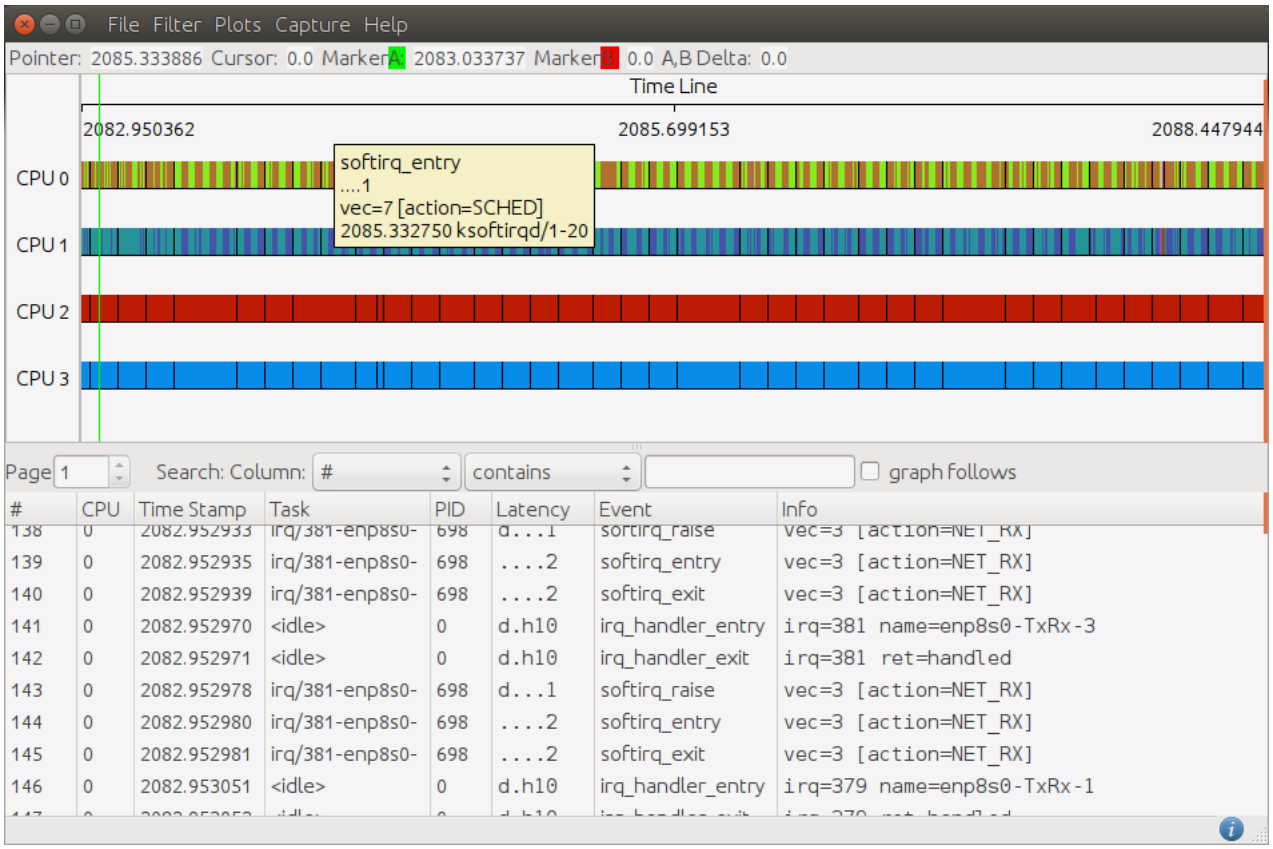
```
root@intel-corei7-64:~# trace-cmd record -b 8192 -T -o trace.dat -e "*" -F sleep 1
```

Tracing IRQs

If the option `-F` is omitted by `tracecmd` all events on the system will be reported. This comes in handy for IRQ analysis.

```
root@intel-corei7-64:~# trace-cmd record -o trace.dat -e "irq:*"
```

Figure J.3. – kernelshark - IRQs



J.3.2. Application List With Real-Time Priorities

To list all applications with their real-time priorities use the command

```
ps -Hxo pid,s,psr,cls,rtprio,comm
```

This will display all processes with their process ID, their state (sleeping or running), the CPU (0...3), the real-time class (TS=Other, FF=real-time fifo, RR=real-time round-robin), the real-time priority from 1 (lowest) to 99 (highest) when class is FF or RR and finally the command name.

```
  PID S PSR CLS RTPRIO COMMAND
    1 S   0  TS      - init
    2 S   1  TS      - kthreadd
    3 S   0  TS      - ksoftirqd/0
    4 S   0  FF      1 ktimersoftd/0
    6 S   0  TS      - kworker/0:0H
    8 S   0  FF     51 rcu_preempt
    9 S   1  FF     51 rcu_sched
   10 S   0  FF     51 rcub/0
   11 S   0  FF     51 rcuc/0
   13 S   0  FF     99 posixcputmr/0
   14 S   0  FF     99 migration/0
   17 S   1  FF     99 migration/1
   26 S   2  FF     51 rcuc/2
   35 S   3  FF     51 rcuc/3
   82 S   1  FF      1 i915/signal:0
  120 S   0  FF     50 irq/41-mmc0
  121 S   1  FF     49 irq/41-s-mmc0
  122 S   0  FF     50 irq/348-0000:00
 1052 S   1  TS      - sh
 1057 S   1  TS      - kworker/u8:0
 1064 T   0  TS      - cyclictest
 1064 T   3  RR     40 cyclictest
 1067 R   1  TS      - ps
```

- There are some threads like `migration` and `posixcputmr` that are required for real-time management and therefore run with priority 99.
- Then there are the `rcuc` threads which are necessary for sharing resources with the other CPUs. Their priority is defined by the kernel variable `CONFIG_RCU_KTHREAD_PRIO` and is set to 51. Real-Time applications should have a priority lower than this value.
- Below it are the interrupt handlers and userspace real-time applications.
- Then there are the non-real-time kernel worker threads and user-space applications.

Even an isolated CPU 3 will always be used by some kernel threads.

```
root@intel-corei7-64:~# ps -Hxo pid,s,psr,cls,rtprio,comm|grep "  3  "
   33 S   3  TS      - cpuhp/3
   34 S   3  FF     99 migration/3
   35 S   3  FF     51 rcuc/3
   36 S   3  FF      1 ktimersoftd/3
   37 S   3  TS      - ksoftirqd/3
   38 S   3  TS      - kworker/3:0
   39 S   3  TS      - kworker/3:0H
   40 S   3  FF     99 posixcputmr/3
  156 S   3  TS      - kworker/3:1
```

J.4. Latency

The system latency can not be calculated on a modern system. It is influenced by too many variables:

- Two or three level caches
- DMA transfers, especially to PCI devices
- Displays and shared frame buffers (continuous DMA)
- Hardware interrupts
- Multiple cores
- Operating system scheduling
- System management interrupts and non-maskable interrupts
- Automatic clock frequency adaption, e.g. `intel_pstate` driver

The OSADL QA farm continuously runs latency benchmarks for various systems which can be used for comparison.

J.4.1. cyclictest

The tool cyclictest can be used to measure the delay of an event, here the clock timer interrupt, and the awaking of a waiting application.

Two threads are used. The main console thread is used for status display and a background thread for the measurement. This will wait until a specific time has been reached. It will then measure the delta time from the expected timer interrupt time and the awake time. The minimum, maximum and average time is calculated and displayed by the main thread.

This application doesn't use much CPU load and is best used at least one priority higher than the own real-time application and below the rcu interrupt priority of 51.

```
root@intel-corei7-64:~# cyclictest -a 0 --policy rr -p 40 -N -t 1
# /dev/cpu_dma_latency set to 0us
policy: rr: loadavg: 0.04 0.01 0.00 1/307 1095

T: 0 ( 1095) P:40 I:1000 C:  16012 Min:  17838 Act:   19703 Avg:   20681 Max:   30451
```

This lists the latency in nanoseconds. The application should be run a long time while forcing hardware events like disk and network transfer, hot-plug events and other activity.

J.4.2. Hardware Latency

Another test is how long an application is interrupt by serving interrupts.

This creates a kernel thread running at real time priority 1. It runs in a loop and determines how long the loop was interrupt the most time. If no other real-time application is running the values represent the interruptions by serving hardware interrupts and inter-cpu synchronization.

```
sudo /bin/zsh
mount -t tracefs none /sys/kernel/tracing
cd /sys/kernel/debug/tracing
echo hwlat >current_tracer
sleep 30
cat trace >/tmp/hwlatresult
echo nop >current_tracer
echo -n >trace
cat /tmp/hwlatresult
```

With the output:

```
#
#                              _-----=> irqs-off
#                             / _-----=> need-resched
#                            | / _----=> hardirq/softirq
#                            || / _---=> preempt-depth
#                            ||| /     delay
#           TASK-PID   CPU#  ||||    TIMESTAMP  FUNCTION
#              | |       |   ||||       |         |
           <...>-1479  [001] d...   337.834497: #1     inner/outer(us):   14/59    ts
               :1539671945.135553380
           <...>-1479  [000] d...   338.840310: #2     inner/outer(us):    3/14    ts
               :1539671946.141366423
```

The maximum latency caused by hardware events is 59us.

Further information can be found here:

1. https://lwn.net/Articles/641678/
2. https://www.osadl.org/Create-a-frequency-plot-from-data-of-har.bash-script-for-hwlat-plots.0.html

J.4.3. Reducing Latency

Isolating CPU cores

On multi-core systems Linux can be instructed to avoid placing threads on specific cores. For example on a quad-core two cores can be used for all the hardware I/O, network management and the standard Linux applications and

the other two cores can be used for the real-time applications and necessary kernel management.

This can be achieved by adding the kernel command line option `isolcpus=2,3`

The real-time application can then be manually placed on the fourth CPU using `taskset`, e.g.

```
root@intel-corei7-64:~# taskset -c 3 cyclictest
```

If hyper-threading is used both virtual cores of the physical core should be isolated as they share the same cache.

Interrupt Affinity

Interrupts can be isolated so they are served only on specific CPUs. Typically Linux will try to process the interrupts on the same core 0 to avoid instruction cache updates.

In some cases it might be best to place the hardware interrupt handler to an isolated CPU as well so the interrupt handler resides in the instruction cache. This can be achieved by modifying it's affinity.

For example, to ensure that the SMBus Interrupts are handled only by core 2 and all other interrupts on one:

- On the kernel command line set the irq-affinity to the cores 0 and 1.

  ```
  irqaffinity=0,1
  ```

  Hardware interrupts but not timer and management interrupts will now be handled by core 0.
- Determine the interrupt number of the SMBus

  ```
  cat /proc/interrupts |grep smbus
    20:          6          0          0          0   IO-APIC   20-fasteoi   i801_smbus
  ```

  The interrupt number is 20.
- Determine the SMP affinity of interrupt 20. `smp_affinity` is a bit-mask for the allowed cores.

  ```
  cat /proc/irq/20/smp_affinity
  3
  ```

- Move it to core 2.

  ```
  echo 4 >/proc/irq/20/smp_affinity
  ```

- Ensure that further interrupts are now handled by core 2.

  ```
  root@intel-corei7-64:/proc/irq# i2cdetect -l
  [...]
  i2c-5   smbus           SMBus I801 adapter at f040              SMBus adapter
  root@intel-corei7-64:/proc/irq# i2cdetect -y 5
  [...]
  root@intel-corei7-64:/proc/irq# cat /proc/interrupts |grep smbus
    20:          6          0        116          0   IO-APIC   20-fasteoi   i801_smbus
  ```

Avoiding Sleeps

With standard BIOS settings the CPU will enter higher c-states when possible to reduce power consumption. As waking up requires a lot of time the CPU should be always fully turned on. This can be achieved by the kernel command line options `isolcpus=2,3 intel.max_cstate=0`

Constant Clock

Both the hardware and the operating system have the possibility to change the clock frequency based on the demand to save power when not needed. As clock frequencies can differ by factor two or more this has a great impact when waiting for events on waking up on them.

The automatic hardware adaption can be disabled by using the kernel command line option `intel_pstate=disable`

To disable software adaption select a different governor. At first determine the current settings.

```
root@intel-corei7-64:~# msc-cpufreq
CPU0: ondemand: 1000 MHz (800 MHz .. 1.60 GHz)
CPU1: ondemand: 1.60 GHz (800 MHz .. 1.60 GHz)
CPU2: ondemand: 800 MHz (800 MHz .. 1.60 GHz)
CPU3: ondemand: 800 MHz (800 MHz .. 1.60 GHz)
```

Some CPUs allow different governors for selected CPUs. Select the performance governor for maximum clock on these CPUs.

```
root@intel-corei7-64:~# for c in cpu2 cpu3; do echo performance >/sys/devices/system/cpu/$c/
    cpufreq/scaling_governor; done
```

And verify the new settings with:

```
root@intel-corei7-64:~# msc-cpufreq
CPU0: ondemand: 800 MHz (800 MHz .. 1.60 GHz)
CPU1: ondemand: 1.60 GHz (800 MHz .. 1.60 GHz)
CPU2: performance: 1.60 GHz (800 MHz .. 1.60 GHz)
CPU3: performance: 1.60 GHz (800 MHz .. 1.60 GHz)
```

J.4.4. Summary

The shortest latencies can be achieved using the kernel command line options:

```
intel_pstate=disable isolcpus=2,3 intel.max_cstate=0 processor.max_cstate=0
```

K. 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 K.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.

K.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.

K.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) ]
```

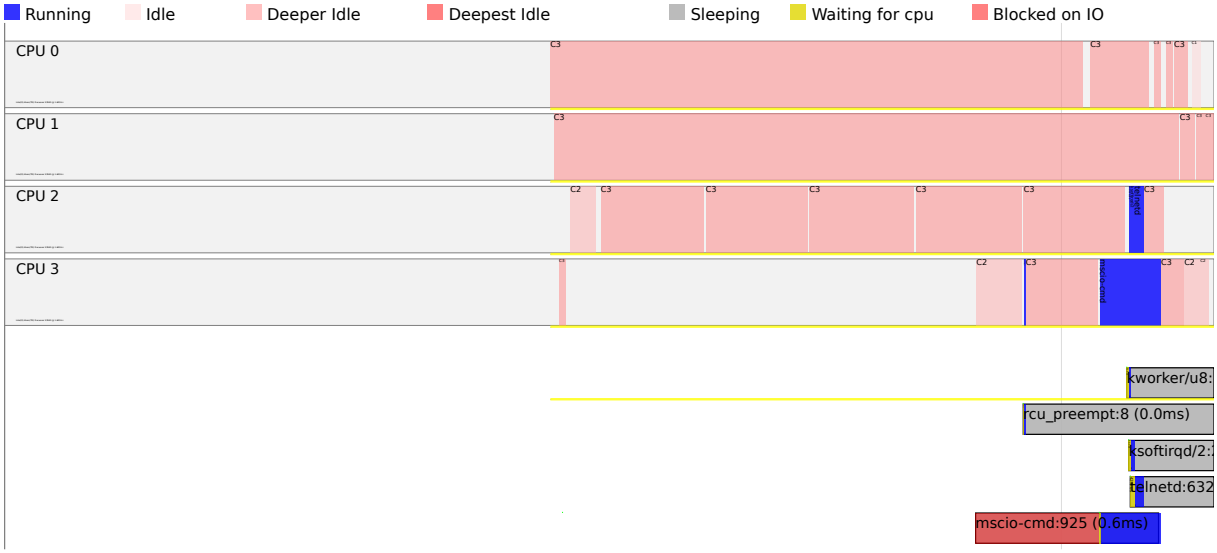
File K.1 – perf timechart record

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 K.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).

K.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.

K.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
```

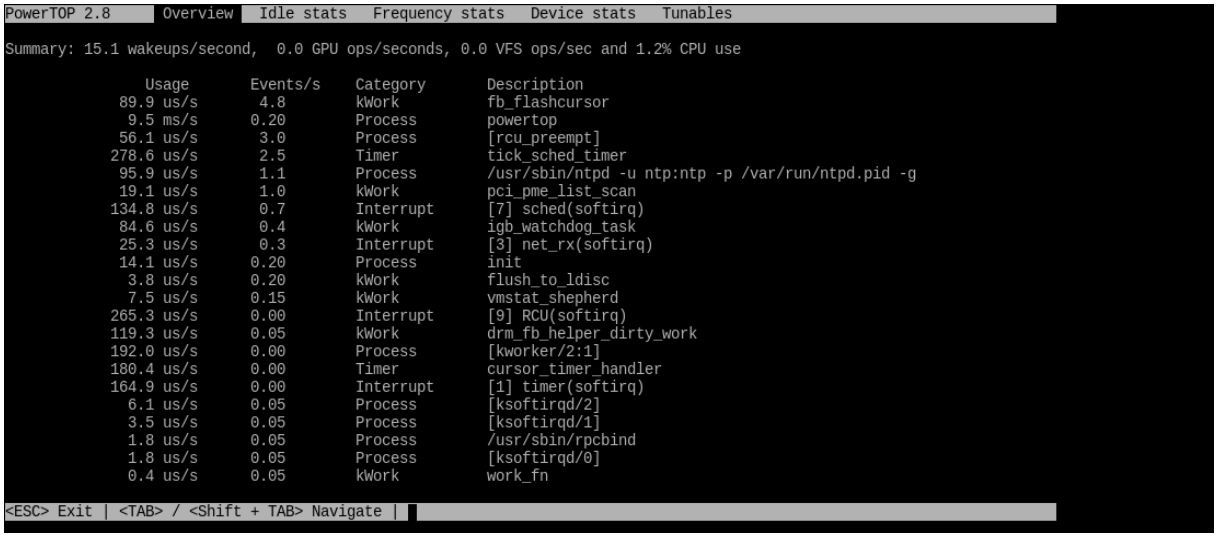
The CPU is mostly in the C6 state while CPU 3 is busy serving one application.

K.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 K.2. – powertop overview



L. Known Issues

L.1. LVDS Displays - REQ-00616

LVDS display support in linux is only supported since hardware revision 3 of the Q7-BT module.

L.2. kexec

Booting a 64bit kernel within Linux by the tool `kexec` is only supported if the executing kernel has the kernel command line options `video=efifb:off video=inteldrmfb:off` set.

L.3. Installation Of 64bit Images With .hddimg - REQ-00645

If a 64bit .hddimg (e.g. `msc-image-sato-valleyisland-64.hddimg` is used for installation to an SSD/SATA/SD/MMC disk, the installed image will not be bootable by the BIOS. This is due the partition table being created as an GPT and not an MBR table.

Workaround: Boot the .hddimg in legacy mode by pressing F10 within the BIOS and selecting the non-UEFI boot mode. Then an MBR partition table will be created by the installer.

L.4. HDMI Audio

The HDMI audio device is listed in kernel 3.14. But any output is not audible. This will be fixed when using the 3.17 kernel.

L.5. Standby And LVDS Display - REQ-00720

When resuming from standby, the backlight of the LVDS display is disabled.

Workaround: Set BIOS option `Chipset/Flat Panel Configuration/Backlight Control/PWM Control` to `EC`.

L.6. Standby And USB Device Support - REQ-00723

Resuming from standby might result in a deadlocked system if the board was used as an USB peripheral device.

Workaround: Don't go to standby if the board is used as an USB peripheral device.

L.7. USB Peripheral Drivers Must Be Reloaded Before Switching Gadget Driver - REQ-00890

When switching USB gadget drivers (g_zero, g_mass_storage), the USB peripheral drivers dwc3 and dwc3-pci must be reloaded, otherwise the USB device is not detected on the bus.

This appeared with linux 3.19 kernel.

Reload with: `rmmod dwc3 dwc3-pci; modprobe dwc3-pci`

M. Links

- MSC Technologies http://www.msc-technologies.eu
- Yocto project https://www.yoctoproject.org
- Yocto project documentation
  http://www.yoctoproject.org/docs/2.7/mega-manual/mega-manual.html
- LXQt http://lxqt.org
- Otter Browser https://otter-browser.org
- Intel Atom Processor E3800 Yocto BSP https://www.yoctoproject.org/downloads/bsps/daisy161/valley-island-64-bit
- Intel Atom Processor E3800 User Guide for Yocto Project BSP Graphics Driver https://www-ssl.intel.com/content/www/us/en/embedded/products/bay-trail/atom-e3800-celeron-n2xx-yocto-bsp-graphics-html
- Video Encoding on Intel Atom Processor E38XX http://www.intel.com/content/www/us/en/intelligent-systems/intel-embedded-media-and-graphics-driver/video-encode-atom-e38xx-emgd-gstreamer-paper.html

N. License Overview

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

Licenses		
((GPLv2+ & LGPLv2.1+) |(GPLv3+ & LGPLv3+))	(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+)	(MPL-2.0 & LGPL-2.1+)	AFL-2
Apache-2.0	Artistic-1.0	BSD
BSD-2-Clause	BSD-3-Clause	BSL-1.0
CC-BY-SA-3.0	CPL-1.0	Firmware-Abilis
Firmware-GPLv2	Firmware-IntcSST2	Firmware-Marvell
Firmware-OLPC	Firmware-adsp_sst	Firmware-agere
Firmware-amd-ucode	Firmware-amdgpu	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-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-with-OpenSSL-exception	GPL-3.0-with-GCC-exception	GPLv1
GPLv2	GPLv2+	GPLv2.0+
GPLv3	GPLv3+	ICU
ISC	Intel-Microcode-License	LGPL-2.0+
LGPL-2.1	LGPL-2.1+	LGPL-3.0
LGPLv2	LGPLv2+	LGPLv2.0+
LGPLv2.1	LGPLv2.1+	LGPLv3
LGPLv3+	Libpng	MIT
MIT-X	MIT-style	MPL-1
MPL-2.0	MPLv1.1	NTP
OFL-1.1	PD	PSF
PSFv2	Python-2.0	SGIv1
Sleepycat	The-Qt-Company-Commercial	WHENCE
Zlib	bzip2	openssl
zsh		

Table N.2. – License overview listing

Recipe	Version	Licenses
acl	2.2.52	GPLv2+
acl	*libacl*	
acpica	20180508	BSD, GPLv2
acpica		
acpid	2.0.31	GPLv2+
acpid		
adwaita-icon-theme	3.30.1	LGPL-3.0, CC-BY-SA-3.0
adwaita-icon-theme-symbolic		
alsa-lib	1.1.8	LGPLv2.1, GPLv2+
alsa-conf	*alsa-lib*	
alsa-plugins	1.1.8	LGPLv2.1, GPLv2+
alsa-plugins-pulseaudio-conf	*libasound-module-conf-pulse*	
libasound-module-ctl-pulse	*libasound-module-pcm-pulse*	
alsa-state	0.2.0	MIT
alsa-state	*alsa-states*	
alsa-utils	1.1.8	GPLv2+
alsa-utils	*alsa-utils-aconnect*	
alsa-utils-alsactl	*alsa-utils-alsaloop*	
alsa-utils-alsamixer	*alsa-utils-alsatplg*	
alsa-utils-alsaucm	*alsa-utils-amixer*	
alsa-utils-aplay	*alsa-utils-aseqdump*	
alsa-utils-aseqnet	*alsa-utils-iecset*	
alsa-utils-midi	*alsa-utils-speakertest*	
app-defaults	1.0	MIT
app-defaults		
at-spi2-atk	2.30.0	LGPLv2
at-spi2-atk		
at-spi2-core	2.30.0	LGPLv2
at-spi2-core	*at-spi2-core-locale-en-gb*	
atk	2.30.0	GPLv2+, LGPLv2+
atk	*atk-locale-en-gb*	
attr	2.4.47	LGPLv2.1+
libattr		
avahi	0.7	GPLv2+, LGPLv2.1+
avahi-daemon	*avahi-locale-en-gb*	
libavahi-client	*libavahi-common*	
libavahi-core		
base-files	3.0.14	GPLv2
base-files		
base-passwd	3.5.29	GPLv2+
base-passwd		
bash	4.4.18	GPLv3+
bash		
bc	1.07.1	GPLv3+
bc		
binutils	2.32.0	GPLv3
libbfd		
bluez5	5.50	GPLv2+, LGPLv2.1+
bluez5	*bluez5-noinst-tools*	
bluez5-obex		
bonnie++	1.03e	GPLv2
bonnie++	*bonnie++-scripts*	
boost	1.69.0	BSL-1.0, MIT, Python-2.0
boost-filesystem	*boost-iostreams*	
boost-regex		
bootlog	1.0	MIT
bootlog		

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

Recipe	Version	Licenses
busybox *busybox* *busybox-syslog*	1.30.1 *busybox-hwclock* *busybox-udhcpc*	GPLv2, bzip2
bzip2 *libbz2*	1.0.6	bzip2
ca-certificates *ca-certificates*	20190110	GPL-2.0+, MPL-2.0
cairo *cairo*	1.16.0 *cairo-gobject*	MPL-1, LGPLv2.1
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
compton *compton*	v0.1-beta2	NTP, MIT
compton-conf *compton-conf*	0.14.1 *compton-conf-locale-en-gb*	LGPLv2.1
connman *connman*	1.36 *connman-client*	GPLv2
connman-info *connman-info*	1.0	GPLv2
consolekit *consolekit*	0.4.6	GPLv2+
cpufrequtils *cpufrequtils*	008	GPLv2
cpupower *cpupower*	1.0	GPLv2
cryptsetup *cryptsetup*	2.1.0	GPL-2.0-with-OpenSSL-exception
cups *cups-lib*	2.2.10	GPLv2, LGPLv2
curl *libcurl*	7.64.0	MIT
db *db*	5.3.28	Sleepycat
dbus *dbus*	1.12.12 *dbus-lib*	AFL-2, GPLv2+
dbus-glib *dbus-glib*	0.110	AFL-2, GPLv2+
dialog *dialog*	1.3-20180621	LGPL-2.1
diffutils *diffutils*	3.7	GPLv3+
dnf *dnf*	4.1.0 *dnf-locale-en-gb*	GPLv2
dosfstools *dosfstools*	4.1	GPLv3
dropbear *dropbear*	2019.77	MIT, BSD-3-Clause, BSD-2-Clause, PD

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

Recipe	Version	Licenses
e2fsprogs	1.44.5	GPLv2, LGPLv2, BSD, MIT
e2fsprogs	*e2fsprogs-badblocks*	
e2fsprogs-e2fsck	*e2fsprogs-mke2fs*	
e2fsprogs-tune2fs	*libcomerr*	
libe2p	*libext2fs*	
libss		
efi-var	git	GPLv2
efi-var		
eject	2.1.5	GPLv2
eject		
elfutils	0.176	GPLv3+
elfutils	*libasm*	
libdw	*libelf*	
ethtool	4.19	GPLv2+
ethtool		
eudev	3.2.7	GPLv2.0+, LGPL-2.1+
eudev	*eudev-hwdb*	
libudev		
expat	2.2.6	MIT
expat		
file	5.36	BSD
file		
flac	1.3.2	GFDL-1.2, GPLv2+, LGPLv2.1+, BSD
libflac		
fontconfig	2.12.6	MIT-style, MIT, PD
fontconfig	*fontconfig-utils*	
formfactor	0.0	MIT
formfactor		
freetype	2.9.1	FreeType, GPLv2+
freetype		
fribidi	1.0.5	LGPLv2.1+
fribidi		
gcc-runtime	8.3.0	GPL-3.0-with-GCC-exception
libatomic	*libgomp*	
libstdc++		
gdbm	1.18.1	GPLv3
gdbm	*gdbm-compat*	
gdk-pixbuf	2.38.0	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	2.20.1	GPLv2
git		
glib-2.0	2.58.3	LGPLv2.1+, BSD, PD
glib-2.0	*glib-2.0-locale-en-gb*	
glib-2.0-utils		
glib-networking	2.60.1	LGPLv2.1
glib-networking	*glib-networking-locale-en-gb*	
glibc	2.29	GPLv2, LGPLv2.1
glibc		
glibc-locale	2.29	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	20190205	GPLv3+, SGIv1
glmark2		

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

Recipe	Version	Licenses
gmmlib *gmmlib*	18.4.1	MIT
gmp *gmp*	6.1.2	GPLv2+, LGPLv3+
gnupg *gnupg*	2.2.13	GPLv3, LGPLv3
gnutls *gnutls*	3.6.5	LGPLv2.1+
gobject-introspection *gobject-introspection*	1.58.3	LGPLv2+, GPLv2+
gottet *gottet*	1.1.7 *gottet-locale-en*	GPLv3
gpgme *gpgme*	1.12.0 *python3-gpg*	GPLv2+, LGPLv2.1+
grep *grep*	3.3	GPLv3
grub *grub-common*	2.02 *grub-editenv*	GPLv3
grub-bootconf *grub-bootconf*	1.00	MIT
grub-efi *grub-efi*	2.02	GPLv3
gst-examples *gst-examples*	0.0.1	LGPL-2.0+
gstreamer1.0 *gstreamer1.0*	1.14.4 *gstreamer1.0-locale-en-gb*	LGPLv2+
gstreamer1.0-libav *gstreamer1.0-libav*	1.14.4	((GPLv2+ & LGPLv2.1+) | (GPLv3+ & LGPLv3+)), GPLv2+, LGPLv2+
gstreamer1.0-plugins-bad	1.14.4	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-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-coloreffects*	
gstreamer1.0-plugins-bad-compositor	*gstreamer1.0-plugins-bad-curl*	
gstreamer1.0-plugins-bad-dashdemux	*gstreamer1.0-plugins-bad-debugutilsbad*	
gstreamer1.0-plugins-bad-decklink	*gstreamer1.0-plugins-bad-dtls*	
gstreamer1.0-plugins-bad-dvb	*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-openglmixers*	
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-rsvg*	

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

Recipe	Version	Licenses
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-stereo	*gstreamer1.0-plugins-bad-subenc*	
gstreamer1.0-plugins-bad-timecode	*gstreamer1.0-plugins-bad-ttmlsubs*	
gstreamer1.0-plugins-bad-uvch264	*gstreamer1.0-plugins-bad-vcdsrc*	
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	*gstreamer1.0-plugins-bad-yadif*	
libgstadaptivedemux-1.0	*libgstbadaudio-1.0*	
libgstbadvideo-1.0	*libgstbasecamerabinsrc-1.0*	
libgstcodecparsers-1.0	*libgstinsertbin-1.0*	
libgstisoff-1.0	*libgstmpegts-1.0*	
libgstphotography-1.0	*libgstplayer-1.0*	
libgsturidownloader-1.0	*libgstwayland-1.0*	
libgstwebrtc-1.0		
gstreamer1.0-plugins-base	1.14.4	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-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-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	1.14.4	GPLv2+, LGPLv2.1+
gstreamer1.0-plugins-good	*gstreamer1.0-plugins-good-alaw*	
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-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-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*	

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

Recipe	Version	Licenses
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	1.14.4	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	1.14.4	LGPLv2.1+
gstreamer1.0-vaapi		
gtk+3	3.24.5	LGPLv2, LGPLv2+, LGPLv2.1+
gtk+3	*gtk+3-locale-en*	
gtk+3-locale-en-gb		
gvfs	1.40.0	LGPLv2
gvfs	*gvfs-locale-en-gb*	
gvfsd-trash		
harfbuzz	2.3.1	MIT
harfbuzz		
hdparm	9.58	BSD
hdparm		
hicolor-icon-theme	0.17	GPLv2
hicolor-icon-theme		
hostapd	2.8	BSD-3-Clause
hostapd		
hsetroot	1.0.2	GPLv2
hsetroot		
htpdate	1.1.3	GPLv2
htpdate		
hunspell	1.7.0	GPLv2, LGPLv2.1
hunspell		
i2c-ids	1.0	MIT
i2c-ids		
i2c-tools	4.1	GPLv2+
i2c-tools		
i2c-write-read	git	GPLv2
i2c-write-read		
icu	63.1	ICU
libicudata	*libicui18n*	
libicuuc		
igt-gpu-tools	1.23	MIT-X, ISC
igt-gpu-tools		
imlib2	1.4.6	MIT, BSD
imlib2	*imlib2-loaders*	
init-ifupdown	1.0	GPLv2
init-ifupdown		
initscripts	1.0	GPLv2
initscripts	*initscripts-functions*	
intel-media-driver	18.4.1	MIT, BSD-3-Clause
intel-media-driver		
intel-microcode	20180807	Intel-Microcode-License
intel-microcode		
intel-vaapi-driver	2.3.0	MIT
intel-vaapi-driver		

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

Recipe	Version	Licenses
iperf3	3.6	BSD
iperf3		
iptables	1.6.2	GPLv2+
iptables	*iptables-module-ebt-802-3*	
iptables-module-ebt-ip	*iptables-module-ebt-log*	
iptables-module-ebt-mark-m	*iptables-module-ip6t-ah*	
iptables-module-ip6t-dnat	*iptables-module-ip6t-dnpt*	
iptables-module-ip6t-dst	*iptables-module-ip6t-eui64*	
iptables-module-ip6t-frag	*iptables-module-ip6t-hbh*	
iptables-module-ip6t-hl	*iptables-module-ip6t-icmp6*	
iptables-module-ip6t-ipv6header	*iptables-module-ip6t-log*	
iptables-module-ip6t-masquerade	*iptables-module-ip6t-mh*	
iptables-module-ip6t-netmap	*iptables-module-ip6t-redirect*	
iptables-module-ip6t-reject	*iptables-module-ip6t-rt*	
iptables-module-ip6t-snat	*iptables-module-ip6t-snpt*	
iptables-module-ip6t-srh	*iptables-module-ipt-ah*	
iptables-module-ipt-clusterip	*iptables-module-ipt-dnat*	
iptables-module-ipt-ecn	*iptables-module-ipt-icmp*	
iptables-module-ipt-log	*iptables-module-ipt-masquerade*	
iptables-module-ipt-netmap	*iptables-module-ipt-realm*	
iptables-module-ipt-redirect	*iptables-module-ipt-reject*	
iptables-module-ipt-snat	*iptables-module-ipt-ttl*	
iptables-module-ipt-ulog	*iptables-module-xt-addrtype*	
iptables-module-xt-audit	*iptables-module-xt-bpf*	
iptables-module-xt-cgroup	*iptables-module-xt-checksum*	
iptables-module-xt-classify	*iptables-module-xt-cluster*	
iptables-module-xt-comment	*iptables-module-xt-connbytes*	
iptables-module-xt-connlimit	*iptables-module-xt-connmark*	
iptables-module-xt-connsecmark	*iptables-module-xt-conntrack*	
iptables-module-xt-cpu	*iptables-module-xt-ct*	
iptables-module-xt-dccp	*iptables-module-xt-devgroup*	
iptables-module-xt-dscp	*iptables-module-xt-ecn*	
iptables-module-xt-esp	*iptables-module-xt-hashlimit*	
iptables-module-xt-helper	*iptables-module-xt-hmark*	
iptables-module-xt-idletimer	*iptables-module-xt-ipcomp*	
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-mangle	*iptables-module-xt-mark*	
iptables-module-xt-multiport	*iptables-module-xt-nfacct*	
iptables-module-xt-nflog	*iptables-module-xt-nfqueue*	
iptables-module-xt-notrack	*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-state*	
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	4.14	BSD
iw		
json-c	0.13.1	MIT
json-c		
kbd	2.0.4	GPLv2+
kbd	*kbd-keymaps*	
kexec-tools	2.0.19	GPLv2
kexec		
keymaps	1.0	GPLv2
keymaps		
kidletime	5.57.0	GPLv2, LGPLv2.1
kidletime		
kmod	26	GPL-2.0+, LGPL-2.1+

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

Recipe	Version	Licenses
kmod	*libkmod*	
kwayland *kwayland*	5.57.0	LGPLv2.1
kwindowsystem *kwindowsystem*	5.57.0 *kwindowsystem-locale-en-gb*	GPLv2, LGPLv2.1
l3afpad *l3afpad*	0.8.18.1.11	GPLv2+
lame *libmp3lame*	3.100	LGPLv2+
lib64-compatibility *lib64-compatibility*	1.0	MIT
liba52 *liba52*	0.7.4	GPLv2+
libaio *libaio*	0.3.111	LGPLv2.1+
libarchive *libarchive*	3.3.3	BSD
libassuan *libassuan*	2.5.3	LGPLv2.1+
libatasmart *libatasmart*	0.19	LGPLv2.1
libblockdev *libblockdev*	2.20	LGPLv2+
libbytesize *libbytesize*	1.4	LGPLv2+
libcap *libcap*	2.26	BSD, GPLv2
libcomps *libcomps*	0.1.10	GPLv2
libconfig *libconfig*	1.7.2	LGPLv2.1
libcroco *libcroco*	0.6.12	LGPLv2, LGPLv2.1
libdaemon *libdaemon*	0.14	LGPLv2.1+
libdbusmenu-qt5 *libdbusmenu-qt5*	0.9.3	GPLv2
libdevmapper *libdevmapper*	2.03.02	GPLv2, LGPLv2.1
libdmx *libdmx*	1.1.4	MIT
libdnf *libdnf*	0.26.0	LGPLv2.1
libdrm *libdrm* *libdrm-nouveau*	2.4.97 *libdrm-intel* *libdrm-radeon*	MIT
libepoxy *libepoxy*	1.5.3	MIT
liberation-fonts *liberation-fonts*	2.00.1	OFL-1.1
libevdev *libevdev*	1.6.0	MIT-X
libexif *libexif*	0.6.21 *libexif-locale-en-gb*	LGPLv2.1
libffi *libffi*	3.2.1	MIT
libfm-extra *libfm-extra*	1.3.1	LGPLv2+

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

Recipe	Version	Licenses
libfm-qt *libfm-qt*	0.14.1 *libfm-qt-locale-en-gb*	LGPLv2.1
libfontenc *libfontenc*	1.1.4	MIT
libgcc *libgcc*	8.3.0	GPL-3.0-with-GCC-exception
libgcrypt *libgcrypt*	1.8.4	LGPLv2.1+
libgpg-error *libgpg-error*	1.35	GPLv2+, LGPLv2.1+
libgphoto2 *libgphoto2* *libgphotoport*	2.5.17 *libgphoto2-camlibs*	LGPLv2.1
libgudev *libgudev*	232	LGPLv2.1
libical *libical*	2.0.0	LGPLv2.1, MPL-1
libice *libice*	1.0.9	MIT-style
libidn2 *libidn2*	2.0.5	(GPLv2+ | LGPLv3)
libinput *libinput*	1.12.6	MIT
libjpeg-turbo *libjpeg-turbo*	2.0.2	BSD-3-Clause
libksba *libksba*	1.3.5	GPLv2+, LGPLv3+, GPLv3+
libkscreen *libkscreen*	5.15.4	GPLv2
liblxqt *liblxqt*	0.14.1 *liblxqt-locale-en-gb*	LGPLv2.1
libmodulemd *libmodulemd*	2.1.0	MIT
libmscboost *libmscboost*	git	LGPLv2.1
libmscboostpython *libmscboostpython*	git	LGPLv2.1
libnl *libnl*	3.4.0 *libnl-genl*	LGPLv2.1
libnotify *libnotify*	0.7.7	LGPLv2.1
libnsl2 *libnsl2*	1.2.0	LGPL-2.1
libnss-mdns *libnss-mdns*	0.10	LGPLv2.1+
libogg *libogg*	1.3.3	BSD
libpcap *libpcap*	1.9.0	BSD
libpciaccess *libpciaccess*	0.14	MIT, MIT-style
libpcre *libpcre*	8.43	BSD
libpng *libpng*	1.6.36	Libpng

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

Recipe	Version	Licenses
libpsl *libpsl*	0.20.2	MIT
libqtxdg *libqtxdg*	3.3.1	LGPLv2.1
librepo *librepo*	1.9.4	LGPLv2.1
librsvg *librsvg*	2.40.20	LGPLv2+
libsamplerate0 *libsamplerate0*	0.1.9	BSD-2-Clause
libsdl *libsdl*	1.2.15	LGPLv2.1
libsecret *libsecret*	0.18.7 *libsecret-locale-en-gb*	LGPLv2.1
libsm *libsm*	1.2.3	MIT-style
libsndfile1 *libsndfile1*	1.0.28	LGPLv2.1
libsolv *libsolv*	0.7.3 *libsolvext*	BSD-3-Clause
libsoup-2.4 *libsoup-2.4*	2.64.2 *libsoup-2.4-locale-en-gb*	LGPLv2
libstatgrab *libstatgrab*	0.91	MIT
libsysstat *libsysstat*	0.4.2	LGPLv2.1
libtheora *libtheora*	1.1.1	BSD
libtirpc *libtirpc*	1.0.3	BSD
libtool *libltdl*	2.4.6	GPLv2, LGPLv2.1
libunistring *libunistring*	0.9.10	LGPLv3+, GPLv2
libunwind *libunwind*	1.3.1	MIT
libusb1 *libusb1*	1.0.22	LGPLv2.1+
libva *libva* *libva-x11*	2.4.0 *libva-wayland*	MIT
libva-utils *libva-utils*	2.4.0	MIT
libvorbis *libvorbis*	1.3.6	BSD
libvpx *libvpx*	1.7.0	BSD
libwebp *libwebp*	1.0.2	BSD
libx11 *libx11* *libx11-xcb*	1.6.7 *libx11-locale*	MIT, MIT-style, BSD
libxau *libxau*	1.0.9	MIT-style
libxaw *libxaw7*	1.0.13	MIT-X
libxcb *libxcb* *libxcb-dri3*	1.13.1 *libxcb-dri2* *libxcb-glx*	MIT

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

Recipe	Version	Licenses
libxcb-present	*libxcb-randr*	
libxcb-render	*libxcb-shape*	
libxcb-shm	*libxcb-sync*	
libxcb-xfixes	*libxcb-xinerama*	
libxcb-xinput	*libxcb-xkb*	
libxcomposite	0.4.5	MIT-style
libxcomposite		
libxcrypt	4.4.2	LGPLv2.1
libxcrypt		
libxcursor	1.2.0	MIT-style
libxcursor		
libxdamage	1.1.5	MIT
libxdamage		
libxdmcp	1.1.2	MIT-style
libxdmcp		
libxext	1.3.3	MIT-style
libxext		
libxfce4ui	4.13.4	GPLv2
libxfce4ui	*libxfce4ui-gtk3*	
libxfce4ui-locale-en-gb		
libxfce4util	4.13.2	GPLv2
libxfce4util	*libxfce4util-locale-en-gb*	
libxfixes	5.0.3	MIT-style
libxfixes		
libxfont2	2.0.3	MIT, MIT-style, BSD
libxfont2		
libxft	2.3.2	MIT
libxft		
libxi	1.7.9	MIT, MIT-style
libxi		
libxinerama	1.1.4	MIT
libxinerama		
libxkbcommon	0.8.4	MIT, MIT-style
libxkbcommon		
libxkbfile	1.0.9	MIT-style
libxkbfile		
libxml2	2.9.8	MIT
libxml2	*libxml2-utils*	
libxmu	1.1.2	MIT, MIT-style
libxmu	*libxmuu*	
libxpm	3.5.12	BSD
libxpm		
libxrandr	1.5.1	MIT-style
libxrandr		
libxrender	0.9.10	MIT-style
libxrender		
libxscrnsaver	1.2.3	MIT
libxscrnsaver		
libxshmfence	1.3	MIT-style
libxshmfence		
libxslt	1.1.33	MIT
libxslt		
libxt	1.1.5	MIT, MIT-style
libxt		
libxtst	1.2.3	MIT-style
libxtst		
libxv	1.0.11	MIT-style
libxv		
libxvmc	1.0.10	MIT

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

Recipe	Version	Licenses
libxvmc		
libxxf86dga *libxxf86dga*	1.1.4	MIT-X
libxxf86misc *libxxf86misc*	1.0.4	MIT
libxxf86vm *libxxf86vm*	1.1.4	MIT
libyaml *libyaml*	0.2.1	MIT
linux-firmware	0.0	Firmware-Abilis, Firmware-agere, Firmware-amdgpu, Firmware-amd-ucode, 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-ar3k*	
linux-firmware-ar3k-license	*linux-firmware-ar9170*	
linux-firmware-ath10k	*linux-firmware-ath10k-license*	
linux-firmware-ath6k	*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*	

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

Recipe	Version	Licenses
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-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-hw-37-7	*linux-firmware-ibt-hw-37-8*	
linux-firmware-ibt-license	*linux-firmware-ibt-misc*	
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-marvell-license	*linux-firmware-mt7601u*	
linux-firmware-mt7601u-license	*linux-firmware-netronome*	
linux-firmware-netronome-license	*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-license	*linux-firmware-qcom-venus-1.8*	
linux-firmware-qcom-venus-4.2	*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-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-intel	4.19.34	GPLv2
kernel	*kernel-base*	
kernel-image	*kernel-image-bzimage*	
kernel-module-8021q-4.19.34-intel-pk-standard	*kernel-module-8250-exar-4.19.34-intel-pk-standard*	
kernel-module-ac97-bus-4.19.34-intel-pk-standard	*kernel-module-acpi-thermal-rel-4.19.34-intel-pk-standard*	
kernel-module-act-mirred-4.19.34-intel-pk-standard	*kernel-module-af-key-4.19.34-intel-pk-standard*	
kernel-module-anubis-4.19.34-intel-pk-standard	*kernel-module-arc4-4.19.34-intel-pk-standard*	
kernel-module-arp-tables-4.19.34-intel-pk-standard	*kernel-module-arpt-mangle-4.19.34-intel-pk-standard*	
kernel-module-arptable-filter-4.19.34-intel-pk-standard	*kernel-module-asix-4.19.34-intel-pk-standard*	
kernel-module-at24-4.19.34-intel-pk-standard	*kernel-module-at25-4.19.34-intel-pk-standard*	
kernel-module-ath-4.19.34-intel-pk-standard	*kernel-module-ath5k-4.19.34-intel-pk-standard*	
kernel-module-ath9k-4.19.34-intel-pk-standard	*kernel-module-ath9k-common-4.19.34-intel-pk-standard*	
kernel-module-ath9k-hw-4.19.34-intel-pk-standard	*kernel-module-atkbd-4.19.34-intel-pk-standard*	
kernel-module-ax88179-178a-4.19.34-intel-pk-standard	*kernel-module-baytrail-edac-4.19.34-intel-pk-standard*	
kernel-module-bcma-4.19.34-intel-pk-standard	*kernel-module-binfmt-misc-4.19.34-intel-pk-standard*	
kernel-module-blowfish-common-4.19.34-intel-pk-standard	*kernel-module-blowfish-generic-4.19.34-intel-pk-standard*	
kernel-module-bnep-4.19.34-intel-pk-standard	*kernel-module-bonding-4.19.34-intel-pk-standard*	
kernel-module-bq25890-charger-4.19.34-intel-pk-standard	*kernel-module-brcmfmac-4.19.34-intel-pk-standard*	
kernel-module-brcmsmac-4.19.34-intel-pk-standard	*kernel-module-brcmutil-4.19.34-intel-pk-standard*	

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

Recipe	Version	Licenses
kernel-module-bridge-4.19.34-intel-pk-standard		*kernel-module-btbcm-4.19.34-intel-pk-standard*
kernel-module-btintel-4.19.34-intel-pk-standard		*kernel-module-btmrvl-4.19.34-intel-pk-standard*
kernel-module-btmrvl-sdio-4.19.34-intel-pk-standard		*kernel-module-btrtl-4.19.34-intel-pk-standard*
kernel-module-btusb-4.19.34-intel-pk-standard		*kernel-module-camellia-generic-4.19.34-intel-pk-standard*
kernel-module-can-4.19.34-intel-pk-standard		*kernel-module-can-bcm-4.19.34-intel-pk-standard*
kernel-module-can-dev-4.19.34-intel-pk-standard		*kernel-module-can-gw-4.19.34-intel-pk-standard*
kernel-module-can-raw-4.19.34-intel-pk-standard		*kernel-module-cast-common-4.19.34-intel-pk-standard*
kernel-module-cast5-generic-4.19.34-intel-pk-standard		*kernel-module-cast6-generic-4.19.34-intel-pk-standard*
kernel-module-ccm-4.19.34-intel-pk-standard		*kernel-module-cdc-acm-4.19.34-intel-pk-standard*
kernel-module-cdc-eem-4.19.34-intel-pk-standard		*kernel-module-cdc-ether-4.19.34-intel-pk-standard*
kernel-module-cdc-ncm-4.19.34-intel-pk-standard		*kernel-module-cdc-subset-4.19.34-intel-pk-standard*
kernel-module-cdc-wdm-4.19.34-intel-pk-standard		*kernel-module-cfg80211-4.19.34-intel-pk-standard*
kernel-module-chipreg-4.19.34-intel-pk-standard		*kernel-module-cifs-4.19.34-intel-pk-standard*
kernel-module-cirrus-4.19.34-intel-pk-standard		*kernel-module-cmdlinepart-4.19.34-intel-pk-standard*
kernel-module-configfs-4.19.34-intel-pk-standard		*kernel-module-cordic-4.19.34-intel-pk-standard*
kernel-module-coretemp-4.19.34-intel-pk-standard		*kernel-module-crc-ccitt-4.19.34-intel-pk-standard*
kernel-module-crc-itu-t-4.19.34-intel-pk-standard		*kernel-module-cryptoloop-4.19.34-intel-pk-standard*
kernel-module-cuse-4.19.34-intel-pk-standard		*kernel-module-deflate-4.19.34-intel-pk-standard*
kernel-module-diag-4.19.34-intel-pk-standard		*kernel-module-dm9601-4.19.34-intel-pk-standard*
kernel-module-dummy-4.19.34-intel-pk-standard		*kernel-module-dummy-hcd-4.19.34-intel-pk-standard*
kernel-module-dw-dmac-4.19.34-intel-pk-standard		*kernel-module-dwc2-4.19.34-intel-pk-standard*
kernel-module-dwc2-pci-4.19.34-intel-pk-standard		*kernel-module-dwc3-4.19.34-intel-pk-standard*
kernel-module-dwc3-haps-4.19.34-intel-pk-standard		*kernel-module-dwc3-pci-4.19.34-intel-pk-standard*
kernel-module-e100-4.19.34-intel-pk-standard		*kernel-module-e1000-4.19.34-intel-pk-standard*
kernel-module-e1000e-4.19.34-intel-pk-standard		*kernel-module-e752x-edac-4.19.34-intel-pk-standard*
kernel-module-ebt-802-3-4.19.34-intel-pk-standard		*kernel-module-ebt-among-4.19.34-intel-pk-standard*
kernel-module-ebt-arp-4.19.34-intel-pk-standard		*kernel-module-ebt-arpreply-4.19.34-intel-pk-standard*
kernel-module-ebt-dnat-4.19.34-intel-pk-standard		*kernel-module-ebt-ip-4.19.34-intel-pk-standard*
kernel-module-ebt-limit-4.19.34-intel-pk-standard		*kernel-module-ebt-log-4.19.34-intel-pk-standard*
kernel-module-ebt-mark-4.19.34-intel-pk-standard		*kernel-module-ebt-mark-m-4.19.34-intel-pk-standard*
kernel-module-ebt-nflog-4.19.34-intel-pk-standard		*kernel-module-ebt-pkttype-4.19.34-intel-pk-standard*
kernel-module-ebt-redirect-4.19.34-intel-pk-standard		*kernel-module-ebt-snat-4.19.34-intel-pk-standard*
kernel-module-ebt-stp-4.19.34-intel-pk-standard		*kernel-module-ebt-vlan-4.19.34-intel-pk-standard*
kernel-module-ebtable-broute-4.19.34-intel-pk-standard		*kernel-module-ebtable-filter-4.19.34-intel-pk-standard*
kernel-module-ebtable-nat-4.19.34-intel-pk-standard		*kernel-module-ebtables-4.19.34-intel-pk-standard*
kernel-module-echainiv-4.19.34-intel-pk-standard		*kernel-module-eeprom-93cx6-4.19.34-intel-pk-standard*
kernel-module-efivarfs-4.19.34-intel-pk-standard		*kernel-module-efivars-4.19.34-intel-pk-standard*
kernel-module-ftdi-sio-4.19.34-intel-pk-standard		*kernel-module-fuse-4.19.34-intel-pk-standard*
kernel-module-g-acm-ms-4.19.34-intel-pk-standard		*kernel-module-g-audio-4.19.34-intel-pk-standard*
kernel-module-g-cdc-4.19.34-intel-pk-standard		*kernel-module-g-dbgp-4.19.34-intel-pk-standard*
kernel-module-g-ether-4.19.34-intel-pk-standard		*kernel-module-g-ffs-4.19.34-intel-pk-standard*
kernel-module-g-hid-4.19.34-intel-pk-standard		*kernel-module-g-mass-storage-4.19.34-intel-pk-standard*
kernel-module-g-midi-4.19.34-intel-pk-standard		*kernel-module-g-multi-4.19.34-intel-pk-standard*
kernel-module-g-ncm-4.19.34-intel-pk-standard		*kernel-module-g-printer-4.19.34-intel-pk-standard*
kernel-module-g-serial-4.19.34-intel-pk-standard		*kernel-module-g-zero-4.19.34-intel-pk-standard*
kernel-module-gadgetfs-4.19.34-intel-pk-standard		*kernel-module-gcm-4.19.34-intel-pk-standard*
kernel-module-gf128mul-4.19.34-intel-pk-standard		*kernel-module-ghash-generic-4.19.34-intel-pk-standard*
kernel-module-gpio-exar-4.19.34-intel-pk-standard		*kernel-module-gpio-keys-4.19.34-intel-pk-standard*
kernel-module-gpio-wcove-4.19.34-intel-pk-standard		*kernel-module-hci-uart-4.19.34-intel-pk-standard*
kernel-module-hid-sensor-hub-4.19.34-intel-pk-standard		*kernel-module-hidp-4.19.34-intel-pk-standard*
kernel-module-i2c-designware-pci-4.19.34-intel-pk-standard		*kernel-module-i2c-i801-4.19.34-intel-pk-standard*
kernel-module-i2c-ismt-4.19.34-intel-pk-standard		*kernel-module-i2c-smbus-4.19.34-intel-pk-standard*
kernel-module-i3000-edac-4.19.34-intel-pk-standard		*kernel-module-i3200-edac-4.19.34-intel-pk-standard*
kernel-module-i5000-edac-4.19.34-intel-pk-standard		*kernel-module-i5100-edac-4.19.34-intel-pk-standard*
kernel-module-i5400-edac-4.19.34-intel-pk-standard		*kernel-module-i7300-edac-4.19.34-intel-pk-standard*
kernel-module-i7core-edac-4.19.34-intel-pk-standard		*kernel-module-i8042-4.19.34-intel-pk-standard*
kernel-module-i82975x-edac-4.19.34-intel-pk-standard		*kernel-module-i915-4.19.34-intel-pk-standard*
kernel-module-igb-4.19.34-intel-pk-standard		*kernel-module-igbvf-4.19.34-intel-pk-standard*
kernel-module-input-polldev-4.19.34-intel-pk-standard		*kernel-module-int3400-thermal-4.19.34-intel-pk-standard*
kernel-module-int3402-thermal-4.19.34-intel-pk-standard		*kernel-module-int3403-thermal-4.19.34-intel-pk-standard*
kernel-module-int340x-thermal-zone-4.19.34-intel-pk-standard		*kernel-module-intel-bxt-pmic-thermal-4.19.34-intel-pk-standard*
kernel-module-intel-lpss-4.19.34-intel-pk-standard		*kernel-module-intel-lpss-acpi-4.19.34-intel-pk-standard*
kernel-module-intel-lpss-pci-4.19.34-intel-pk-standard		*kernel-module-intel-pch-thermal-4.19.34-intel-pk-standard*
kernel-module-intel-powerclamp-4.19.34-intel-pk-standard		*kernel-module-intel-rapl-4.19.34-intel-pk-standard*
kernel-module-intel-soc-dts-iosf-4.19.34-intel-pk-standard		*kernel-module-intel-soc-dts-thermal-4.19.34-intel-pk-standard*
kernel-module-ip-tables-4.19.34-intel-pk-standard		*kernel-module-ip-tunnel-4.19.34-intel-pk-standard*
kernel-module-ipt-ah-4.19.34-intel-pk-standard		*kernel-module-ipt-clusterip-4.19.34-intel-pk-standard*
kernel-module-ipt-ecn-4.19.34-intel-pk-standard		*kernel-module-ipt-masquerade-4.19.34-intel-pk-standard*
kernel-module-ipt-reject-4.19.34-intel-pk-standard		*kernel-module-ipt-rpfilter-4.19.34-intel-pk-standard*
kernel-module-ipt-synproxy-4.19.34-intel-pk-standard		*kernel-module-iptable-filter-4.19.34-intel-pk-standard*
kernel-module-iptable-mangle-4.19.34-intel-pk-standard		*kernel-module-iptable-nat-4.19.34-intel-pk-standard*
kernel-module-iptable-raw-4.19.34-intel-pk-standard		*kernel-module-iptable-security-4.19.34-intel-pk-standard*

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

Recipe	Version	Licenses
kernel-module-itco-wdt-4.19.34-intel-pk-standard		*kernel-module-iwl3945-4.19.34-intel-pk-standard*
kernel-module-iwl4965-4.19.34-intel-pk-standard		*kernel-module-iwldvm-4.19.34-intel-pk-standard*
kernel-module-iwlegacy-4.19.34-intel-pk-standard		*kernel-module-iwlmvm-4.19.34-intel-pk-standard*
kernel-module-iwlwifi-4.19.34-intel-pk-standard		*kernel-module-jme-4.19.34-intel-pk-standard*
kernel-module-kaweth-4.19.34-intel-pk-standard		*kernel-module-khazad-4.19.34-intel-pk-standard*
kernel-module-lcd-4.19.34-intel-pk-standard		*kernel-module-libcomposite-4.19.34-intel-pk-standard*
kernel-module-libcrc32c-4.19.34-intel-pk-standard		*kernel-module-libphy-4.19.34-intel-pk-standard*
kernel-module-libps2-4.19.34-intel-pk-standard		*kernel-module-llc-4.19.34-intel-pk-standard*
kernel-module-lm75-4.19.34-intel-pk-standard		*kernel-module-lp-4.19.34-intel-pk-standard*
kernel-module-lpc-ich-4.19.34-intel-pk-standard		*kernel-module-m25p80-4.19.34-intel-pk-standard*
kernel-module-mac80211-4.19.34-intel-pk-standard		*kernel-module-mce-inject-4.19.34-intel-pk-standard*
kernel-module-mcs7830-4.19.34-intel-pk-standard		*kernel-module-md4-4.19.34-intel-pk-standard*
kernel-module-mei-4.19.34-intel-pk-standard		*kernel-module-mei-me-4.19.34-intel-pk-standard*
kernel-module-mei-txe-4.19.34-intel-pk-standard		*kernel-module-mei-wdt-4.19.34-intel-pk-standard*
kernel-module-michael-mic-4.19.34-intel-pk-standard		*kernel-module-mii-4.19.34-intel-pk-standard*
kernel-module-mt7601u-4.19.34-intel-pk-standard		*kernel-module-mtd-4.19.34-intel-pk-standard*
kernel-module-mwifiex-4.19.34-intel-pk-standard		*kernel-module-mwifiex-sdio-4.19.34-intel-pk-standard*
kernel-module-nbd-4.19.34-intel-pk-standard		*kernel-module-net1080-4.19.34-intel-pk-standard*
kernel-module-net2280-4.19.34-intel-pk-standard		*kernel-module-nf-conncount-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-4.19.34-intel-pk-standard		*kernel-module-nf-conntrack-amanda-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-broadcast-4.19.34-intel-pk-standard		*kernel-module-nf-conntrack-ftp-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-h323-4.19.34-intel-pk-standard		*kernel-module-nf-conntrack-irc-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-netbios-ns-4.19.34-intel-pk-standard		*kernel-module-nf-conntrack-netlink-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-pptp-4.19.34-intel-pk-standard		*kernel-module-nf-conntrack-proto-gre-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-sane-4.19.34-intel-pk-standard		*kernel-module-nf-conntrack-sip-4.19.34-intel-pk-standard*
kernel-module-nf-conntrack-tftp-4.19.34-intel-pk-standard		*kernel-module-nf-defrag-ipv4-4.19.34-intel-pk-standard*
kernel-module-nf-log-common-4.19.34-intel-pk-standard		*kernel-module-nf-log-ipv4-4.19.34-intel-pk-standard*
kernel-module-nf-nat-4.19.34-intel-pk-standard		*kernel-module-nf-nat-amanda-4.19.34-intel-pk-standard*
kernel-module-nf-nat-ftp-4.19.34-intel-pk-standard		*kernel-module-nf-nat-h323-4.19.34-intel-pk-standard*
kernel-module-nf-nat-ipv4-4.19.34-intel-pk-standard		*kernel-module-nf-nat-irc-4.19.34-intel-pk-standard*
kernel-module-nf-nat-pptp-4.19.34-intel-pk-standard		*kernel-module-nf-nat-proto-gre-4.19.34-intel-pk-standard*
kernel-module-nf-nat-sip-4.19.34-intel-pk-standard		*kernel-module-nf-nat-tftp-4.19.34-intel-pk-standard*
kernel-module-nf-reject-ipv4-4.19.34-intel-pk-standard		*kernel-module-nf-synproxy-core-4.19.34-intel-pk-standard*
kernel-module-nfnetlink-4.19.34-intel-pk-standard		*kernel-module-nfnetlink-log-4.19.34-intel-pk-standard*
kernel-module-nfnetlink-queue-4.19.34-intel-pk-standard		*kernel-module-nls-cp850-4.19.34-intel-pk-standard*
kernel-module-nls-utf8-4.19.34-intel-pk-standard		*kernel-module-p8022-4.19.34-intel-pk-standard*
kernel-module-parport-4.19.34-intel-pk-standard		*kernel-module-parport-pc-4.19.34-intel-pk-standard*
kernel-module-pcspkr-4.19.34-intel-pk-standard		*kernel-module-pegasus-4.19.34-intel-pk-standard*
kernel-module-pktgen-4.19.34-intel-pk-standard		*kernel-module-pl2303-4.19.34-intel-pk-standard*
kernel-module-pnd2-edac-4.19.34-intel-pk-standard		*kernel-module-portmux-intel-drcfg-4.19.34-intel-pk-standard*
kernel-module-powr1220-4.19.34-intel-pk-standard		*kernel-module-ppdev-4.19.34-intel-pk-standard*
kernel-module-processor-thermal-device-4.19.34-intel-pk-standard		*kernel-module-psmouse-4.19.34-intel-pk-standard*
kernel-module-psnap-4.19.34-intel-pk-standard		*kernel-module-pwm-lpss-4.19.34-intel-pk-standard*
kernel-module-pwm-lpss-pci-4.19.34-intel-pk-standard		*kernel-module-pwm-lpss-platform-4.19.34-intel-pk-standard*
kernel-module-r8152-4.19.34-intel-pk-standard		*kernel-module-radeon-4.19.34-intel-pk-standard*
kernel-module-rfcomm-4.19.34-intel-pk-standard		*kernel-module-roles-4.19.34-intel-pk-standard*
kernel-module-rpcsec-gss-krb5-4.19.34-intel-pk-standard		*kernel-module-rt2800lib-4.19.34-intel-pk-standard*
kernel-module-rt2800mmio-4.19.34-intel-pk-standard		*kernel-module-rt2800pci-4.19.34-intel-pk-standard*
kernel-module-rt2x00lib-4.19.34-intel-pk-standard		*kernel-module-rt2x00mmio-4.19.34-intel-pk-standard*
kernel-module-rt2x00pci-4.19.34-intel-pk-standard		*kernel-module-rtc-rv8803-4.19.34-intel-pk-standard*
kernel-module-rtl8150-4.19.34-intel-pk-standard		*kernel-module-sch-cbq-4.19.34-intel-pk-standard*
kernel-module-sch-codel-4.19.34-intel-pk-standard		*kernel-module-sch-dsmark-4.19.34-intel-pk-standard*
kernel-module-sch-gred-4.19.34-intel-pk-standard		*kernel-module-sch-hfsc-4.19.34-intel-pk-standard*
kernel-module-sch-htb-4.19.34-intel-pk-standard		*kernel-module-sch-ingress-4.19.34-intel-pk-standard*
kernel-module-sch-netem-4.19.34-intel-pk-standard		*kernel-module-sch-prio-4.19.34-intel-pk-standard*
kernel-module-sch-red-4.19.34-intel-pk-standard		*kernel-module-sch-sfq-4.19.34-intel-pk-standard*
kernel-module-sch-tbf-4.19.34-intel-pk-standard		*kernel-module-sch-teql-4.19.34-intel-pk-standard*
kernel-module-serio-4.19.34-intel-pk-standard		*kernel-module-serpent-generic-4.19.34-intel-pk-standard*
kernel-module-sha1-generic-4.19.34-intel-pk-standard		*kernel-module-sha512-generic-4.19.34-intel-pk-standard*
kernel-module-skx-edac-4.19.34-intel-pk-standard		*kernel-module-slhc-4.19.34-intel-pk-standard*
kernel-module-slip-4.19.34-intel-pk-standard		*kernel-module-smsc75xx-4.19.34-intel-pk-standard*
kernel-module-smsc95xx-4.19.34-intel-pk-standard		*kernel-module-snd-ac97-codec-4.19.34-intel-pk-standard*
kernel-module-snd-ak4113-4.19.34-intel-pk-standard		*kernel-module-snd-ak4114-4.19.34-intel-pk-standard*
kernel-module-snd-ak4xxx-adda-4.19.34-intel-pk-standard		*kernel-module-snd-aloop-4.19.34-intel-pk-standard*
kernel-module-snd-atiixp-4.19.34-intel-pk-standard		*kernel-module-snd-atiixp-modem-4.19.34-intel-pk-standard*
kernel-module-snd-ca0106-4.19.34-intel-pk-standard		*kernel-module-snd-cmipci-4.19.34-intel-pk-standard*
kernel-module-snd-cs8427-4.19.34-intel-pk-standard		*kernel-module-snd-ctxfi-4.19.34-intel-pk-standard*
kernel-module-snd-emu10k1-4.19.34-intel-pk-standard		*kernel-module-snd-emu10k1-synth-4.19.34-intel-pk-standard*
kernel-module-snd-emu10k1x-4.19.34-intel-pk-standard		*kernel-module-snd-emux-synth-4.19.34-intel-pk-standard*
kernel-module-snd-hda-codec-4.19.34-intel-pk-standard		*kernel-module-snd-hda-codec-cirrus-4.19.34-intel-pk-standard*
kernel-module-snd-hda-codec-generic-4.19.34-intel-pk-standard		*kernel-module-snd-hda-codec-hdmi-4.19.34-intel-pk-standard*
kernel-module-snd-hda-codec-realtek-4.19.34-intel-pk-standard		*kernel-module-snd-hda-codec-via-4.19.34-intel-pk-standard*

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

Recipe	Version	Licenses
kernel-module-snd-hda-core-4.19.34-intel-pk-standard		*kernel-module-snd-hda-intel-4.19.34-intel-pk-standard*
kernel-module-snd-hdsp-4.19.34-intel-pk-standard		*kernel-module-snd-hdspm-4.19.34-intel-pk-standard*
kernel-module-snd-hrtimer-4.19.34-intel-pk-standard		*kernel-module-snd-hwdep-4.19.34-intel-pk-standard*
kernel-module-snd-i2c-4.19.34-intel-pk-standard		*kernel-module-snd-ice1712-4.19.34-intel-pk-standard*
kernel-module-snd-ice1724-4.19.34-intel-pk-standard		*kernel-module-snd-ice17xx-ak4xxx-4.19.34-intel-pk-standard*
kernel-module-snd-mpu401-uart-4.19.34-intel-pk-standard		*kernel-module-snd-opl3-lib-4.19.34-intel-pk-standard*
kernel-module-snd-opl3-synth-4.19.34-intel-pk-standard		*kernel-module-snd-oxygen-4.19.34-intel-pk-standard*
kernel-module-snd-oxygen-lib-4.19.34-intel-pk-standard		*kernel-module-snd-pcm-4.19.34-intel-pk-standard*
kernel-module-snd-pt2258-4.19.34-intel-pk-standard		*kernel-module-snd-rawmidi-4.19.34-intel-pk-standard*
kernel-module-snd-rme96-4.19.34-intel-pk-standard		*kernel-module-snd-rme9652-4.19.34-intel-pk-standard*
kernel-module-snd-seq-4.19.34-intel-pk-standard		*kernel-module-snd-seq-device-4.19.34-intel-pk-standard*
kernel-module-snd-seq-midi-4.19.34-intel-pk-standard		*kernel-module-snd-seq-midi-emul-4.19.34-intel-pk-standard*
kernel-module-snd-seq-midi-event-4.19.34-intel-pk-standard		*kernel-module-snd-seq-virmidi-4.19.34-intel-pk-standard*
kernel-module-snd-timer-4.19.34-intel-pk-standard		*kernel-module-snd-util-mem-4.19.34-intel-pk-standard*
kernel-module-snd-virtuoso-4.19.34-intel-pk-standard		*kernel-module-soc-button-array-4.19.34-intel-pk-standard*
kernel-module-softdog-4.19.34-intel-pk-standard		*kernel-module-spi-nor-4.19.34-intel-pk-standard*
kernel-module-spidev-4.19.34-intel-pk-standard		*kernel-module-stp-4.19.34-intel-pk-standard*
kernel-module-tcpm-4.19.34-intel-pk-standard		*kernel-module-tcrypt-4.19.34-intel-pk-standard*
kernel-module-tea-4.19.34-intel-pk-standard		*kernel-module-tipc-4.19.34-intel-pk-standard*
kernel-module-tpm-4.19.34-intel-pk-standard		*kernel-module-tpm-atmel-4.19.34-intel-pk-standard*
kernel-module-tpm-crb-4.19.34-intel-pk-standard		*kernel-module-tpm-infineon-4.19.34-intel-pk-standard*
kernel-module-tpm-nsc-4.19.34-intel-pk-standard		*kernel-module-tpm-tis-4.19.34-intel-pk-standard*
kernel-module-tpm-tis-core-4.19.34-intel-pk-standard		*kernel-module-tpm-vtpm-proxy-4.19.34-intel-pk-standard*
kernel-module-ts-bm-4.19.34-intel-pk-standard		*kernel-module-ts-fsm-4.19.34-intel-pk-standard*
kernel-module-ts-kmp-4.19.34-intel-pk-standard		*kernel-module-tun-4.19.34-intel-pk-standard*
kernel-module-twofish-common-4.19.34-intel-pk-standard		*kernel-module-twofish-generic-4.19.34-intel-pk-standard*
kernel-module-typec-4.19.34-intel-pk-standard		*kernel-module-typec-wcove-4.19.34-intel-pk-standard*
kernel-module-u-audio-4.19.34-intel-pk-standard		*kernel-module-u-ether-4.19.34-intel-pk-standard*
kernel-module-u-serial-4.19.34-intel-pk-standard		*kernel-module-udp-tunnel-4.19.34-intel-pk-standard*
kernel-module-usb-f-acm-4.19.34-intel-pk-standard		*kernel-module-usb-f-ecm-4.19.34-intel-pk-standard*
kernel-module-usb-f-ecm-subset-4.19.34-intel-pk-standard		*kernel-module-usb-f-fs-4.19.34-intel-pk-standard*
kernel-module-usb-f-hid-4.19.34-intel-pk-standard		*kernel-module-usb-f-mass-storage-4.19.34-intel-pk-standard*
kernel-module-usb-f-midi-4.19.34-intel-pk-standard		*kernel-module-usb-f-ncm-4.19.34-intel-pk-standard*
kernel-module-usb-f-obex-4.19.34-intel-pk-standard		*kernel-module-usb-f-printer-4.19.34-intel-pk-standard*
kernel-module-usb-f-rndis-4.19.34-intel-pk-standard		*kernel-module-usb-f-serial-4.19.34-intel-pk-standard*
kernel-module-usb-f-ss-lb-4.19.34-intel-pk-standard		*kernel-module-usb-f-uac2-4.19.34-intel-pk-standard*
kernel-module-usblp-4.19.34-intel-pk-standard		*kernel-module-usbnet-4.19.34-intel-pk-standard*
kernel-module-usbserial-4.19.34-intel-pk-standard		*kernel-module-usbtest-4.19.34-intel-pk-standard*
kernel-module-usbtouchscreen-4.19.34-intel-pk-standard		*kernel-module-uvesafb-4.19.34-intel-pk-standard*
kernel-module-video-4.19.34-intel-pk-standard		*kernel-module-virtio-input-4.19.34-intel-pk-standard*
kernel-module-wp512-4.19.34-intel-pk-standard		*kernel-module-x-tables-4.19.34-intel-pk-standard*
kernel-module-x38-edac-4.19.34-intel-pk-standard		*kernel-module-x86-pkg-temp-thermal-4.19.34-intel-pk-standard*
kernel-module-xcbc-4.19.34-intel-pk-standard		*kernel-module-xfrm-algo-4.19.34-intel-pk-standard*
kernel-module-xfrm-user-4.19.34-intel-pk-standard		*kernel-module-xhci-plat-hcd-4.19.34-intel-pk-standard*
kernel-module-xt-classify-4.19.34-intel-pk-standard		*kernel-module-xt-comment-4.19.34-intel-pk-standard*
kernel-module-xt-connbytes-4.19.34-intel-pk-standard		*kernel-module-xt-connlimit-4.19.34-intel-pk-standard*
kernel-module-xt-connmark-4.19.34-intel-pk-standard		*kernel-module-xt-conntrack-4.19.34-intel-pk-standard*
kernel-module-xt-ct-4.19.34-intel-pk-standard		*kernel-module-xt-dccp-4.19.34-intel-pk-standard*
kernel-module-xt-dscp-4.19.34-intel-pk-standard		*kernel-module-xt-ecn-4.19.34-intel-pk-standard*
kernel-module-xt-esp-4.19.34-intel-pk-standard		*kernel-module-xt-hashlimit-4.19.34-intel-pk-standard*
kernel-module-xt-helper-4.19.34-intel-pk-standard		*kernel-module-xt-hl-4.19.34-intel-pk-standard*
kernel-module-xt-length-4.19.34-intel-pk-standard		*kernel-module-xt-limit-4.19.34-intel-pk-standard*
kernel-module-xt-log-4.19.34-intel-pk-standard		*kernel-module-xt-mac-4.19.34-intel-pk-standard*
kernel-module-xt-mark-4.19.34-intel-pk-standard		*kernel-module-xt-multiport-4.19.34-intel-pk-standard*
kernel-module-xt-nat-4.19.34-intel-pk-standard		*kernel-module-xt-netmap-4.19.34-intel-pk-standard*
kernel-module-xt-nflog-4.19.34-intel-pk-standard		*kernel-module-xt-nfqueue-4.19.34-intel-pk-standard*
kernel-module-xt-pkttype-4.19.34-intel-pk-standard		*kernel-module-xt-policy-4.19.34-intel-pk-standard*
kernel-module-xt-quota-4.19.34-intel-pk-standard		*kernel-module-xt-realm-4.19.34-intel-pk-standard*
kernel-module-xt-redirect-4.19.34-intel-pk-standard		*kernel-module-xt-sctp-4.19.34-intel-pk-standard*
kernel-module-xt-state-4.19.34-intel-pk-standard		*kernel-module-xt-statistic-4.19.34-intel-pk-standard*
kernel-module-xt-string-4.19.34-intel-pk-standard		*kernel-module-xt-tcpmss-4.19.34-intel-pk-standard*
kernel-module-xt-tcpudp-4.19.34-intel-pk-standard		*kernel-module-xt-trace-4.19.34-intel-pk-standard*
kernel-module-xt-u32-4.19.34-intel-pk-standard		*kernel-module-zaurus-4.19.34-intel-pk-standard*
kernel-modules		
linux-tools-testusb *linux-tools-testusb*	1.0	GPL-2.0
lmsensors *lmsensors-libsensors*	3.4.0	GPLv2+, LGPLv2.1+
lmsensors-config *lmsensors-config-libsensors*	1.0	MIT-X

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

Recipe	Version	Licenses
lrzsz *lrzsz*	0.12.20	GPLv2+
lvm2 *lvm2* *lvm2-udevrules*	2.03.02 *lvm2-scripts*	GPLv2, LGPLv2.1
lximage-qt *lximage-qt*	0.14.1 *lximage-qt-locale-en-gb*	GPLv2
lxmenu-data *lxmenu-data*	0.1.5	LGPLv2.1
lxqt-about *lxqt-about*	0.14.1 *lxqt-about-locale-en-gb*	LGPLv2.1
lxqt-admin *lxqt-admin*	0.14.1 *lxqt-admin-locale-en-gb*	LGPLv2.1
lxqt-build-tools *lxqt-build-tools*	0.6.0	BSD-3-Clause
lxqt-config *lxqt-config*	0.14.1 *lxqt-config-locale-en-gb*	LGPLv2.1
lxqt-globalkeys *lxqt-globalkeys*	0.14.1	LGPLv2.1
lxqt-notificationd *lxqt-notificationd*	0.14.1	LGPLv2.1
lxqt-openssh-askpass *lxqt-openssh-askpass*	0.14.1	LGPLv2.1
lxqt-panel *lxqt-panel*	0.14.1	LGPLv2.1
lxqt-policykit *lxqt-policykit*	0.14.1	LGPLv2.1
lxqt-powermanagement *lxqt-powermanagement*	0.14.1	LGPLv2.1
lxqt-qtplugin *lxqt-qtplugin*	0.14.0	LGPLv2.1
lxqt-runner *lxqt-runner*	0.14.1	LGPLv2.1
lxqt-session *lxqt-session*	0.14.1	LGPLv2.1
lxqt-sudo *lxqt-sudo*	0.14.1	LGPLv2.1
lxqt-themes *lxqt-themes*	0.14.0	LGPLv2.1
lxqt-world *lxqt-world*	1.0	MIT
lzo *lzo*	2.10	GPLv2+
mcelog *mcelog*	162	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*	19.0.1 *libgbm* *libglapi* *libgles2-mesa* *mesa-vulkan-drivers*	MIT
mesa-demos *mesa-demos*	8.3.0	MIT, PD
mini-x-session *mini-x-session*	0.1	GPLv2
minicom	2.7.1	GPLv2+

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

Recipe	Version	Licenses
minicom		
mmc-utils *mmc-utils*	0.1	GPLv2
mobile-broadband-provider-info *mobile-broadband-provider-info*	20190116	PD
modutils-initscripts *modutils-initscripts*	1.0	PD
mozjs *libmozjs*	52.9.1	MPL-2.0
mpeg2dec *libmpeg2*	0.5.1	GPLv2+
mpfr *mpfr*	4.0.2	LGPLv3+
mpg123 *mpg123*	1.25.10	LGPLv2.1
msc-bug-report *msc-bug-report*	1.0	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
msc-screensaver *msc-screensaver*	git	GPLv2
msc-wallpaper *msc-wallpaper*	git	GPLv2
mscio-cmd *mscio-cmd*	git	LGPLv2.1
mscio-drivers *kernel-module-devreg-4.19.34-intel-pk-standard* *kernel-module-eapi-ec-bl-4.19.34-intel-pk-standard* *kernel-module-eapi-ec-running-time-4.19.34-intel-pk-standard* *kernel-module-i2c-ids-4.19.34-intel-pk-standard* *kernel-module-user-gpios-4.19.34-intel-pk-standard*	git *kernel-module-eapi-ec-4.19.34-intel-pk-standard* *kernel-module-eapi-ec-hwm-4.19.34-intel-pk-standard* *kernel-module-eapi-ec-wdt-4.19.34-intel-pk-standard* *kernel-module-leds-gpio-acpi-4.19.34-intel-pk-standard* *mscio-drivers*	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*	2.0.2	GPLv2+
mtdev *mtdev*	1.1.5	MIT

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

Recipe	Version	Licenses
muparser *muparser*	2.2.6	MIT
nano *nano*	2.2.5	GPLv2
ncurses *ncurses-libformw* *ncurses-libncurses* *ncurses-libpanelw* *ncurses-terminfo*	6.1 *ncurses-libmenuw* *ncurses-libncursesw* *ncurses-libtinfo* *ncurses-terminfo-base*	MIT
ndctl *ndctl*	v63	GPLv2+
neard *neard*	0.16	GPLv2
netbase *netbase*	5.6	GPLv2
netpipe *netpipe*	3.7.2	GPLv1
nettle *nettle*	3.4.1	LGPLv3+, GPLv2+
nfs-utils *nfs-utils-client*	2.3.3 *nfs-utils-mount*	MIT, GPLv2+, BSD
npth *npth*	1.6	LGPLv2+
nspr *nspr*	4.21	GPL-2.0, MPL-2.0, LGPL-2.1
nss *nss*	3.42.1	(MPL-2.0 & LGPL-2.1+), (MPL-2.0 & GPL-2.0+), MPL-2.0
ntp *ntp* *ntpdate*	4.2.8p13 *ntp-tickadj*	NTP
obconf-qt *obconf-qt*	0.14.1	LGPLv2.1
ofono *ofono*	1.25	GPLv2
openbox *openbox* *openbox-core* *openbox-theme-bear2* *openbox-theme-clearlooks-3.4* *openbox-theme-mikachu* *openbox-theme-onyx* *openbox-theme-orang*	3.6.1 *openbox-config* *openbox-theme-artwiz-boxed* *openbox-theme-clearlooks* *openbox-theme-clearlooks-olive* *openbox-theme-natura* *openbox-theme-onyx-citrus* *openbox-theme-syscrash*	GPLv2+
openssl *libcrypto* *openssl*	1.1.1b *libssl* *openssl-conf*	openssl
opkg-utils *update-alternatives-opkg*	0.4.0	GPLv2+
orc *liborc-0.4*	0.4.28	BSD-2-Clause, BSD-3-Clause
otter-browser *otter-browser*	1.0.01	GPLv3
oxygen-icons5 *oxygen-icons5*	5.57.0	LGPLv3
packagegroup-base *packagegroup-base* *packagegroup-base-acpi*	1.0 *packagegroup-base-3g* *packagegroup-base-alsa*	MIT

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

Recipe	Version	Licenses
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-dropbear	1.0	MIT
packagegroup-core-ssh-dropbear		
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		
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		
pango	1.42.4	LGPLv2.0+
pango		
parole	1.0.2	GPLv2
parole	*parole-locale-en-gb*	
parted	3.2	GPLv3+
parted		
pavucontrol-qt	0.14.1	GPLv2
pavucontrol-qt		
pci2uio	git	GPLv2
kernel-module-pci2uio-4.19.34-intel-pk-standard	*pci2uio*	
pciutils	3.6.2	GPLv2+
libpci	*pciutils*	
pciutils-ids		
pcmanfm-qt	0.14.1	GPLv2
pcmanfm-qt	*pcmanfm-qt-locale-en-gb*	
perf	1.0	GPLv2
perf		
perl	5.28.1	Artistic-1.0, GPL-1.0+
perl	*perl-module-config-heavy*	
pinentry	1.1.0	GPLv2
pinentry		
pixman	0.38.0	MIT, MIT-style, PD
pixman		
pm-utils	1.4.1	GPLv2
pm-utils		
polkit	0.115	LGPLv2+
polkit		
polkit-group-rule-datetime	1.0	MIT
polkit-group-rule-datetime		
polkit-qt-1	0.112.0	LGPLv2.1
polkit-qt-1		
popt	1.16	MIT
popt		

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

Recipe	Version	Licenses
powertop	2.10	GPLv2
powertop	*powertop-locale-en-gb*	
powertop-locale-en-us		
procps	3.3.15	GPLv2+, LGPLv2+
procps		
proxy-config	1.0	MIT
proxy-config		
psplash	0.1	GPLv2+
psplash	*psplash-default*	
pulseaudio	12.2	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*	
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*	
python	2.7.15	PSFv2
libpython2	*python-codecs*	
python-compression	*python-contextlib*	
python-core	*python-crypt*	
python-ctypes	*python-datetime*	
python-email	*python-fcntl*	
python-io	*python-lang*	
python-math	*python-mime*	
python-netclient	*python-pickle*	
python-shell	*python-stringold*	
python-subprocess	*python-textutils*	
python-threading	*python-unixadmin*	
python-zlib		
python-imaging	1.1.7	MIT
python-imaging		
python3	3.7.2	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-runpy	*python3-shell*	
python3-smtpd	*python3-sqlite3*	

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

Recipe	Version	Licenses
python3-stringold *python3-terminal* *python3-tkinter* *python3-unittest* *python3-venv* *python3-xmlrpc*	*python3-syslog* *python3-threading* *python3-typing* *python3-unixadmin* *python3-xml*	
python3-async *python3-async*	0.6.2	BSD
python3-dbus *python3-dbus*	1.2.8	MIT
python3-ewmh *python3-ewmh*	0.1.5	LGPL-3.0
python3-git *python3-git*	2.1.11	BSD-3-Clause
python3-gitdb *python3-gitdb*	2.0.5	BSD-3-Clause
python3-iniparse *python3-iniparse*	0.4	MIT, PSF
python3-pycairo *python3-pycairo*	1.15.6	LGPLv2.1, MPLv1.1
python3-pygobject *python3-pygobject*	3.28.3	LGPLv2.1
python3-setuptools *python3-setuptools*	40.8.0	MIT
python3-six *python3-six*	1.10.0	MIT
python3-smmap *python3-smmap*	2.0.5	BSD
python3-xlib *python3-xlib*	0.25	LGPL-2.1
pyxdg *pyxdg*	0.26	LGPLv2
qps *qps*	1.10.20	GPLv2
qtbase *qtbase* *qtbase-qmlplugins*	5.12.3 *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*	5.12.3 *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*	0.14.1	GPLv2
qtermwidget *qtermwidget*	0.14.1	GPLv2

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

Recipe	Version	Licenses
qtimageformats	5.12.3	(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
qtimageformats-plugins		
qtlocation	5.12.3	(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
qtlocation	*qtlocation-plugins*	
qtlocation-qmlplugins		
qtmultimedia	5.12.3	(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
qtmultimedia	*qtmultimedia-plugins*	
qtmultimedia-qmlplugins		
qtsensors	5.12.3	(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-plugins*	
qtsensors-qmlplugins		
qtsvg	5.12.3	(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-plugins*	
qtsvg-qmlplugins		
qtwebchannel	5.12.3	(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-plugins*	
qtwebchannel-qmlplugins		
qtwebkit	5.12.3	BSD, LGPLv2+, GPL-2.0
qtwebkit	*qtwebkit-plugins*	
qtwebkit-qmlplugins		

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

Recipe	Version	Licenses
qtx11extras	5.12.3	(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-plugins*	
qtx11extras-qmlplugins		
qtxmlpatterns	5.12.3	(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-plugins*	
qtxmlpatterns-qmlplugins		
readline	8.0	GPLv3+
readline		
rgb	1.0.6	MIT-X
rgb		
rpcbind	1.2.5	BSD
rpcbind		
rpm	4.14.2.1	GPL-2.0
python3-rpm	*rpm*	
run-postinsts	1.0	MIT
run-postinsts		
sbc	1.4	LGPLv2.1+
sbc		
shadow	4.6	BSD, Artistic-1.0
shadow	*shadow-base*	
shadow-securetty	4.6	MIT
shadow-securetty		
shared-mime-info	1.10	GPLv2
shared-mime-info	*shared-mime-info-data*	
slang	2.3.2	GPLv2
slang		
smartmontools	6.3	GPLv2
smartmontools-ctl		
solid	5.57.0	LGPLv2.1
solid	*solid-locale-en-gb*	
speex	1.2.0	BSD
speex		
speexdsp	1.2rc3	BSD
speexdsp		
spi-register	git	GPLv2
spi-register		
sqlite3	3.27.2	PD
libsqlite3		
strace	4.26	LGPL-2.1+, GPL-2+
strace		
stress	1.0.4	GPLv2
stress		
sudo	1.8.27	ISC, BSD, Zlib
sudo		
superiotool	6637	GPLv2
superiotool		
system-test-controller-config	1.0	MIT
system-test-controller-config		
system-test-controller-v2	git	GPLv2

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

Recipe	Version	Licenses
system-test-controller-v2		
sysvinit *sysvinit*	2.88dsf *sysvinit-pidof*	GPLv2+
sysvinit-inittab *sysvinit-inittab*	2.88dsf	GPLv2
taglib *taglib*	1.11.1 *taglib-c*	LGPLv2.1, MPL-1
tcp-wrappers *libwrap*	7.6	BSD
tcpdump *tcpdump*	4.9.2	BSD
thin-provisioning-tools *thin-provisioning-tools*	0.7.6	GPLv3
tiff *tiff*	4.0.10	BSD-2-Clause
tiny-shell *tiny-shell*	git	GPLv2
tpm-tools *tpm-tools*	1.3.9.1	CPL-1.0
tpm2-abrmd *tpm2-abrmd*	2.0.3	BSD-2-Clause
tpm2-tools *tpm2-tools*	3.1.1	BSD
tpm2-tss *libtss2* *libtss2-tcti-device* *tpm2-tss*	2.0.0 *libtss2-mu* *libtss2-tcti-mssim*	BSD-2-Clause
trousers *libtspi*	0.3.14 *trousers*	BSD
turbostat *turbostat*	3.4	GPLv2
tzdata *tzdata* *tzdata-americas* *tzdata-arctic* *tzdata-atlantic* *tzdata-core* *tzdata-misc* *tzdata-posix*	2019a *tzdata-africa* *tzdata-antarctica* *tzdata-asia* *tzdata-australia* *tzdata-europe* *tzdata-pacific* *tzdata-right*	PD, BSD, BSD-3-Clause
udev-extraconf *udev-extraconf*	1.1	MIT
udisks2 *udisks2* *udisks2-locale-en-gb*	2.8.2 *udisks2-libs*	GPLv2+, LGPLv2+
update-flash *update-flash*	git	GPLv2
update-rc.d *update-rc.d*	0.8	GPLv2+
usbutils *usbutils*	010	GPLv2+
useradd-msc *useradd-msc*	1.0	MIT
util-linux *util-linux-blkid* *util-linux-fstrim* *util-linux-libmount* *util-linux-libuuid*	2.32.1 *util-linux-dmesg* *util-linux-libblkid* *util-linux-libsmartcols* *util-linux-lsblk*	GPLv2+, LGPLv2.1+, BSD

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

Recipe	Version	Licenses
util-linux-lscpu	*util-linux-mcookie*	
util-linux-mount	*util-linux-sulogin*	
util-linux-swapoff	*util-linux-swapon*	
util-linux-swaponoff	*util-linux-umount*	
volume-key	0.3.12	GPLv2
volume-key	*volume-key-locale-en-gb*	
vulkan	1.1.73.0	Apache-2.0
vulkan		
wayland	1.17.0	MIT
wayland		
wireless-regdb	2019.03.01	ISC
wireless-regdb-static		
wpa-supplicant	2.7	BSD
wpa-supplicant	*wpa-supplicant-cli*	
wpa-supplicant-passphrase		
xauth	1.0.10	MIT-X
xauth		
xcb-util	0.4.0	MIT
xcb-util		
xcb-util-image	0.4.0	MIT
xcb-util-image		
xcb-util-keysyms	0.4.0	MIT
xcb-util-keysyms		
xcb-util-renderutil	0.3.9	MIT
xcb-util-renderutil		
xcb-util-wm	0.4.1	MIT
xcb-util-wm		
xdg-user-dirs	0.17	GPLv2
xdg-user-dirs		
xdpyinfo	1.3.2	MIT-X
xdpyinfo		
xf86-input-libinput	0.28.2	MIT-X
xf86-input-libinput		
xf86-video-ast	1.1.5	MIT-X
xf86-video-ast		
xf86-video-fbdev	0.5.0	MIT-X
xf86-video-fbdev		
xf86-video-intel	2.99.917	MIT-X
xf86-video-intel		
xf86-video-vesa	2.4.0	MIT-X
xf86-video-vesa		
xfconf	4.13.6	GPLv2
xfconf	*xfconf-locale-en-gb*	
xhost	1.0.8	MIT-X
xhost		
xinit	1.4.1	MIT-X
xinit		
xinput	1.6.2	MIT-X
xinput		
xinput-calibrator	0.7.5	MIT-X
xinput-calibrator		
xkbcomp	1.4.2	MIT-X
xkbcomp		
xkeyboard-config	2.26	MIT, MIT-style
xkeyboard-config	*xkeyboard-config-locale-en-gb*	
xmodmap	1.0.10	MIT
xmodmap		
xrandr	1.5.0	MIT
xrandr		

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

Recipe	Version	Licenses
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*	1.20.4 *xserver-xorg* *xserver-xorg-module-libint10*	MIT-X
xset *xset*	1.2.4	MIT
xssstate *xssstate*	1.1	MIT
xterm *xterm*	330	MIT-X
xvinfo *xvinfo*	1.1.4	MIT-X
xz *liblzma*	5.2.4	PD
zip *zip*	3.0	BSD-3-Clause
zlib *zlib*	1.2.11	Zlib
zsh *zsh*	5.4.2	zsh
zsh-config *zsh-config*	1.0	MIT