

**Engineering Leadership**

**MSC-LDK Manual**  
**BSP 0584\_nanoRISC\_imx6 V1.6.0**  
2020-08-03

## **Copyright Notice**

Copyright © 2014-2020 by MSC Technologies GmbH. All rights reserved. Copying of this document, and giving it to others and the use or communication of the contents thereof, are forbidden without express authority. Offenders are liable to the payment of damages. All rights are reserved in the event of the grant of a patent or the registration of a utility model or design.

## **Trademarks**

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

## **Certification**

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

## **Audience**

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

## **Life-Cycle-Management**

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

## **Disclaimer**

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

# Contents

<b>1. History</b>	<b>9</b>
1.1. Document	9
1.2. MSC-LDK	10
1.3. BSP 0584_nanoRISC_imx6	13
<b>2. Introduction</b>	<b>14</b>
2.1. Scope	14
2.2. Out Of Scope	14
2.3. Features	14
2.4. Conventions	14
2.5. Product Support	15
<b>3. Getting Started</b>	<b>16</b>
3.1. Requirements	16
3.2. Registration On The MSC Git Server	16
3.2.1. Creating An SSH Key	16
3.2.2. Configuring For HTTP proxies	17
<b>4. Using MSC-LDK</b>	<b>18</b>
4.1. Yocto/MSC-LDK Terms	18
4.2. Setup Optional Docker Container	18
4.3. Installation Of MSC-LDK	20
4.4. Directory Layout	20
4.4.1. MSC-LDK Layers	21
4.5. Installation Of MSC-LDK BSPs	22
4.5.1. Repeated setup for target	22
4.6. BSP Aliases	22
4.7. Reduced Cleanup Time	22
4.8. Building Images	23
4.9. Traceable And Reproducible Images	23
4.10. Building Documentation	24
4.11. Image Types	24
4.11.1. Headless	24
4.11.2. LXQt	25
4.11.3. LXQt SDK	25
4.11.4. Sato	25
4.11.5. Sato SDK	25
4.12. Deploying Images To The Hardware	26
4.12.1. SD card	26
4.13. Login	32
4.14. Enabled Services	32
4.15. Enhancing The Images	32
4.15.1. Adding Other Packages	32
4.15.2. Adding Own Layers	32
4.15.3. Network Configuration	33
4.15.4. Time Zone Setup	33
4.16. Using LXQt	34
4.17. ConnMan Configuration	34
4.17.1. Proxy Configuration For The otter-browser	37
4.18. Using Sato	38
4.19. Using The SDK Images	38
4.19.1. Setting Up The Development Host	38
4.19.2. Setting Up The Target	38
4.19.3. Building hello_world Manually	39

4.19.4. Building hello_world With CMake	39
4.19.5. Building Qt Based hello_world (Only In msc-image-lxqt-sdk)	39
4.20. Bug Reporting	40
4.21. Hotfixes And Updating MSC-LDK	42
<b>5. Package List</b>	<b>43</b>
5.1. mscio	43
5.2. system-test-controller	43
<b>A. Glossary</b>	<b>44</b>
<b>B. Bootloader Configuration</b>	<b>45</b>
B.1. Booting	45
B.2. Boardinfo	46
B.3. Configure Devicetree	46
B.4. Update U-Boot and SPL	47
B.5. Change Display settings in U-Boot	47
B.6. Change Boot order	48
<b>C. Boarddata</b>	<b>49</b>
C.1. Boarddata EEPROM Layout	49
C.2. Reading Boarddata	49
<b>D. Devicetree Configuration</b>	<b>50</b>
D.1. Concept	50
D.2. Structure used for nanoRISC	50
D.3. Creating own variants	52
<b>E. QT5 Graphics Demos</b>	<b>56</b>
E.1. QML_BillboardDemo	56
E.2. Qt5_CinematicExperienceDemo	57
E.3. Qt5_EverywhereDemo	57
E.4. Qt5_NMapperDemo	58
E.5. Qt5_NmapcarouseDemo	58
E.6. Qt5_QuitBatteryDemo	59
E.7. Qt5_SmartHomeDemo	59
<b>F. Supported Periphery</b>	<b>60</b>
F.1. eMMC	61
F.2. GPIOs	61
F.3. Audio	61
F.3.1. Playback on Headphone	61
F.3.2. Playback on Speaker	61
F.3.3. Record from Mic	62
F.3.4. Record from LineIn	62
F.3.5. Playback With HDMI	62
F.3.6. Persistent Mixer Settings	63
F.4. I2C	63
F.5. USB Device Support	63
F.5.1. Testing Connection	63
F.5.2. Setting Up Serial Gadget	63
F.5.3. Setting Up Ethernet Gadget	63
F.5.4. Setting Up Mass/File Storage Gadget	64
F.6. UART	64
F.7. SPI	64
F.8. Graphic Output	64
F.8.1. Display Support	64
<b>G. Troubleshooting</b>	<b>65</b>
<b>H. Tips And Tricks</b>	<b>66</b>
H.1. Using A Standard Browser	66
H.1.1. Using The google-chrome Browser	66

H.2. Timekeeping	66
H.2.1. htpdate	66
<b>I. Security</b>	<b>67</b>
I.1. Introduction	67
I.2. Layer meta-security-isafw	67
I.3. Setting Up A Project	67
<b>J. Power Analysis</b>	<b>69</b>
J.1. powertop	69
<b>K. Links</b>	<b>70</b>
<b>L. License Overview</b>	<b>71</b>

# List of Figures

- 4.1. MSC-LDK Layers . . . . . 21
- 4.2. LXQt - Mainwindow . . . . . 34
- 4.3. Start ConnMan Property Editor . . . . . 34
- 4.4. View ConnMan Properties . . . . . 35
- 4.5. Manual IP Assignment . . . . . 35
- 4.6. Otter Browser - Proxy Setup . . . . . 37
- 4.7. Sato - Mainwindow . . . . . 38
- 4.8. Sato - Groups . . . . . 38
- 4.9. Sato - Selecting Running Apps . . . . . 38
- 4.10. Bug Reporting - Start . . . . . 40
- 4.11. Bug Reporting - Start . . . . . 40
- 4.12. Bug Reporting - View . . . . . 41
- 4.13. Bug Reporting - Create Zip . . . . . 41
- 4.14. Bug Reporting - Save Zip . . . . . 41
  
- D.1. example BoardMark . . . . . 51
  
- E.1. Billboard Demo . . . . . 56
- E.2. Cinematic Experience Demo . . . . . 57
- E.3. Everywhere Demo . . . . . 57
- E.4. NMapper Demo . . . . . 58
- E.5. Nmapcarouse Demo . . . . . 58
- E.6. QuitBattery Demo . . . . . 59
- E.7. SmartHome Demo . . . . . 59
  
- J.1. powertop overview . . . . . 69

# List of Tables

4.1. MSC-LDK Directory Layout . . . . .	20
4.2. MSC-LDK Layer Description . . . . .	21
4.3. User Accounts . . . . .	32
F.1. Supported periphery . . . . .	60
F.2. GPIOs . . . . .	61
G.1. Errors . . . . .	65
J.1. CPU core states . . . . .	69
L.1. list of licenses used in this build . . . . .	71
L.2. License overview listing . . . . .	72
L.2. License overview listing (continued) . . . . .	73
L.2. License overview listing (continued) . . . . .	74
L.2. License overview listing (continued) . . . . .	75
L.2. License overview listing (continued) . . . . .	76
L.2. License overview listing (continued) . . . . .	77
L.2. License overview listing (continued) . . . . .	78
L.2. License overview listing (continued) . . . . .	79
L.2. License overview listing (continued) . . . . .	80

# Listings

- bsp/yocto-output.txt . . . . . 23
- 4.1. /etc/exports . . . . . 38
- 4.2. hello\_world.c . . . . . 39
- 4.3. CMakeLists.txt . . . . . 39
- 4.4. hello\_world.cpp . . . . . 39
- 4.5. hello\_world.pro . . . . . 40
  
- bsp/example-devicetree.dts . . . . . 53
- bsp/msc-nR-imx6-custom-lvds.dtsi . . . . . 54



# 1. History

## 1.1. Document

Revision	Changes
1.6.0	<ul style="list-style-type: none"><li>• Added chapter “Hardware monitoring sensors”</li><li>• Added chapters “Enabling WiFi” and “Use the board as an access point”</li><li>• Various fixes and cleanups</li></ul>
1.5.0	<ul style="list-style-type: none"><li>• Added chapter “I2C Bus Identification”</li><li>• Added chapter “Reduced Cleanup Time”</li><li>• Various fixes</li></ul>
1.4.0	<ul style="list-style-type: none"><li>• Removed chapter “Microsoft Azure”</li></ul>
1.3.0	<ul style="list-style-type: none"><li>• Added chapter “Power Analysis”</li><li>• Added chapter “MSC-IO”</li><li>• Added section “Setup Optional Docker Container”</li></ul>
1.2.0	<ul style="list-style-type: none"><li>• Added chapter “Microsoft Azure”</li><li>• Added chapter “Real-Time”</li><li>• Added chapter “Security”</li><li>• Added chapter “Tips And Tricks”</li><li>• Added section “ConnMan Configuration”</li><li>• Updated section “Deploying Images To The Hardware” (new image type .wic)</li><li>• Updated section “Enhancing The Images”</li><li>• Updated section “Installation Of MSC-LDK”</li></ul>
1.1.0	<ul style="list-style-type: none"><li>• Added section “Traceable and Reproducible Images”</li><li>• Added section “Hotfixed and Updating MSC-LDK”</li><li>• Updated section “Installation of MSC-LDK”</li></ul>

---

Revision	Changes
----------	---------

---

1.0.0

- Added section “Using LXQt”
  - Added section “Using the SDK images”
  - Added section “Bug Reporting”
- 

## 1.2. MSC-LDK

Revision	Changes
----------	---------

---

1.6.0

- Updated to yocto 3.0.1 (zeus)
- Added layers meta-chrome-instead-of-otter and boot2qt
- Added support for Ryzen and .iMX8 BSPs
- "make install" copies .wic.bmap files for use with bmaptool

1.5.0

- Updated to yocto 2.7 (warrior)
- Added layer-marvell
- Removed meta-iot-cloud and meta-java dependencies for msc-image-lxqt
- Added layer meta-bsp-arm-verification
- Various improvements to build MSC-LDK BSPs in a Docker container
- Various improvements to build feature branches of the BSPs

1.4.0

- Updated to yocto 2.5 (sumo)
  - Added layer-marvell
  - Removed meta-iot-cloud and meta-java dependencies for msc-image-lxqt
-

1.3.0

- Updated to yocto 2.4 (rocko)
- ApolloLake, Braswell, Skylake and Baytrail BSPs have been merged to IntelCombined BSP
- Added layers-debug for simplify kernel debugging
- Various fixes for building MSC-LDK when /bin/sh is dash instead of bash.
- Added experimental setup option `--with-tmp-image` to speed up cleanup of build directories
- Uses MSC-IO v3 to support ARM and setup MSC-IO and EAPI by ACPI BIOS entries
- zsh is used as login shell
- Four instead of one text consoles are available (access with Alt-F1...Alt-F4)
- On x86 the port for the serial console is guessed instead of hardcoded `/dev/ttyV0` which might be unavailable.
- Various tool changes and smaller bug fixes.

1.2.0

- Updated to yocto 2.2 (morty)
- All kernels are compiled with `CONFIG_DYNAMIC_DEBUG` for easier hardware bring-up
- X desktop is run as user "msc" instead of "root"
- Supporting real-time kernel
- LXQt is the preferred desktop environment instead of sato
- setup supports `--bsp-build-dir-name`
- setup supports `--layers-security`
- setup supports installation of 3rdparty layers
- Builds of different BSPs don't share the sstate-cache any longer
- MSC-LDK repository has been relocated

1.1.0

- Using setup.py instead of setup.sh
- setup.py supports configuration of the BSP exactly as a previous image has been built (traceable and reproducible images)
- Added certified azure libraries to standard image
- Supporting speaking names for BSPs, e.g. Q7-BT or Baytrail instead of C984
- Added fallback mirror server `ftp4.ebv.com`

Revision	Changes
----------	---------

---

1.0.0

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

### 1.3. BSP 0584\_nanoRISC\_imx6

Revision	Changes
0.4.0	<ul style="list-style-type: none"><li>• Updated to kernel 4.14.0</li></ul>
0.3.0	<ul style="list-style-type: none"><li>• Updated to kernel 4.1.15</li></ul>
0.2.0	<ul style="list-style-type: none"><li>• Added nanoRISC i.MX6 modules</li></ul>
0.1.0	Initial

## 2. Introduction

This document is intended to be used by developers creating or adapting Linux systems for MSC Technologies hardware with the MSC Linux Development Kit. The [MSC-LDK](#) provides an environment to create Linux kernels, bootloaders and root filesystems. It is based on the Yocto 3.0.1 project.

### 2.1. Scope

This document gives a hand in:

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

### 2.2. Out Of Scope

Detailed information about Yocto is not part of this document but available at:

<https://www.yoctoproject.org>.

### 2.3. Features

Features of the MSC-LDK are:

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

### 2.4. Conventions

This section describes the conventions used in this manual.



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



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



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



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

Typographical conventions:

*Courier New 9Pt*      Screen text, user-typed command-line entries, or source code.

[Ctrl]+C              Two or more keys that must be pressed simultaneously.

## 2.5. Product Support

MSC engineers and technicians are committed to provide support to our customers whenever needed.

If the information provided there does not solve your problem, please contact our Technical Support:

WWW: <http://www.msc-technologies.eu/de/support.html>

## 3. Getting Started

### 3.1. Requirements

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

### 3.2. Registration On The MSC Git Server

Downloading any files from the MSC Git server requires a registration on:

<http://www.msc-technologies.eu/register.html>.

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

#### 3.2.1. Creating An SSH Key

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

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

```

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



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





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

### 3.2.2. Configuring For HTTP proxies

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

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

## 4. Using MSC-LDK

### 4.1. Yocto/MSK-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=https://<ip>:<port>
#ENV http_proxy ${http_proxy}
#ENV https_proxy ${https_proxy}
#RUN echo "Acquire::http::Proxy \"${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 ${https_proxy}
RUN echo "Acquire::http::Proxy \"${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_0199/docker-msc-ldk
user@devhost:~$ cd docker-msc-ldk
user@devhost:~$ git checkout v1.6.0
```

```

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

```

To use the container on an Ubuntu host:

```

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

```

To use the container on a CentOS host:

```

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

```

Inside the docker container MSC LDK has to be cloned as described in **??**. Inside the cloned directory BSPs can be installed and built as shown in sections **??** and **??**, respectively.

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

```

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

```

## 4.3. Installation Of MSC-LDK

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

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_o199/msc-ldk
user@devhost:$ cd msc-ldk
user@devhost:$ git checkout v1.6.0
```

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

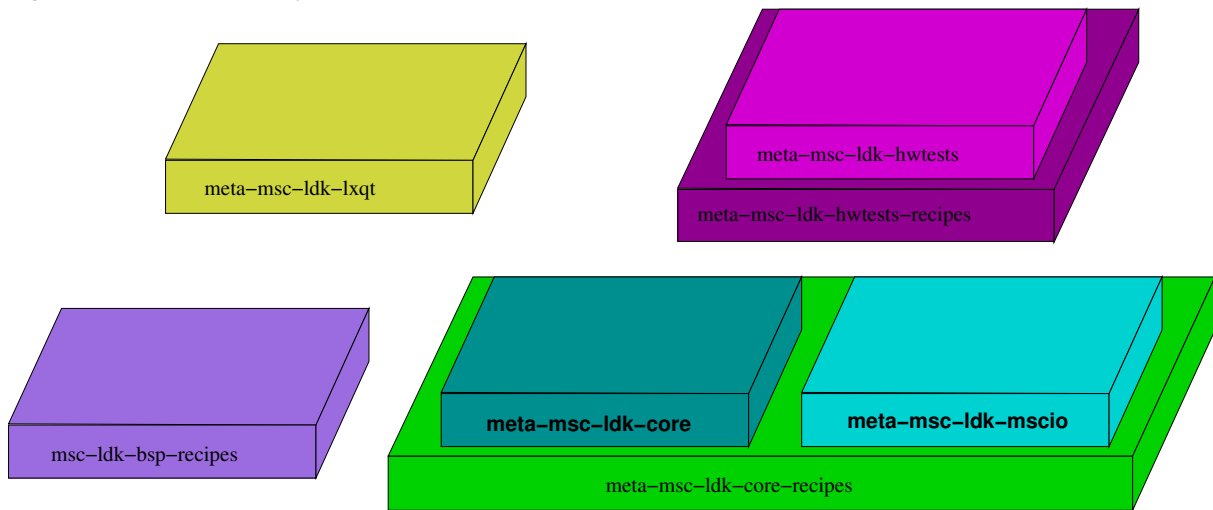
**Table 4.1.** – MSC-LDK Directory Layout

Directory	Contents
<code>build/0584</code>	BSP build directory.
<code>doc</code>	This contains all the generated Yocto documentation. (only if build with: <code>make doc</code> )
<code>downloads/</code>	All downloaded sources are stored here. This directory can be shared with other MSC-LDK installations.
<code>scripts/</code>	Build helper scripts.
<code>sources/0584/*.git</code>	BSP specific layers. (will be created in chapter 4.5)
<code>sources/meta-freescale.git/</code>	Freescale specific layers, only on Freescale architectures.
<code>sources/meta-msc-ldk-*.git/</code>	MSC-LDK layers.
<code>sources/meta-openembedded-*.git/</code>	Additional useful tools that are not part of Yocto.
<code>sources/meta-qt5*.git/</code>	Qt5 layers used by <code>msc-image-lxqt</code> .
<code>sources/yocto.git/</code>	Yocto sources used by MSC-LDK.
<code>sstate-cache/</code>	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
<code>msc-ldk-bsp-recipes</code>	BSP related layer, e.g. Kernel patches for a board. Mandatory for a given board.
<code>meta-msc-ldk-core-recipes</code>	Required recipes to build MSC specific applications.
<code>meta-msc-ldk-core</code>	MSC specific image recipes and package groups. Depends on <code>meta-msc-ldk-core-recipes</code> .
<code>meta-msc-ldk-hwtests-recipes</code>	Hardware test specific recipes.
<code>meta-msc-ldk-hwtests</code>	Board specific image <code>msc-image-hwtests-&lt;BOARD&gt;</code> . Depends on <code>meta-msc-ldk-hwtests-recipes</code> .
<code>meta-msc-ldk-mscio</code>	MSC-IO layer. Depends on <code>meta-msc-ldk-core-recipes</code> .
<code>meta-msc-ldk-lxqt</code>	Support for the Lightweight Qt Desktop Environment LXQt ( <a href="http://lxqt.org">http://lxqt.org</a> ).

## 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):

So you command to setup for target is:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584
```

This will clone the repository `0584/msc-ldk-bsp-recipes.git` and several more, depending on the recipes in this layer. This is reported by several messages. If all clonings are succeeded, you may change into the specific build directory.

```
user@devhost:msc-ldk$ cd build/0584
user@devhost:msc-ldk/build/0584$
```

Further steps, building a dedicated image, are described in section [4.11](#)

### 4.5.1. Repeated setup for target

If you have setup a bsp variant before and repeat the above command, you get the following error:

```
ERROR: '/work/msc-ldk/build/0584*' does already exist -> Skipping .conf generation (use -{}->
re-create-conf to force .conf generation)
```

This is not really an error, perhaps some recipes or layer have been changed. Then you can update the build configuration whith:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --re-create-conf
```

And then continue as usual:

```
user@devhost:msc-ldk$ cd build/0584
user@devhost:msc-ldk/build/0584$
```

## 4.6. BSP Aliases

`setup.py` supports one of the following aliases for the option `--bsp=`  
`0584,nanoRiscimx6,nRimx6`

## 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=0584
user@devhost:msc-ldk$ cd build/0584
user@devhost:0584$ 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 Reproducible 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.

```
Build Configuration:
BB_VERSION           = "1.42.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING     = "universal"
TARGET_SYS          = "arm-poky-linux-gnueabi"
MACHINE              = "msc-q7-imx6plusx"
DISTRO               = "poky"
DISTRO_VERSION       = "2.7"
TUNE_FEATURES        = "arm armv7a vfp thumb neon callconvention-hard"
TARGET_FPU           = "hard"
meta-poky
meta-yocto-bsp       = "warrior-msc:8dbaa2305eca79d2f45d1c242c61cdcc65a70f8d"
meta
meta
meta-oe
meta-networking
meta-python          = "warrior-msc:f1b18c0da7f247998c2269069d7b47ed1e62bf12"
meta-qt5.git         = "warrior-msc:8586caf2f2f5c11fb2ddff29eefe5dd7fc08cf38"
meta
meta-integrity
meta-signing-key
meta-tpm
meta-tpm2            = "warrior:0fbc3c26c5836b5a040d8799ffa70d95493930e3"
meta-msc-ldk-core-recipes.git = "develop:69ab08087b019f3531e66756b857f68d964c4f7e"
meta-msc-ldk-core.git = "develop:a2cbdalbf4ef5b06ff3be76438e0d3a187a35d"
meta-msc-ldk-mscio.git = "develop:266fa8a8b3a6e2359d6cbb92ea56408cfd6f9a95"
meta-freescale.git  = "warrior:39313622374d93dee232777ed2f7e8e6364a0744"
meta-qt5-extra.git  = "warrior-msc:2b32b3c0318c45b6a69614dde8e98968ae4d6e75"
meta-msc-arm-extensions.git = "develop:6730ae7b9c043e06fdelac3dc74f689a4afe7a73"
meta-msc-qt5-testapplication.git = "develop:2533)
ccc13fe8230fa88d24b1a60e6c8a9c6979db"
meta-msc-ldk-marvell.git = "develop:1ecb63050d3df63d6abb9a47d3cf0672d7303502"
meta-multimedia
meta-gnome           = "warrior-msc:f1b18c0da7f247998c2269069d7b47ed1e62bf12"
```

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/msc_nr_imx6-poky-linux-gnueabi/<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/0584
user@devhost:msc-ldk/build/0584$ 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. `enpls0`.

### 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 built unified for Solo till QuadPlus CPUs by:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --layers-lxqt
user@devhost:msc-ldk$ cd build/0584-lxqt
user@devhost:msc-ldk/build/0584-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 is build for Solo till QuadPlus CPUs by:

```
user@devhost:msc-ldk$ cd build/0584-lxqt
user@devhost:msc-ldk/build/0584-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/0584
user@devhost:msc-ldk/build/0584$ 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/0584
user@devhost:msc-ldk/build/0584$ 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. SD card

To create a bootable SD Card with a root filesystem, copy the image `msc-image-sato-msc-nr-imx6.sdcard` to the SD-Card (e.g. on `/dev/sdd`) with:

```
user@devhost:~$ cd msc-ldk/build/0584/tmp/deploy/images/msc-nr-imx6
user@devhost:images$ sudo dd if=msc-image-sato-msc-nr-imx6.sdcard of=/dev/sdd bs=4M
```

Then insert the SD-Card into the module slot and reset the system. The boot process is listed here as a reference, the output may vary using a different device tree.

```
U-Boot SPL 2018.03-+g26c8f2bad6 (Nov 22 2019 - 16:53:22 +0000)
Boardinfo:
  name ..... msc-nR-imx6
  variant ..... 112
  feature ..... 014
  serial ..... 1002477607
  revision ..... N/A
Trying to boot from MMC1

U-Boot 2018.03-+g26c8f2bad6 (Nov 22 2019 - 16:53:22 +0000)

CPU:   Freescale i.MX6SOLO rev1.1 at 792MHz
CPU:   Industrial temperature grade (-40C to 105C) at 40C
CPU:   Reset cause is POR
Board: MSC nanoRISC i.MX6
I2C:   ready
DRAM:  1 GiB
LCD,   HDMI,
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
Loading Environment from MMC... OK
Display: hdmi:1024x768M@60 (1024x768)
splashfile is already set to 'logos/msc-logo-1024x768.bmp'
In:    serial
Out:   serial
Err:   serial
SF: Detected m25p16 with page size 256 Bytes, erase size 64 KiB, total 2 MiB
Net:   setup_iomux_enet: variante = 112 rob=0
FEC [PRIME]
Error: FEC address not set.

Hit any key to stop autoboot:  0
Boardinfo: OK, complete.
Attempting usb boot...
starting USB...
USB0:  USB EHCI 1.00
scanning bus 0 for devices... unable to get device descriptor (error=-1)
1 USB Device(s) found
USB1:  USB EHCI 1.00
scanning bus 1 for devices... 2 USB Device(s) found
       scanning usb for storage devices... 1 Storage Device(s) found

Device 0: Vendor: Intenso  Rev: 8.07 Prod: Alu Line
          Type: Removable Hard Disk
          Capacity: 3840.0 MB = 3.7 GB (7864320 x 512)
... is now current device
Loading environment (uEnv.txt) from USB ...
** Unable to read file boot/uEnv.txt **
Loading linux image (boot/zImage) from USB ...
** Unrecognized filesystem type **
ERR: Load image(s) from USB failed
Attempting mmc boot...
switch to partitions #0, OK
mmc0 is current device
Loading environment (uEnv.txt) from MMC0 ...
** Unable to read file uEnv.txt **
Loading linux image (boot/zImage) from MMC0 ...
7077816 bytes read in 517 ms (13.1 MiB/s)
Booting from MMC0 ...
Loading FDT image (boot/msc-nr-imx6-112-014-800x480-lcd.dtb) from MMC0 ...
48724 bytes read in 56 ms (849.6 KiB/s)
## Flattened Device Tree blob at 18000000
   Booting using the fdt blob at 0x18000000
   Using Device Tree in place at 18000000, end 1800ee53
```

Starting kernel ...

```
[ 0.000000] Booting Linux on physical CPU 0x0
[ 0.000000] Linux version 4.14.0-yocto-standard (oe-user@oe-host) (gcc version 8.3.0 (GCC))
) #1 SMP Thu Dec 12 07:22:58 UTC 2019
[ 0.000000] CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c5387d
[ 0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
[ 0.000000] OF: fdt: Machine model: MSC nanoRISC i.MX6SOLO
[ 0.000000] Memory policy: Data cache writeback
[ 0.000000] cma: Reserved 64 MiB at 0x4c000000
[ 0.000000] CPU: All CPU(s) started in SVC mode.
[ 0.000000] random: fast init done
[ 0.000000] percpu: Embedded 16 pages/cpu @ef7b3000 s34956 r8192 d22388 u65536
[ 0.000000] Built 1 zonelists, mobility grouping on. Total pages: 260608
[ 0.000000] Kernel command line: console=ttyMXC1,115200 root=/dev/mmcblk0p2 rootwait rw
board_name=msc-nR-imx6 board_variant=112-014 serial_number=1002477607 uboot_version=<U-
Boot 2018.03-+g26c8f2bad6 (Nov 22 2019 - 16:53:22 +0000)> consoleblank=0
[ 0.000000] PID hash table entries: 4096 (order: 2, 16384 bytes)
[ 0.000000] Dentry cache hash table entries: 131072 (order: 7, 524288 bytes)
[ 0.000000] Inode-cache hash table entries: 65536 (order: 6, 262144 bytes)
[ 0.000000] Memory: 958444K/1048576K available (9216K kernel code, 471K rwdma, 3256K
rodata, 1024K init, 435K bss, 24596K reserved, 65536K cma-reserved, 196608K highmem)
[ 0.000000] Virtual kernel memory layout:
[ 0.000000] vector : 0xffff0000 - 0xffff1000 ( 4 kB)
[ 0.000000] fixmap : 0xffc00000 - 0xffff0000 (3072 kB)
[ 0.000000] vmalloc : 0xf0800000 - 0xff800000 ( 240 MB)
[ 0.000000] lowmem : 0xc0000000 - 0xf0000000 ( 768 MB)
[ 0.000000] pkmap : 0xbfe00000 - 0xc0000000 ( 2 MB)
[ 0.000000] modules : 0xbf000000 - 0xbfe00000 ( 14 MB)
[ 0.000000] .text : 0xc0008000 - 0xc0a00000 (10208 kB)
[ 0.000000] .init : 0xc0e00000 - 0xc0f00000 (1024 kB)
[ 0.000000] .data : 0xc0f00000 - 0xc0f75e50 ( 472 kB)
[ 0.000000] .bss : 0xc0f77000 - 0xc0fe3f4c ( 436 kB)
[ 0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=2, Nodes=1
[ 0.000000] Hierarchical RCU implementation.
[ 0.000000] RCU event tracing is enabled.
[ 0.000000] RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=2.
[ 0.000000] RCU: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=2
[ 0.000000] NR_IRQS: 16, nr_irqs: 16, preallocated irq: 16
[ 0.000000] L2C-310 erratum 769419 enabled
[ 0.000000] L2C-310 enabling early BRESP for Cortex-A9
[ 0.000000] L2C-310 full line of zeros enabled for Cortex-A9
[ 0.000000] L2C-310 ID prefetch enabled, offset 16 lines
[ 0.000000] L2C-310 dynamic clock gating enabled, standby mode enabled
[ 0.000000] L2C-310 cache controller enabled, 16 ways, 512 kB
[ 0.000000] L2C-310: CACHE_ID 0x410000c8, AUX_CTRL 0x76450001
[ 0.000000] Switching to timer-based delay loop, resolution 333ns
[ 0.000007] sched_clock: 32 bits at 3000kHz, resolution 333ns, wraps every 715827882841ns
[ 0.000024] clocksource: mxc_timer1: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
637086815595 ns
[ 0.001721] Console: colour dummy device 80x30
[ 0.001752] Calibrating delay loop (skipped), value calculated using timer frequency..
6.00 BogoMIPS (lpj=30000)
[ 0.001768] pid_max: default: 32768 minimum: 301
[ 0.001900] Mount-cache hash table entries: 2048 (order: 1, 8192 bytes)
[ 0.001923] Mountpoint-cache hash table entries: 2048 (order: 1, 8192 bytes)
[ 0.002490] CPU: Testing write buffer coherency: ok
[ 0.002841] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[ 0.003211] Setting up static identity map for 0x10100000 - 0x10100078
[ 0.003341] Hierarchical SRCU implementation.
[ 0.007508] smp: Bringing up secondary CPUs ...
[ 0.007961] smp: Brought up 1 node, 1 CPU
[ 0.007975] SMP: Total of 1 processors activated (6.00 BogoMIPS).
[ 0.007983] CPU: All CPU(s) started in SVC mode.
[ 0.008837] devtmpfs: initialized
[ 0.018351] VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 4
[ 0.018698] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
19112604462750000 ns
[ 0.018722] futex hash table entries: 512 (order: 3, 32768 bytes)
[ 0.021070] pinctrl core: initialized pinctrl subsystem
[ 0.021980] NET: Registered protocol family 16
[ 0.024337] DMA: preallocated 256 KiB pool for atomic coherent allocations
[ 0.025162] cpuidle: using governor menu
[ 0.025543] CPU identified as i.MX6DL, silicon rev 1.1
[ 0.032458] vdd1p1: supplied by regulator-dummy
[ 0.032952] vdd3p0: supplied by regulator-dummy
[ 0.033370] vdd2p5: supplied by regulator-dummy
[ 0.033812] vddarm: supplied by regulator-dummy
[ 0.034287] vddpu: supplied by regulator-dummy
[ 0.034725] vddsoc: supplied by regulator-dummy
[ 0.045052] No ATAGs?
[ 0.045112] hw-breakpoint: found 5 (+1 reserved) breakpoint and 1 watchpoint registers.
[ 0.045126] hw-breakpoint: maximum watchpoint size is 4 bytes.
[ 0.046691] imx6dl-pinctrl 20e0000.iomuxc: initialized IMX pinctrl driver
[ 0.066053] GPIO line 166 (u0405-en) hogged as output/high
[ 0.070676] mxs-dma 110000.dma-apbh: initialized
[ 0.073699] vgaarb: loaded
[ 0.074248] SCSI subsystem initialized
```

```

[ 0.074686] usbcore: registered new interface driver usbfs
[ 0.074738] usbcore: registered new interface driver hub
[ 0.074833] usbcore: registered new device driver usb
[ 0.075989] i2c i2c-0: IMX I2C adapter registered
[ 0.076009] i2c i2c-0: can't use DMA, using PIO instead.
[ 0.076624] i2c i2c-1: IMX I2C adapter registered
[ 0.076644] i2c i2c-1: can't use DMA, using PIO instead.
[ 0.076832] media: Linux media interface: v0.10
[ 0.076878] Linux video capture interface: v2.00
[ 0.076931] pps_core: LinuxPPS API ver. 1 registered
[ 0.076939] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <
giometti@linux.it>
[ 0.076959] PTP clock support registered
[ 0.077218] Advanced Linux Sound Architecture Driver Initialized.
[ 0.078131] Bluetooth: Core ver 2.22
[ 0.078175] NET: Registered protocol family 31
[ 0.078184] Bluetooth: HCI device and connection manager initialized
[ 0.078199] Bluetooth: HCI socket layer initialized
[ 0.078212] Bluetooth: L2CAP socket layer initialized
[ 0.078259] Bluetooth: SCO socket layer initialized
[ 0.079348] clocksource: Switched to clocksource mxc_timer1
[ 0.079450] VFS: Disk quotas dquot_6.6.0
[ 0.079514] VFS: Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
[ 0.087903] NET: Registered protocol family 2
[ 0.088484] TCP established hash table entries: 8192 (order: 3, 32768 bytes)
[ 0.088568] TCP bind hash table entries: 8192 (order: 4, 65536 bytes)
[ 0.088701] TCP: Hash tables configured (established 8192 bind 8192)
[ 0.088885] UDP hash table entries: 512 (order: 2, 16384 bytes)
[ 0.088926] UDP-Lite hash table entries: 512 (order: 2, 16384 bytes)
[ 0.089099] NET: Registered protocol family 1
[ 0.089551] RPC: Registered named UNIX socket transport module.
[ 0.089562] RPC: Registered udp transport module.
[ 0.089569] RPC: Registered tcp transport module.
[ 0.089576] RPC: Registered tcp NFSv4.1 backchannel transport module.
[ 0.090586] hw perfevents: no interrupt-affinity property for /soc/pmu, guessing.
[ 0.090750] hw perfevents: enabled with armv7_cortex_a9 PMU driver, 7 counters available
[ 0.092407] workingset: timestamp_bits=30 max_order=18 bucket_order=0
[ 0.096753] NFS: Registering the id_resolver key type
[ 0.096789] Key type id_resolver registered
[ 0.096797] Key type id_legacy registered
[ 0.096841] jffs2: version 2.2. (NAND) (C) 2001-2006 Red Hat, Inc.
[ 0.097276] fuse init (API version 7.26)
[ 0.102232] bounce: pool size: 64 pages
[ 0.102259] io scheduler noop registered
[ 0.102268] io scheduler deadline registered
[ 0.102483] io scheduler cfq registered (default)
[ 0.102494] io scheduler mq-deadline registered
[ 0.102503] io scheduler kyber registered
[ 0.104570] OF: PCI: host bridge /soc/pcie@1ffc000 ranges:
[ 0.104604] OF: PCI: IO 0x01f80000..0x01f8ffff -> 0x00000000
[ 0.104621] OF: PCI: MEM 0x01000000..0x01efffff -> 0x01000000
[ 0.319435] imx6q-pcie 1ffc000.pcie: link up
[ 0.319451] imx6q-pcie 1ffc000.pcie: Link: Gen2 disabled
[ 0.319463] imx6q-pcie 1ffc000.pcie: Link up, Gen1
[ 0.319645] imx6q-pcie 1ffc000.pcie: PCI host bridge to bus 0000:00
[ 0.319662] pci_bus 0000:00: root bus resource [bus 00-ff]
[ 0.319674] pci_bus 0000:00: root bus resource [io 0x0000-0xffff]
[ 0.319686] pci_bus 0000:00: root bus resource [mem 0x01000000-0x01efffff]
[ 0.320045] PCI: bus0: Fast back to back transfers disabled
[ 0.320968] pci 0000:01:00.0: disabling ASPM on pre-1.1 PCIe device. You can enable it )
with 'pcie_aspm=force'
[ 0.321066] PCI: bus1: Fast back to back transfers disabled
[ 0.321090] pci 0000:01:00.0: bridge configuration invalid ([bus 00-00]), reconfiguring
[ 0.322392] PCI: bus2: Fast back to back transfers enabled
[ 0.322441] pci_bus 0000:02: [bus 02] partially hidden behind bridge 0000:01 [bus 01]
[ 0.322464] pci 0000:00:00.0: bridge has subordinate 01 but max busn 02
[ 0.322575] pci 0000:00:00.0: BAR 0: assigned [mem 0x01000000-0x010fffff]
[ 0.322592] pci 0000:00:00.0: BAR 6: assigned [mem 0x01100000-0x0110ffff pref]
[ 0.322603] pci 0000:00:00.0: BAR 7: assigned [io 0x1000-0x1fff]
[ 0.322620] pci 0000:01:00.0: BAR 7: assigned [io 0x1000-0x1fff]
[ 0.322634] pci 0000:02:00.0: BAR 5: assigned [io 0x1000-0x100f]
[ 0.322666] pci 0000:02:00.0: BAR 0: assigned [io 0x1010-0x1017]
[ 0.322694] pci 0000:02:00.0: BAR 1: assigned [io 0x1018-0x101f]
[ 0.322722] pci 0000:02:00.0: BAR 2: assigned [io 0x1020-0x1027]
[ 0.322750] pci 0000:02:00.0: BAR 3: assigned [io 0x1028-0x102f]
[ 0.322780] pci 0000:02:00.0: BAR 4: assigned [io 0x1030-0x1037]
[ 0.322811] pci 0000:01:00.0: PCI bridge to [bus 02]
[ 0.322832] pci 0000:01:00.0: bridge window [io 0x1000-0x1fff]
[ 0.322921] pci 0000:00:00.0: PCI bridge to [bus 01]
[ 0.322933] pci 0000:00:00.0: bridge window [io 0x1000-0x1fff]
[ 0.323299] pcieport 0000:00:00.0: Signaling PME with IRQ 298
[ 0.323514] pcieport 0000:00:00.0: AER enabled with IRQ 298
[ 0.324797] pwm-backlight lcd_backlight: lcd_backlight supply power not found, using dummy)
regulator
[ 0.326480] imx-sdma 20ec000.sdma: external firmware not found, using ROM firmware
[ 0.329820] imx-pgc-pd imx-pgc-power-domain.0: Linked as a consumer to 20dc000.gpc
[ 0.330089] imx-pgc-pd imx-pgc-power-domain.1: Linked as a consumer to 20dc000.gpc
[ 0.331211] 2020000.serial: ttymxc0 at MMIO 0x2020000 (irq = 26, base_baud = 5000000) is a)
IMX

```

```

[ 0.332015] 21e8000.serial: ttyMX1 at MMIO 0x21e8000 (irq = 70, base_baud = 5000000) is a)
IMX
[ 1.305229] console [ttyMX1] enabled
[ 1.309578] 21ec000.serial: ttyMX2 at MMIO 0x21ec000 (irq = 71, base_baud = 5000000) is a)
IMX
[ 1.324597] panel_simple_platform_probe: found: amfire,am800480stmqw00
[ 1.331203] panel-simple panel-lcd: panel-lcd supply power not found, using dummy )
regulator
[ 1.341612] etnaviv gpu-subsystem: bound 134000.gpu (ops gpu_ops)
[ 1.347807] etnaviv gpu-subsystem: bound 130000.gpu (ops gpu_ops)
[ 1.353950] etnaviv-gpu 134000.gpu: model: GC320, revision: 5007
[ 1.412500] etnaviv-gpu 130000.gpu: model: GC880, revision: 5106
[ 1.462947] [drm] Initialized etnaviv 1.1.0 20151214 for gpu-subsystem on minor 0
[ 1.472199] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[ 1.478823] [drm] No driver support for vblank timestamp query.
[ 1.484973] imx-drm display-subsystem: bound imx-ipuv3-crtc.2 (ops ipu_crtc_ops)
[ 1.492548] imx-drm display-subsystem: bound imx-ipuv3-crtc.3 (ops ipu_crtc_ops)
[ 1.500135] imx-drm display-subsystem: bound display@di0 (ops imx_pd_ops)
[ 1.564022] Console: switching to colour frame buffer device 100x30
[ 1.586729] imx-drm display-subsystem: fb0: frame buffer device
[ 1.593110] [drm] Initialized imx-drm 1.0.0 20120507 for display-subsystem on minor 1
[ 1.601037] imx-ipuv3 2400000.ipu: IPUv3H probed
[ 1.613998] brd: module loaded
[ 1.625998] loop: module loaded
[ 1.632626] spi_imx 2008000.ecspi: set DMA-mode
[ 1.637858] spi_imx 2008000.ecspi: probed
[ 1.642262] spi_imx 2014000.ecspi: set PIO-mode
[ 1.647435] spi_imx 2014000.ecspi: probed
[ 1.652667] m25p80 spi4.0: m25p16 (2048 Kbytes)
[ 1.657235] 4 ofpart partitions found on MTD device spi4.0
[ 1.662780] Creating 4 MTD partitions on "spi4.0":
[ 1.667596] 0x000000000000-0x000000020000 : "SPL"
[ 1.673288] 0x000000020000-0x000000040000 : "env"
[ 1.678818] 0x000000040000-0x0000000c0000 : "uboot"
[ 1.684534] 0x0000000c0000-0x000000200000 : "user"
[ 1.690763] libphy: Fixed MDIO Bus: probed
[ 1.695220] CAN device driver interface
[ 1.700081] flexcan 2090000.flexcan: device registered (reg_base=f1064000, irq=32)
[ 1.709273] fec 2188000.ethernet: 2188000.ethernet supply phy not found, using dummy )
regulator
[ 1.718379] pps pps0: new PPS source ptp0
[ 1.723207] fec 2188000.ethernet (unnamed net_device) (uninitialized): Invalid MAC address)
: 00:00:00:00:00:00
[ 1.733197] fec 2188000.ethernet (unnamed net_device) (uninitialized): Using random MAC )
address: d2:f0:43:f7:ea:17
[ 1.751341] libphy: fec_enet_mii_bus: probed
[ 1.756052] fec 2188000.ethernet eth0: registered PHC device 0
[ 1.762301] e1000e: Intel(R) PRO/1000 Network Driver - 3.2.6-k
[ 1.768139] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[ 1.774215] igb: Intel(R) Gigabit Ethernet Network Driver - version 5.4.0-k
[ 1.781240] igb: Copyright (c) 2007-2014 Intel Corporation.
[ 1.787572] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[ 1.794178] ehci-pci: EHCI PCI platform driver
[ 1.798715] ehci-mxc: Freescale On-Chip EHCI Host driver
[ 1.804435] usbcore: registered new interface driver usb-storage
[ 1.816256] ci_hdrc ci_hdrc.0: EHCI Host Controller
[ 1.821418] ci_hdrc ci_hdrc.0: new USB bus registered, assigned bus number 1
[ 1.849378] ci_hdrc ci_hdrc.0: USB 2.0 started, EHCI 1.00
[ 1.855534] hub 1-0:1.0: USB hub found
[ 1.859399] hub 1-0:1.0: 1 port detected
[ 1.867436] ci_hdrc ci_hdrc.1: EHCI Host Controller
[ 1.872614] ci_hdrc ci_hdrc.1: new USB bus registered, assigned bus number 2
[ 1.909379] ci_hdrc ci_hdrc.1: USB 2.0 started, EHCI 1.00
[ 1.915534] hub 2-0:1.0: USB hub found
[ 1.919329] hub 2-0:1.0: 1 port detected
[ 1.925908] i2c /dev entries driver
[ 1.943378] boarddata.c serial_number=1002477607
[ 1.952475] boarddata.c product_name=msc-nR-imx6
[ 1.958226] boarddata.c variant_key=112
[ 1.963200] boarddata.c feature_key=014
[ 1.968018] boarddata.c revision=
[ 1.978020] da9063 4-0058: Device detected (chip-ID: 0x61, var-ID: 0x50)
[ 2.023293] bio: Bringing 1240000uV into 1280000-1280000uV
[ 2.082004] input: da9063-onkey as /devices/soc0/soc/2100000.aips-bus/21a4000.i2c/i2c-1/)
i2c-4/4-0058/da9063-onkey/input/input0
[ 2.111692] da9063-rtc da9063-rtc: rtc core: registered da9063-rtc as rtc0
[ 2.125215] i2c i2c-1: Added multiplexed i2c bus 4
[ 2.132307] maxTouch 5-004a: Atmel maxTouch (Family mXT224E (81), Variant Calibrated (1)) )
Firmware version [2.0] Build 171
[ 2.143422] maxTouch 5-004a: Atmel maxTouch Configuration [X: 16] x [Y: 14]
[ 2.150676] input: Atmel maxTouch Touchscreen controller as /devices/soc0/soc/2100000.aips)
-bus/21a4000.i2c/i2c-1/i2c-5/5-004a/input/input1
[ 2.187651] maxTouch 5-004a: irq is 78.
[ 2.191697] i2c i2c-1: Added multiplexed i2c bus 5
[ 2.196536] i2c-mux-gpio 2100000.aips-bus:i2cmux: 2 port mux on 21a4000.i2c adapter
[ 2.204450] IR NEC protocol handler initialized
[ 2.209012] IR RC5(x/sz) protocol handler initialized
[ 2.214133] IR RC6 protocol handler initialized
[ 2.218691] IR JVC protocol handler initialized

```

```

[ 2.223264] IR Sony protocol handler initialized
[ 2.227906] IR SANYO protocol handler initialized
[ 2.232681] usb 1-1: new full-speed USB device number 2 using ci_hdrc
[ 2.239150] IR Sharp protocol handler initialized
[ 2.243934] IR MCE Keyboard/mouse protocol handler initialized
[ 2.249904] IR XMP protocol handler initialized
[ 2.254738] maXTouch 5-004a: maXTouch chip reset
[ 2.263336] da9063-battery da9063-battery: Iset=6000uA, Vset=3100mV
[ 2.271205] imx2-wdt 20bc000.wdog: timeout 60 sec (nowayout=0)
[ 2.277595] Bluetooth: HCI UART driver ver 2.3
[ 2.282111] Bluetooth: HCI UART protocol H4 registered
[ 2.287724] sdhci: Secure Digital Host Controller Interface driver
[ 2.293938] sdhci: Copyright(c) Pierre Ossman
[ 2.298319] sdhci-pltfm: SDHCI platform and OF driver helper
[ 2.369403] mmc0: SDHCI controller on 2190000.usdhc [2190000.usdhc] using ADMA
[ 2.382598] sdhci-esdhc-imx 2194000.usdhc: Got CD GPIO
[ 2.387821] sdhci-esdhc-imx 2194000.usdhc: Got WP GPIO
[ 2.439160] mmc0: host does not support reading read-only switch, assuming write-enable
[ 2.453308] mmc0: new high speed SDHC card at address 59b4
[ 2.459234] mmcblk0: mmc0:59b4 SS04G 3.69 GiB
[ 2.463991] mmc1: SDHCI controller on 2194000.usdhc [2194000.usdhc] using ADMA
[ 2.472676] mmcblk0: p1 p2
[ 2.480381] sdhci-esdhc-imx 2198000.usdhc: could not get ultra high speed state, work on
normal mode
[ 2.543317] mmc1: new high speed SDHC card at address 59b4
[ 2.549221] mmcblk1: mmc1:59b4 SDC 3.75 GiB (ro)
[ 2.554085] mmc2: SDHCI controller on 2198000.usdhc [2198000.usdhc] using ADMA
[ 2.564893] mmcblk1: p1
[ 2.572746] caam 2100000.caam: Entropy delay = 3200
[ 2.638420] caam 2100000.caam: Instantiated RNG4 SH0
[ 2.699182] caam 2100000.caam: Instantiated RNG4 SH1
[ 2.704309] caam 2100000.caam: device ID = 0x0a16010000000100 (Era 4)
[ 2.710782] caam 2100000.caam: job rings = 2, qi = 0, dpaa2 = no
[ 2.731473] caam algorithms registered in /proc/crypto
[ 2.740300] caam_jr 2101000.jr0: registering rng-caam
[ 2.746049] usbcore: registered new interface driver usbhid
[ 2.751729] usbhid: USB HID core driver
[ 2.758322] imx-media: Registered subdev ipul_vdic
[ 2.763398] imx-media: Registered subdev ipu2_vdic
[ 2.768395] imx-media: Registered subdev ipul_ic_prp
[ 2.773631] imx-media: Registered subdev ipul_ic_prpenc
[ 2.779103] ipul_ic_prpenc: Registered ipul_ic_prpenc capture as /dev/video0
[ 2.786287] imx-media: Registered subdev ipul_ic_prpvf
[ 2.791653] ipul_ic_prpvf: Registered ipul_ic_prpvf capture as /dev/video1
[ 2.798693] imx-media: Registered subdev ipu2_ic_prp
[ 2.803823] imx-media: Registered subdev ipu2_ic_prpenc
[ 2.810281] ipu2_ic_prpenc: Registered ipu2_ic_prpenc capture as /dev/video2
[ 2.817502] imx-media: Registered subdev ipu2_ic_prpvf
[ 2.822881] ipu2_ic_prpvf: Registered ipu2_ic_prpvf capture as /dev/video3
[ 2.830038] imx-media: Registered subdev ipul_csi0
[ 2.835106] ipul_csi0: Registered ipul_csi0 capture as /dev/video4
[ 2.841466] imx-media: Registered subdev ipul_csi1
[ 2.846515] ipul_csi1: Registered ipul_csi1 capture as /dev/video5
[ 2.862552] mmc2: new DDR MMC card at address 0001
[ 2.867893] mmcblk2: mmc2:0001 H4G1d 3.64 GiB
[ 2.872743] mmcblk2boot0: mmc2:0001 H4G1d partition 1 4.00 MiB
[ 2.878911] mmcblk2boot1: mmc2:0001 H4G1d partition 2 4.00 MiB
[ 2.885101] mmcblk2rpm: mmc2:0001 H4G1d partition 3 4.00 MiB
[ 2.894263] fsl-ssi-dai 2028000.ssi: No cache defaults, reading back from HW
[ 2.901544] mmcblk2: p1 p2
[ 2.914935] imx-tlv320aic3107 sound: tlv320aic3x-hifi <-> 2028000.ssi mapping ok
[ 2.924150] imx-tlv320aic3107 sound: snd_soc_register_card tlv320aic3x-hifi success
[ 2.932681] NET: Registered protocol family 10
[ 2.938136] Segment Routing with IPv6
[ 2.941939] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[ 2.948450] NET: Registered protocol family 17
[ 2.952947] can: controller area network core (rev 20170425 abi 9)
[ 2.959214] NET: Registered protocol family 29
[ 2.963790] can: raw protocol (rev 20170425)
[ 2.968068] can: broadcast manager protocol (rev 20170425 t)
[ 2.973766] can: netlink gateway (rev 20170425) max_hops=1
[ 2.979536] Key type dns_resolver registered
[ 2.985172] Registering SWP/SWPB emulation handler
[ 3.001107] imx-media: Registered subdev ipul_csi0_mux
[ 3.006683] imx-media: Registered subdev ipul_csi1_mux
[ 3.011980] imx-media: imx_media_create_link: ipul_csi0:1 -> ipul_ic_prp:0
[ 3.018866] imx-media: imx_media_create_link: ipul_csi0:1 -> ipul_vdic:0
[ 3.025625] imx-media: imx_media_create_link: ipul_csi0_mux:5 -> ipul_csi0:0
[ 3.032719] imx-media: imx_media_create_link: ipul_csi1:1 -> ipul_ic_prp:0
[ 3.039638] imx-media: imx_media_create_link: ipul_csi1:1 -> ipul_vdic:0
[ 3.046362] imx-media: imx_media_create_link: ipul_csi1_mux:5 -> ipul_csi1:0
[ 3.053437] imx-media: imx_media_create_link: ipul_vdic:2 -> ipul_ic_prp:0
[ 3.060335] imx-media: imx_media_create_link: ipul_ic_prp:1 -> ipul_ic_prpenc:0
[ 3.067667] imx-media: imx_media_create_link: ipul_ic_prp:2 -> ipul_ic_prpvf:0
[ 3.074915] imx-media: imx_media_create_link: ipu2_vdic:2 -> ipu2_ic_prp:0
[ 3.081828] imx-media: imx_media_create_link: ipu2_ic_prp:1 -> ipu2_ic_prpenc:0
[ 3.089144] imx-media: imx_media_create_link: ipu2_ic_prp:2 -> ipu2_ic_prpvf:0
[ 3.099001] imx_thermal 2000000.aips-bus:tempmon: Industrial CPU temperature grade - max)
:105C critical:100C passive:95C

```

```

[ 3.112910] da9063-rtc da9063-rtc: setting system clock to 2019-12-12 07:35:48 UTC ]
(1576136148)
[ 3.122190] ALSA device list:
[ 3.125170] #0: imx-tlv320aic3107
[ 3.137760] EXT4-fs (mmcblk0p2): couldn't mount as ext3 due to feature incompatibilities
[ 3.211704] EXT4-fs (mmcblk0p2): recovery complete
[ 3.219553] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
[ 3.227692] VFS: Mounted root (ext4 filesystem) on device 179:2.
[ 3.234045] devtmpfs: mounted
[ 3.238983] Freeing unused kernel memory: 1024K
INIT: version 2.88 booting[ 3.379394] usb 2-1: new high-speed USB device number 2 using ]
ci_hdrc

[ 3.588183] usb-storage 2-1:1.0: USB Mass Storage device detected
[ 3.595515] scsi host0: usb-storage 2-1:1.0
Starting udev
[ 3.833058] udevd[157]: starting version 3.2.7
[ 3.905127] udevd[158]: starting eudev-3.2.7
[ 4.314666] at25 spi0.0: 8 KByte at25 eeprom, pagesize 32
[ 4.367024] at25 spi4.1: 8 KByte at25 eeprom, pagesize 32
[ 4.490792] coda 2040000.vpu: Firmware code revision: 570363
[ 4.496502] coda 2040000.vpu: Initialized CODA960.
[ 4.501387] coda 2040000.vpu: Firmware version: 3.1.1
[ 4.619922] coda 2040000.vpu: codec registered as /dev/video[6-7]
[ 4.674973] scsi 0:0:0:0: Direct-Access Intenso Alu Line 8.07 PQ: 0 ANSI: 4
[ 4.726060] sd 0:0:0:0: [sda] 7864320 512-byte logical blocks: (4.03 GB/3.75 GiB)
[ 4.794594] sd 0:0:0:0: [sda] Write Protect is off
[ 4.833684] sd 0:0:0:0: [sda] Write cache: disabled, read cache: enabled, doesn't support ]
DPO or FUA
[ 4.934869] sda: sda1
[ 4.972655] sd 0:0:0:0: [sda] Attached SCSI removable disk
[ 5.123768] user_gpios: loading out-of-tree module taints kernel.
[ 7.342937] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be ]
corrupt. Please run fsck.
[ 7.539005] EXT4-fs (mmcblk2p2): recovery complete
[ 7.581059] EXT4-fs (mmcblk2p2): mounted filesystem with ordered data mode. Opts: (null)
[ 7.692339] FAT-fs (mmcblk2p1): Volume was not properly unmounted. Some data may be ]
corrupt. Please run fsck.
[ 8.616188] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt. ]
Please run fsck.
[ 8.708216] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
ALSA: Restoring mixer settings...
INIT: Entering runlevel: 5
No state is present for card imx-tlv320aic310
Found hardware: "imx-tlv320aic31" "" "" "" ""
Hardware is initialized using a generic method
No state is present for card imx-tlv320aic310
Configuring network interfaces... done.
Starting system message bus: dbus.
Starting OpenBSD Secure Shell server: sshd
done.
Starting rpcbind daemon...done.
starting statd: done
Starting bluetooth: bluetoothd.
Starting HOSTAP Daemon: Configuration file: /etc/hostapd.conf
Could not read interface wlan0 flags: No such device
nl80211: Driver does not support authentication/association or connect commands
nl80211: deinit ifname=wlan0 disabled_11b_rates=0
Could not read interface wlan0 flags: No such device
nl80211 driver initialization failed.
wlan0: interface state UNINITIALIZED->DISABLED
wlan0: AP-DISABLED
wlan0: CTRL-EVENT-TERMINATING
hostapd_free_hapd_data: Interface wlan0 wasn't started
Starting ntpd: done
Starting syslogd/klogd: done
* Starting Avahi mDNS/DNS-SD Daemon: avahi-daemon
...done.
Starting Telephony daemon
Starting Linux NFC daemon
Starting internet superserver: inetd.
Starting TCG TSS2 Access Broker and Resource Management daemon: device driver not loaded, ]
skipping.
Starting Trusted Computing daemon:
device driver not loaded, skipping.

MSC-LDK LOL99_20190718_V1_4_0-16-ge2e36f5 built on Thu Dec 12 07:29:21 UTC 2019 by dher@msc-]
aac-mint06
Poky (Yocto Project Reference Distro) 2.7 msc-nr-imx6x /dev/ttymxcl

msc-nr-imx6x login:

```

## 4.13. Login

Login is enabled via console or serial console (/dev/ttyMXC1, 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/ttyMXC1 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/0584/conf/local.conf:

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

Further information can be found here:

[http://www.yoctoproject.org/docs/3.0.1/ref-manual/ref-manual.html#var-IMAGE\\_INSTALL](http://www.yoctoproject.org/docs/3.0.1/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/0584/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=0584 --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/0584/conf/bblayers.conf` by these lines:

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

Then create the layer as described here:

<http://www.yoctoproject.org/docs/3.0.1/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=0584 --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/0584/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 be customized by setting the Yocto variable `DEFAULT_TIMEZONE` in `build/0584/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`.

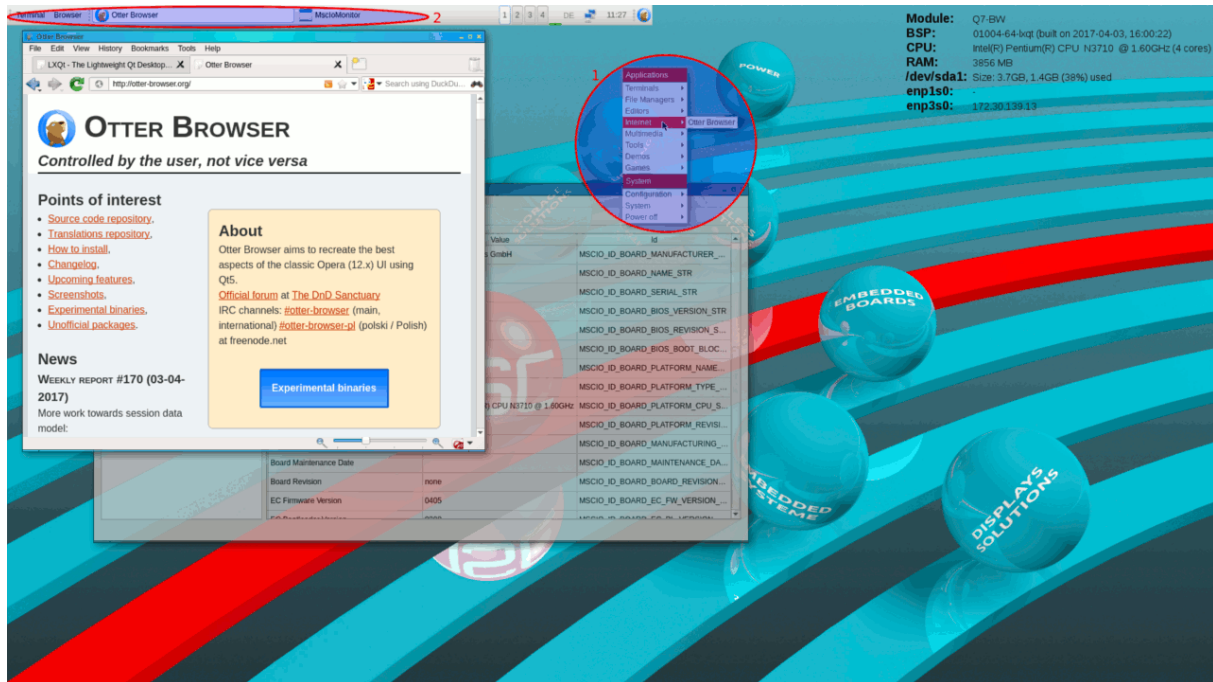
It is also possible to change the time zone of a running system to e.g. `New York`, using the following shell commands:

```
# 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.

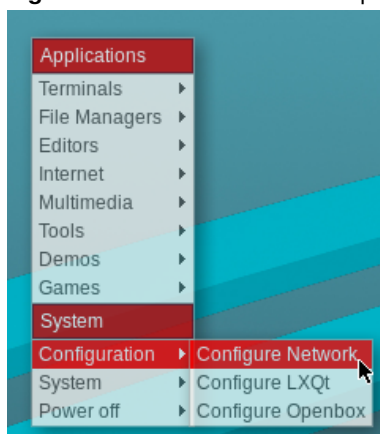
- Application Launcher (1) - If the right mouse button is pressed on the desktop, the pre-configured application list is opened for launching terminal, browser, editor etc.
- Window List (2) - The top panel lists the windows applications.

## 4.17. ConnMan Configuration

The internet connection daemon ConnMan (<https://01.org/connman>) is used in LXQt images to manage the internet connections.

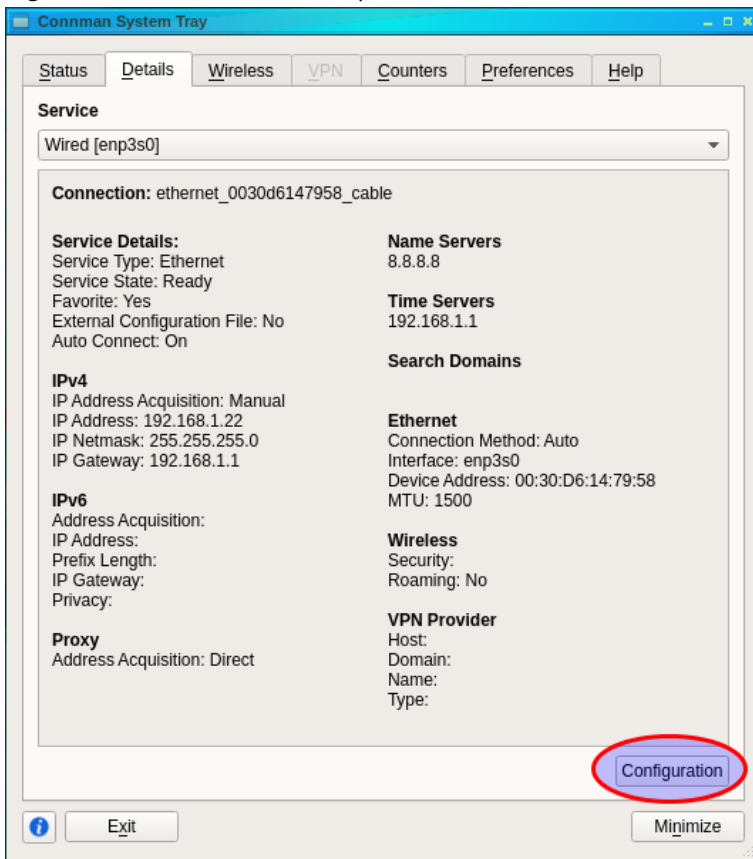
The ConnMan Property Editor can be launched using a right-click on the LXQt desktop:

Figure 4.3. – Start ConnMan Property Editor



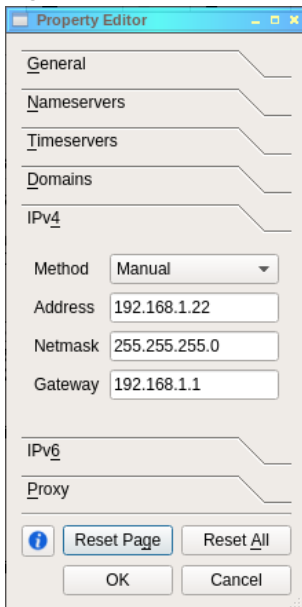
There is a detailed overview for each connection when the Details tab is selected:

**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:

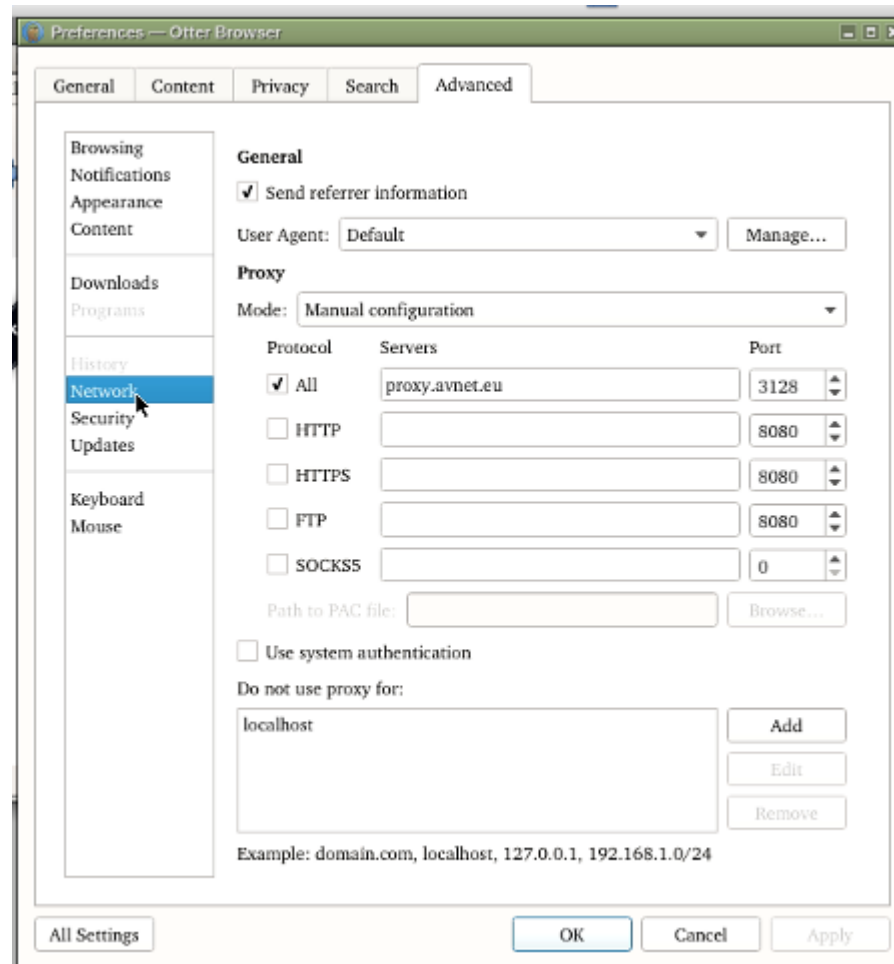
```
# connman-info
[/net/connman/service/ethernet_0044d6143d58_cable]
Name = Wired
IPv6 = {}
IPv4.Configuration = {Method=dhcp}
Nameservers = [192.168.1.49, 192.168.1.57]
IPv6.Configuration = {Method=off}
IPv4 = {Netmask=255.255.255.0, Method=dhcp, Gateway=192.168.1.201, Address=192.168.1.13}
Domains.Configuration = []
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).

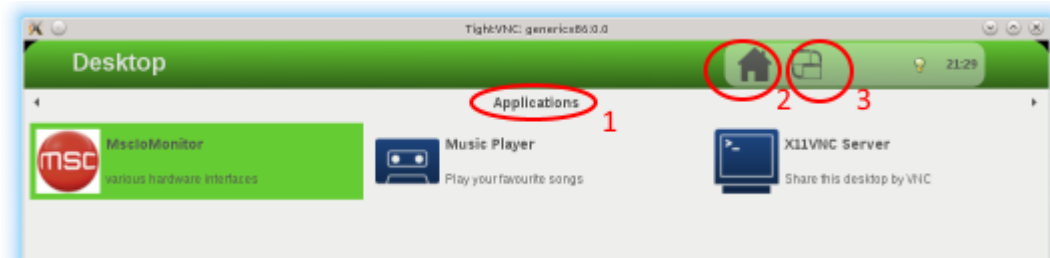
**Figure 4.6.** – Otter Browser - Proxy Setup



## 4.18. Using Sato

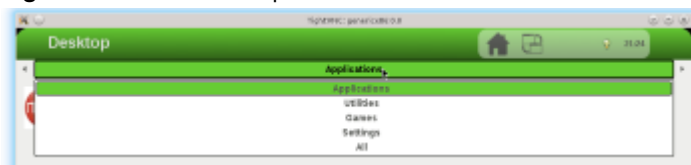
Sato is a window manager in which applications run full-screen, optimized for use with touch-displays.

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-ixqt-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:

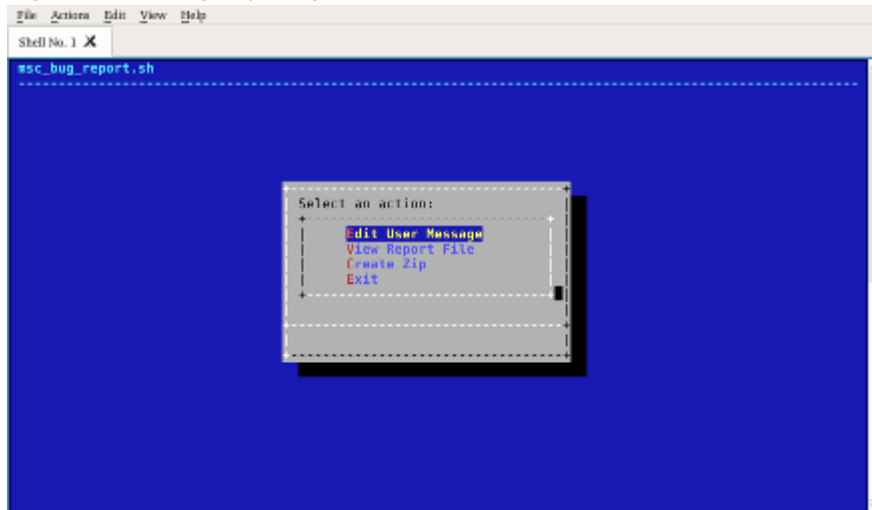
```
qmake hello_world.pro && make && ./hello_world
```

## 4.20. 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.

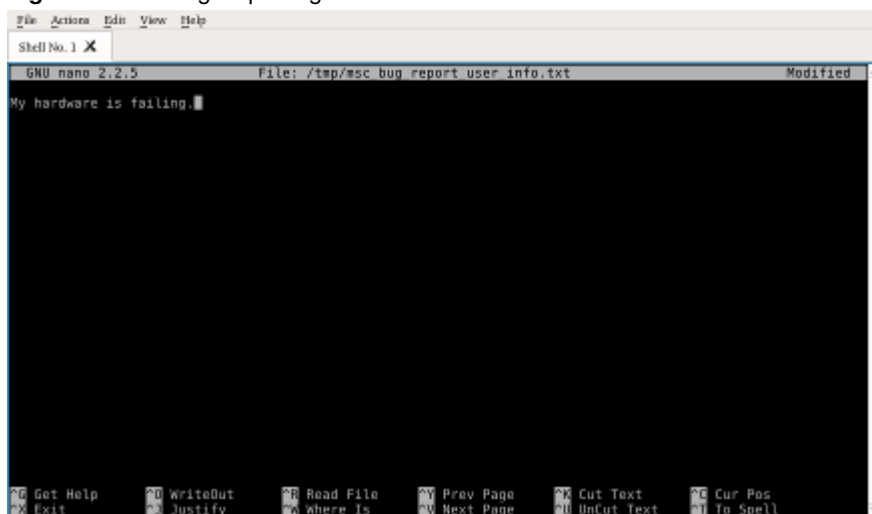
- Run `msc_bug_report.sh`.

**Figure 4.10.** – Bug Reporting - Start



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

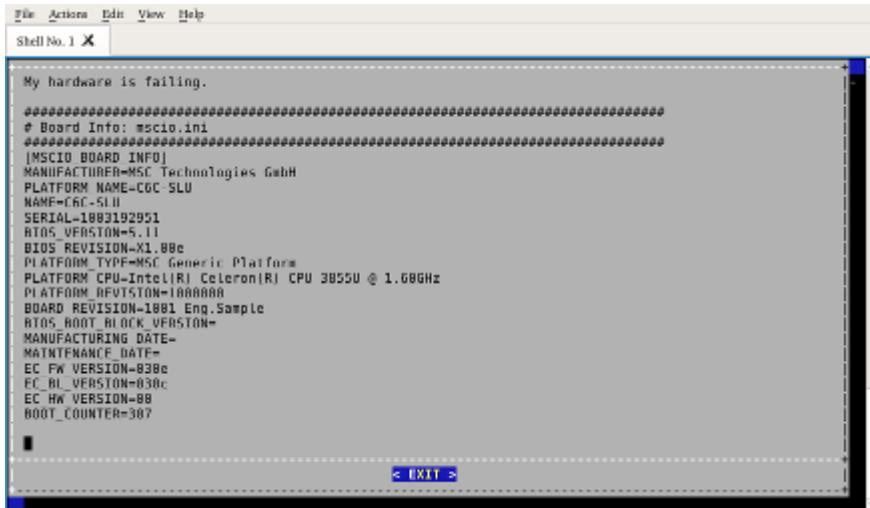
**Figure 4.11.** – Bug Reporting - Start



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

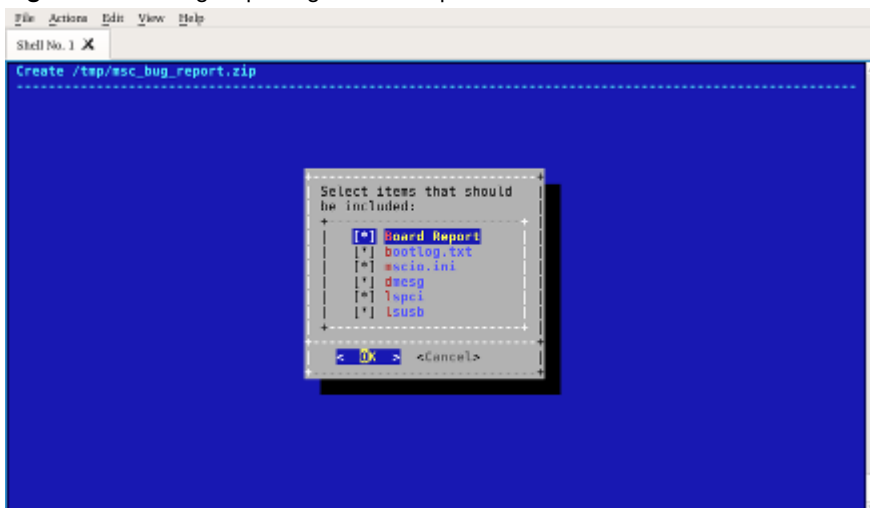


Figure 4.12. – Bug Reporting - View



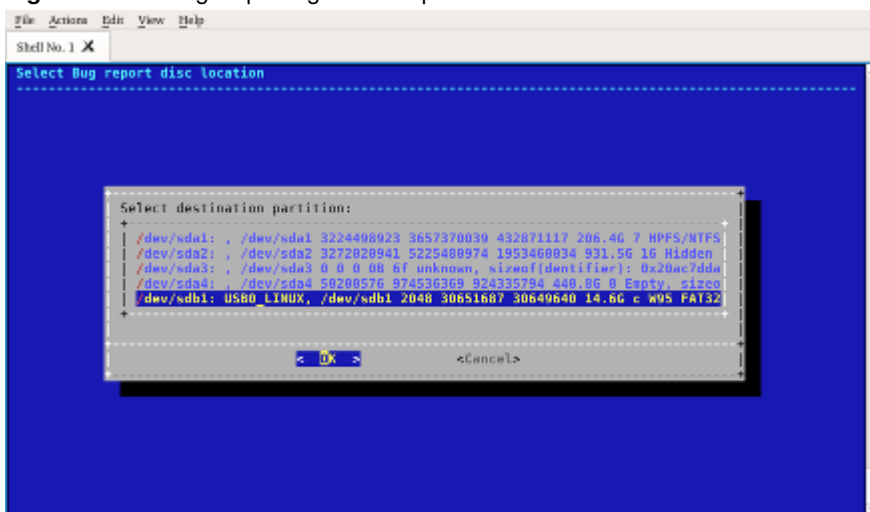
- Press “Create a zip file” and select the components you want to send (e.g. bootlog, mscio.ini, last kernel logs (dmesg) or the installed hardware).

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>)

## 4.21. 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 v1.6.0`) 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. mscio

This package provides mscio-setup and libMsclo 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`.

### 5.2. 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. [18](#)

**MSC-LDK** MSC Linux Development Kit. [14](#), [18](#)

# B. Bootloader Configuration

## B.1. Booting

It is possible to install the Linux images on a  $\mu$ SD-Card and to boot from.

Advantages are:

- The  $\mu$ SD Card can be modified externally - adding files and/or tools, applications
- The boot media can be setup by a windows host using Win32 Disk Imager
- It is easy to duplicate the image on several  $\mu$ SD-Cards

Disadvantages are:

- The boot media cannot be removed once Linux is running.
- the system should always be shut safely down, otherwise a file system damage may occur

To install, copy the appropriate .sdcard-file from:

```
msc-ldk/build/0584/tmp/deploy/images/msc-nr-imx6
```

to SD card device. Often you need root rights to have permission to write to devices.

```
user@devhost:msc-ldk$ sudo dd if=<name of image>.sdcard of=/dev/<sd-card-device> bs=4M
```

The option **bs=4M** tells **dd** to use 4MB blocks during copy. This optimizes the write speed to SD card.

After inserting the  $\mu$ SD card into the modules  $\mu$ SD Slot and powering the system it will boot:

```
This is an example booting an i.MX6Solo (BoardMark: IMX6D-112_014)
```

```
U-Boot SPL 2018.03-+g26c8f2bad6 (Nov 22 2019 - 16:53:22 +0000)
```

```
Boardinfo:
```

```
name ..... msc-nR-imx6
variant ..... 112
feature ..... 014
serial ..... 1002477607
revision ..... N/A
```

```
Trying to boot from MMC1
```

```
U-Boot 2018.03-+g26c8f2bad6 (Nov 22 2019 - 16:53:22 +0000)
```

```
CPU: Freescale i.MX6SOLO rev1.1 at 792MHz
```

```
CPU: Industrial temperature grade (-40C to 105C) at 47C
```

```
CPU: Reset cause is POR
```

```
Board: MSC nanoRISC i.MX6
```

```
I2C: ready
```

```
DRAM: 1 GiB
```

```
LCD, HDMI,
```

```
MMC: FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
```

```
Loading Environment from MMC... OK
```

```
Display: hdmi:1024x768M@60 (1024x768)
```

```
splashfile is already set to 'logos/msc-logo-1024x768.bmp'
```

```
In: serial
```

```
Out: serial
```

```
Err: serial
```

```
SF: Detected m25p16 with page size 256 Bytes, erase size 64 KiB, total 2 MiB
```

```
Net: setup_iomux_enet: variante = 112 rob=0
```

```
FEC [PRIME]
```

```
Error: FEC address not set.
```

```
Hit any key to stop autoboot: 0
```

```

Boardinfo: OK, complete.
Attempting usb boot...
starting USB...
USB0:   USB EHCI 1.00
scanning bus 0 for devices... unable to get device descriptor (error=-1)
1 USB Device(s) found
USB1:   USB EHCI 1.00
scanning bus 1 for devices... 2 USB Device(s) found
      scanning usb for storage devices... 1 Storage Device(s) found

Device 0: Vendor: Intenso  Rev: 8.07 Prod: Alu Line
          Type: Removable Hard Disk
          Capacity: 3840.0 MB = 3.7 GB (7864320 x 512)
... is now current device
Loading environment (uEnv.txt) from USB ...
** Unable to read file boot/uEnv.txt **
Loading linux image (boot/zImage) from USB ...
** Unrecognized filesystem type **
ERR: Load image(s) from USB failed
Attempting mmc boot...
switch to partitions #0, OK
mmc0 is current device
Loading environment (uEnv.txt) from MMC0 ...
** Unable to read file uEnv.txt **
Loading linux image (boot/zImage) from MMC0 ...
7077816 bytes read in 517 ms (13.1 MiB/s)
Booting from MMC0 ...
Loading FDT image (boot/mmc-nR-imx6-112-014-800x480-lcd.dtb) from MMC0 ...
48724 bytes read in 56 ms (849.6 KiB/s)
## Flattened Device Tree blob at 18000000
   Booting using the fdt blob at 0x18000000
   Using Device Tree in place at 18000000, end 1800ee53

Starting kernel ...

```

## B.2. Boardinfo

The Boardinfo is a set of data describing the specific system variant. Some hardware settings are derived from Boardinfo in U-Boot, e.g. DDR3 supply voltage, LVDS/TTL display Interface or eMMC/NAND configuration.

Example Boardinfo output from SPL in startup phase:

```

Boardinfo:
  name ..... msc-nR-imx6
  variant ..... 112
  feature ..... 014
  serial ..... 1002477607
  revision ..... N/A

```

## B.3. Configure Devicetree

Depending on the specific Hardware a different devicetree has to be used. By default U-Boot generates the name of the devicetree by boarddata [Boardinfo](#) entries in this way:

```
<board_name>-<variant>-<feature>-headless.dtb
```

If you want to enable any display the device tree must be changed. This has to be done from U-boot prompt. To see the list of all available device trees type the following command:

```
=> ls mmc 0:2 boot
```

Then all devicetrees are shown. Be aware that you select one devicetree that has the same variant and feature number as shown in boardinfo. Here is an example of setting an 800x480 Pixel LCD Panel for the given variant 143 and feature 014:

```

=> setenv fdtfile msc-nR-imx6-112-014-800x480-lcd.dtb
=> saveenv
Saving Environment to MMC...
Writing to MMC(0)... done
=>

```

## B.4. Update U-Boot and SPL

If you intend to update U-boot and/or SPL there are at least two ways:

- update yocto recipes and add patches / take another revision, build new .sdcardimage.
- generate U-Boot separated from yocto, update  $\mu$ SD Card manually

The first variant is described in Chapter Image Types.

For the second (but not recommended) variant, use this method:

- Copy the files SPL and u-boot.img to any accessible medium from Linux (USB-Stick, SD-Card or mounted Filesystem)
- Start the system.
- Log-in as root user.
- Change to the location where the above files are located
- # dd if=SPL of=/dev/mmcblk0 bs=1k skip=0 seek=1
- # dd if=u-boot.img of=/dev/mmcblk0 skip=0 seek=69
- reboot the system to use the new u-boot

## B.5. Change Display settings in U-Boot

Several displays are supported in U-Boot, and several interfaces are supported: HDMI, LVDS, RGB. The list of supported displays can be shown by this command:

```
=> fbpanel
      clock-frequency hactive vactive hback-porch hfront-porch vback-porch vfront-porch hsync-len vsync-len
hdmi 1024x768M@60:m24:64998375,1024,768,220,40,21,7,60,10
      64998375 1024 768 220 40 21 7 60 10
hdmi 1920x1080M@60:m24:148500148,1920,1080,148,88,36,4,44,5
      148500148 1920 1080 148 88 36 4 44 5
lvds NLT8048AC19-14F:m24:38170852,800,480,220,40,21,7,60,60
      38170852 800 480 220 40 21 7 60 60
lvds MITSUBISHI-AA150XT11-DE-01:m24:64998375,1024,768,220,40,21,7,60,10
      64998375 1024 768 220 40 21 7 60 10
lvds MITSUBISHI-AA121TH01:m24:71108582,1280,800,40,40,10,3,80,10
      71108582 1280 800 40 40 10 3 80 10
lvds MI1040GT:m24:46234222,800,600,220,40,21,7,60,60
      46234222 800 600 220 40 21 7 60 60
lcd AM800480STMQW00:24:33295598,800,480,128,128,22,22,1,1
      33295598 800 480 128 128 22 22 1 1
```

The output can be limited to a specific interface by adding the interface name [hdmi|lcd|lvds|lvds2]

```
=> fbpanel hdmi
      clock-frequency hactive vactive hback-porch hfront-porch vback-porch vfront-porch hsync-len vsync-len
hdmi 1024x768M@60:m24:64998375,1024,768,220,40,21,7,60,10
      64998375 1024 768 220 40 21 7 60 10
hdmi 1920x1080M@60:m24:148500148,1920,1080,148,88,36,4,44,5
      148500148 1920 1080 148 88 36 4 44 5
```

To select one of these given displays type for example:

```
=> fbpanel lcd AM800480STMQW00
```

This command takes all parameters from above table of line starting with lcd AM800480STMQW00.

It is possible to specify different resolutions and timings:

```
=> fbpanel lcd AM800480STMQW00:24:33295598,800,400,128,128,62,62,1,1
```

The settings of a specified display is stored in an environment variable:

```
fb_<interface_type>="*mode_str>:bpp:pixclkfreq,xres,yres,hback-porch,hfront-porch,vback-porch,vfront-
porch,hsync-len,vsync-len"
```

this results for the setting done above in this setting:

```
fb_lcd=*AM800480STMQW00:24:33295598,800,480,128,128,22,22,1,1
```

## B.6. Change Boot order

The universal Bootloader U-Boot is scriptable, so the boot is determined by the contents of `bootcmd`:

```
if boardinfo complete; then
    for btype in ${bootdevs};
        do echo Attempting ${btype} boot...;
            if run ${btype}_boot; then;
                exit;
            fi;
        done;
    else
        echo ERR: Aborting boot OS, boardinfo is not complete!; false; fi;
```

The above script shows that a for-loop iterating over `bootdevs` entries and expects a corresponding `<btype>_boot` script. The normal (default) sequence is:

```
bootdevs:  usb mmc emmc
```

And the predefined `<btype>_boot` scripts are:

```
usb_boot :  if usb start && usb dev 0; then setenv uenvpart ${usbfatpart};
             if run usbloadenv; then run importenv; fi;
             if test -n $uenvcmd; then echo Running uenvcmd ...;run uenvcmd; fi;
             else echo ERR: USB start failed; false; fi;

mmc_boot :  mmc dev ${mmcdev};
             if mmc rescan; then setenv uenvdev ${mmcdev}; setenv uenvpart ${mmcfatpart};
             if run mmcloadenv; then run importenv; fi;
             if test -n $uenvcmd; then echo Running uenvcmd ...;run uenvcmd; fi;
             if run mmcloadimage; then run mmcboot;
             else echo ERR: Load image(s) from MMC${uenvdev} failed; false; fi;
             else echo ERR: MMC scan failed; false; fi;

emmc_boot:  mmc dev ${emmcdev};
             if mmc rescan; then
             if run emmcloadenv; then run importenv; fi;
             if test -n $uenvcmd; then echo Running uenvcmd ...;run uenvcmd; fi;
             if run emmcloadimage; then run emmcboot;
             else echo ERR: Load image(s) from eMMC failed; false; fi;
             else echo ERR: eMMC scan failed; false; fi;
```

There are more than one mmc interfaces, so the `mmc_boot`-Script uses the environment variable `mmcdev` to determine the mmc to boot from. Hence here is also another option to change the boot device. On default, the  $\mu$ MMC-slot [0] on the module is set in `mmcdev`. Using the command `mmc list` the available mmc-devices will be shown:

```
=> mmc list
FSL_SDHC: 0 (SD)
FSL_SDHC: 1
FSL_SDHC: 2
```

Since all `*_boot` scripts are available, it is possible to change or restrict the boot order by redefining `bootdevs`. Followed by a `saveenv` and `reset` the changes take effect.



## C. Boarddata

Each board has an unique serial number and a specific variant and features. This information is stored inside an EEPROM on the module, which is programmed during board level test. The following two sections describe layout and access to the stored information.

### C.1. Boarddata EEPROM Layout

The layout of the stored information is described in this section. The information is stored in 80 bytes.

Offset	Length	Description
0	4	Magic
4	1	Major version number
5	1	Minor version number
6	2	Checksumme
8	8	Reserve
16	4	Feature Bits
20	32	Board name
52	4	Variant key
56	4	Feature key
60	12	Serial number
72	4	Boot counter
76	2	MES revision
78	2	Reserve

### C.2. Reading Boarddata

The stored information can be read out under Linux in sysfs. This feature is provided by `drivers/misc/boarddata.c`. The keys for the values are located in `/sys/class/boarddata`:

```
root@msc-nr-imx6:/sys/class/boarddata# ls -l
feature_key
product_name
revision
serial_number
variant_key
```

Each value can be printed by the command `cat`:

```
root@msc-nr-imx6:/sys/class/boarddata# cat feature_key
071
root@msc-nr-imx6:/sys/class/boarddata# cat product_name
msc-nr-imx6
root@msc-nr-imx6:/sys/class/boarddata# cat revision
ES
root@msc-nr-imx6:/sys/class/boarddata# cat serial_number
1003578759
root@msc-nr-imx6:/sys/class/boarddata# cat variant_key
143
```

# D. Devicetree Configuration

## D.1. Concept

The main concept of devicetrees is to abstract the different hardware configurations while using the same kernel. There are a lot of files in the folder `arch/arm/boot/dts`, you will recognize files with `.dts` and `.dtsi` suffix, and perhaps compiled files with `.dtb` ending. Device trees are organized hierarchically. The `.dts` (Device Tree Source) files are the topmost ones. They include the `.dtsi` (Device Tree Source Include) files, which can include another `.dtsi`.

## D.2. Structure used for nanoRISC

The i.MX6 nanoRISC uses the following devicetree files:

```
msc-nR-imx6-010-014-headless.dts
msc-nR-imx6-010-014-800x480-lcd.dts
msc-nR-imx6-010-014-hdmi.dts
msc-nR-imx6-010-014.dtsi

msc-nR-imx6-011-014-headless.dts
msc-nR-imx6-011-014-800x480-lcd.dts
msc-nR-imx6-011-014-hdmi.dts

msc-nR-imx6-112-014-headless.dts
msc-nR-imx6-112-014-800x480-lcd.dts
msc-nR-imx6-112-014-800x480-lvds.dts
msc-nR-imx6-112-014-1024x768-lvds.dts
msc-nR-imx6-112-014-1280x800-lvds.dts
msc-nR-imx6-112-014-hdmi.dts
msc-nR-imx6-112-014.dtsi

msc-nR-imx6-mb2-maxtouch.dtsi

msc-nR-imx6s.dtsi
msc-nR-imx6dl.dtsi
msc-nR-imx6sdl.dtsi
msc-nR-imx6d.dtsi
msc-nR-imx6q.dtsi
msc-nR-imx6qd.dtsi
msc-nR-imx6x.dtsi

msc-nR-imx6-123-014-headless.dts
msc-nR-imx6-123-014-800x480-lcd.dts
msc-nR-imx6-123-014-800x480-lvds.dts
msc-nR-imx6-123-014-1280x800-lvds.dts
msc-nR-imx6-123-014-hdmi.dts
msc-nR-imx6-123-014.dtsi

msc-nR-imx6-133-014-headless.dts
msc-nR-imx6-133-014-800x480-lcd.dts
msc-nR-imx6-133-014-800x480-lvds.dts
msc-nR-imx6-133-014-1024x768-lvds.dts
msc-nR-imx6-133-014-1280x800-lvds.dts
msc-nR-imx6-133-014-hdmi.dts
msc-nR-imx6-133-014.dtsi

msc-nR-imx6-143-014-headless.dts
msc-nR-imx6-143-014-800x480-lcd.dts
msc-nR-imx6-143-014-800x480-lvds.dts
msc-nR-imx6-143-014-1024x768-lvds.dts
msc-nR-imx6-143-014-1280x800-lvds.dts
msc-nR-imx6-143-014-hdmi.dts
msc-nR-imx6-143-014.dtsi

msc-nR-imx6-183-014-headless.dts
msc-nR-imx6-183-014-800x480-lcd.dts
msc-nR-imx6-183-014-1024x768-lvds.dts
msc-nR-imx6-183-014-1280x800-lvds.dts
msc-nR-imx6-183-014-hdmi.dts
```

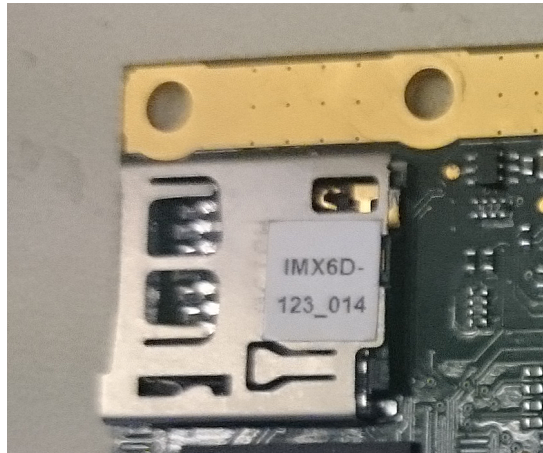
In the lower left corner of the above listing you can see some files which differ in naming scheme from the others, these are the CPU specific files, their names reflect the CPU-variant they are describing.

dtsi-file	Usage
msc-nR-imx6s.dtsi	only for I.MX6Solo
msc-nR-imx6dl.dtsi	only for I.MX6DL
msc-nR-imx6sdl.dtsi	for I.MX6Solo and I.MX6DL
msc-nR-imx6d.dtsi	only for I.MX6D
msc-nR-imx6q.dtsi	only for I.MX6Q
msc-nR-imx6qd.dtsi	for I.MX6Q and I.MX6D
msc-nR-imx6x.dtsi	for all i.MX6 variants

Just above the CPU variant specific files, there is the file: `msc-nR-imx6-mb2-maxtouch.dtsi`. This file contains the definitions for the Atmel MXT224 touch controller on the MB2 eval board. It is included in all variants except the headless variants.

The other files are named like `msc-nR-imx6-<type>-<variant>-<display>.dts`. These are the top-level device tree files, which define the display settings and include the corresponding base file `msc-nR-imx6-<type>-<variant>.dtsi`.

The values for <type> and <variant> must match to the board mark label on the module. The board mark defines the used CPU type and the design variant. Here is an example of a BoardMark label showing <type>=123 and <variant>=014:



**Figure D.1.** – example BoardMark

The boardmarks type number describes in each digit the temperature range, CPU-variant and memory configuration. Assuming the filenames have the format name-abc-def, then fields a-f will be explained in this table:

digit	value	description
a	0	commercial temperature range
	1	industrial temperature range
	2	extended commercial temperatur range
b	1	Solo
	2	Dual
	3	DualLite
	4	Quad
	6	Dual Plus
	8	Quad Plus
c	0	512MB NAND, 512MB RAM
	1	4GB eMMC, 512MB RAM
	2	4GB eMMC, 1GBRAM
	3	4GB eMMC, 2GBRAM
d		unused
e	xxxx.xxx1	RAM 1,35V, else 1,5 V
	xxxx.xx1x	LVDS, else TTL
	xxxx.x1xx	CPU: CSI_MCLK (P4) -> IIS Master clock, else CAM HSYNC
	xxxx.1xxx	SPI NOR: 2 MB, else 512MB
f	3	DV3
	4	DV4

Looking closer to the file **msc-nR-imx6-133-014-800x480-lcd.dts**:

```
#include "msc-nR-imx6-133-014.dtsi"
#include "msc-nR-imx6-mb2-maxtouch.dtsi"
#include "msc-nR-imx6-800x480-lcd.dtsi"
```

This file is a glue-file. It combines the specific setting for the variant 133 board by including `msc-nR-imx6-133-014.dtsi`  
The include of `msc-nR-imx6-mb2-maxtouch.dtsi` enables the touchscreen.

Taking a look to the head of the file **msc-nR-imx6-133-014.dtsi**:

```
#include "msc-nR-imx6dl.dtsi"
```

and the first lines of **msc-nR-imx6dl.dtsi**:

```
#include "msc-nR-imx6sdl.dtsi"

/ {
    model = "MSC nanoRISC i.MX6DL";
    compatible = "msc,imx6dl_nanoRISC", "fsl,imx6dl";
    memory: memory {
        reg = <0x10000000 0x40000000>; /* 1 GiB */
    };
};
```

and the first lines of **msc-nR-imx6sdl.dtsi**:

```
include "imx6dl.dtsi"
#include "msc-nR-imx6x.dtsi"
```

The file `imx6dl.dtsi` is a generic file which is included in all i.MX6Q designs - also in Freescales reference board `sabresd` - and must not be modified.

And last but not least **msc-nR-imx6x.dtsi**:

```
#include "skeleton.dtsi"
#include "msc-nR-imx6x.dtsi"
```

And last but not least **msc-nR-imx6x.dtsi**:

```
#include <dt-bindings/gpio/gpio.h>
```

## D.3. Creating own variants

The above description is only for educational purposes, since using `msc-ldk` you don't operate in kernel source tree, all changes are usually to be applied as patches or recipes.

To create your own device tree, you can use the prepared example `example_devicetree.dts` provided in the layer `meta-msc-arm-extensions`. This layer is already included in the setup phase of BSP

**NOTE:** The layer is checked out from `msc-git02.msc-ge.com` and therefore it is configured as remote repository. You cannot push any changes here. If you intend to do, you have to connect it to your own repository with:

```
user@devhost:msc-ldk/source/meta-msc-arm-extensions.git $ git remote set-url origin <url-to->
your-repository>
```

Coming back to the example devicetree, which is located in

```
.../meta-msc-arm-extensions.git/recipes-kernel/device-tree/files,
```

it is designed as base to be copied/expanded by your needs. Copying it, you have to add the name (ending with `.dtb`) in

```
.../meta-msc-arm-extensions.git/recipes-kernel/device-tree/linux-yocto-custom_%.bbappend
```

in the variable named **KERNEL\_DEVICETREE**

There are HDMI and LVDS display interfaces as examples listed, but commented out to be inactive. You may activate one or more of them at the same time. Here is the base version of this file:

```

1 /*
2  * Copyright (C) 2017 MSC Technologies, Design Center Aachen
3  *
4  * This program is free software; you can redistribute it and/or modify
5  * it under the terms of the GNU General Public License as published by
6  * the Free Software Foundation; either version 2 of the License, or
7  * (at your option) any later version.
8
9  * This program is distributed in the hope that it will be useful,
10 * but WITHOUT ANY WARRANTY; without even the implied warranty of
11 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE. See the
12 * GNU General Public License for more details.
13
14 * You should have received a copy of the GNU General Public License along
15 * with this program; if not, write to the Free Software Foundation, Inc.,
16 * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
17 */
18 /dts-v1/;
19
20 /*****
21  *
22  * Example Devicetree
23  *
24  * please select the dtsti file according to your board.
25  *
26  *****/
27 /{};
28 /* enable by changing #if 0 to #if 1 */
29 #if 0
30 #include "msc-<modul>-<variant>.dtsti"
31 #endif
32
33 /* example HDMI Display setting */
34 /* enable by changing #if 0 to #if 1 */
35
36 #if 0
37 /* modul := [ msc-nR-imx6 | msc-q7 | msc-sm2-imx6 ] */
38 #include "msc-<modul>-hdmi.dtsi"
39 #endif
40
41 /* example LVDS Display 800x480 */
42 /* enable by changing #if 0 to #if 1 */
43
44 #if 0
45 /* modul := [ msc-nR-imx6 | msc-q7 | msc-sm2-imx6 ] */
46 #include "msc-<modul>-custom-lvds.dtsi"
47 #endif
48
49 /* add your setting below */

```

You can see an example base devicetree include in line 30, this is a template you must change to one valid filename to compile successfully. The possible files are:

```

msc-nR-imx6-112-014.dtsi
msc-nR-imx6-123-014.dtsi
msc-nR-imx6-123-0b4.dtsi
msc-nR-imx6-133-014.dtsi
msc-nR-imx6-143-014.dtsi
msc-nR-imx6-183-014.dtsi

```

depending on the used module.

NOTE: Don't forget to change the #if 0's into #if 1's or remove them in your copy.

The #if 0s are necessary, because the example\_devicetree is always compiled within a kernel build, otherwise there are syntax errors and the build breaks.

In line 46 the LVDS display config file is included. This is also a template you can copy and/or modify to your own display settings.

Additional settings can be added below the last line, which is highlighted in green in the above listing. Some information can be found in chapter D. The general syntax and what kind of entries can be made can be found in internet or other Freescale information sources.

As an example for LVDS display configuration, there is the file `msc-nR-imx6-custom-lvds.dtsi`:

```
1 /*
2  * Copyright 2018 MSC technologies GmbH
3  *
4  * The code contained herein is licensed under the GNU General Public
5  * License. You may obtain a copy of the GNU General Public License
6  * Version 2 or later at the following locations:
7  *
8  * http://www.opensource.org/licenses/gpl-license.html
9  * http://www.gnu.org/copyleft/gpl.html
10 */
11
12 &pwm1 {
13     status = "okay";
14 };
15
16 &lvds1_bl {
17     status = "okay";
18 };
19
20 / {
21     panel-lvds0 {
22         compatible = "panel-lvds-msc";
23         width-mm = <154>;
24         height-mm = <93>;
25         data-mapping = "vesa-24";
26         backlight = <&lvds1_bl>;
27         prepare-ms = <10>;
28         enable-ms = <200>;
29         disable-ms = <200>;
30         unprepare-ms = <550>;
31         enable-gpio = <&gpio3 16 GPIO_ACTIVE_LOW>;
32
33         panel-timing {
34             clock-frequency = <34000000>;
35             hactive = <800>;
36             vactive = <480>;
37             hback-porch = <88>;
38             hfront-porch = <88>;
39             vback-porch = <10>;
40             vfront-porch = <10>;
41             hsync-len = <80>;
42             vsync-len = <25>;
43         };
44
45         port {
46             panel_in_lvds0: endpoint {
47                 remote-endpoint = <&lvds0_out>;
48             };
49         };
50     };
51 };
52
53 &ldb {
54     status = "okay";
55     pinctrl-names = "default";
56     pinctrl-0 = <&pinctrl_lvds1_blen>;
57
58     lvds0: lvds-channel@0 {
59         fsl,data-mapping = "spwg";
60         fsl,data-width = <24>;
61         primary;
62         status = "okay";
63
64         port@4 {
65             reg = <4>;
66             lvds0_out: endpoint {
67                 remote-endpoint = <&panel_in_lvds0>;
68             };
69         };
70     };
71 };
```

NOTE: in line 22 is the special `msc-variant` of `panel-lvds` driver used, which allows an adjustable power sequencing for displays which needs a special timing (line27-30). If the values are omitted no additional delay is made. This is the same behaviour as if `compatible="panel-lvds"` is set. The meanings of the four values are:

- prepare\_ms**: delay between display power enable to video data
- enable\_ms**: delay between video data to valid frame (delay for backlight)
- disable\_ms**: delay for turning display off
- unprepare\_ms**: delay for poweroff completely and powerofftime

The values of the above setting are printed in `dmesg`:

```
panel-lvds-msc panel-lvds0: prepare (POW_ON to video data) set to: 10 ms
panel-lvds-msc panel-lvds0: enable (video data to valid frame) set to: 200 ms
panel-lvds-msc panel-lvds0: disable (time to turn display off) set to: 200 ms
panel-lvds-msc panel-lvds0: unprepare (time to poweroff completely and powerofftime) set to: 550 ms
```

If this lines are missed, probably the compatible entry is not **panel-lvds-msc**.

In the section **panel-timing** you can modify the timings for the needs of your used display. Take a look at the **data-mapping**, here are three possible settings: **jeida-18**, **jeida-24** and **vesa-24**. Keep this settings in sync with the entries **fsl,data-width** and **fsl,data-mapping** in line 55 and 56. Enter here **spwg** for vesa-modes and **jeida** for jeida-modes. You may also update **width-mm** (line 23) and **height-mm** (line 24) to get the right dpi scaling. All other settings may be unchanged.

Additional settings for other devices can be added below the last line, which is highlighted in green in the above listing. Some information can be found in chapter [D](#). The general syntax and what kind of entries can be made can be found in internet or other freescale information sources. This device tree is compiled with the kernel. To rebuild, you can compile the kernel by typing:

```
msc-ldk/build/0584$ ./build.sh bitbake linux-yocto-custom -c deploy
```

Then the device tree will be generated in the deploy directory:

```
./tmp/deploy/images/msc-nr-imx6/zImage-example_devicetree.dtb
```

Then you may copy it to your  $\mu$ SD-Card into the /boot directory on the (second) partition.

The most common way will be to generate a complete new sdcard-image:

```
msc-ldk/build/0584$ ./build.sh bitbake msc-image-base
```

## E. QT5 Graphics Demos

This chapter describes the QT5 demos which are accessible by the following scripts in `/home/root`:

- `QML_BillboardDemo.sh`
- `Qt5_CinematicExperienceDemo.sh`
- `Qt5_EverywhereDemo.sh`
- `Qt5_NMapperDemo.sh`
- `Qt5_NmapcarouseDemo.sh`
- `Qt5_QuitBatteryDemo.sh`
- `Qt5_SmartHomeDemo.sh`

They are described in detail as follows:

### E.1. QML\_BillboardDemo

This Demo shows an animated Text on a LED screen:



**Figure E.1.** – Billboard Demo



## E.2. Qt5\_CinematicExperienceDemo

This Demo shows several cinematic features of QT5. it ist intercative and can be managed by mouse.

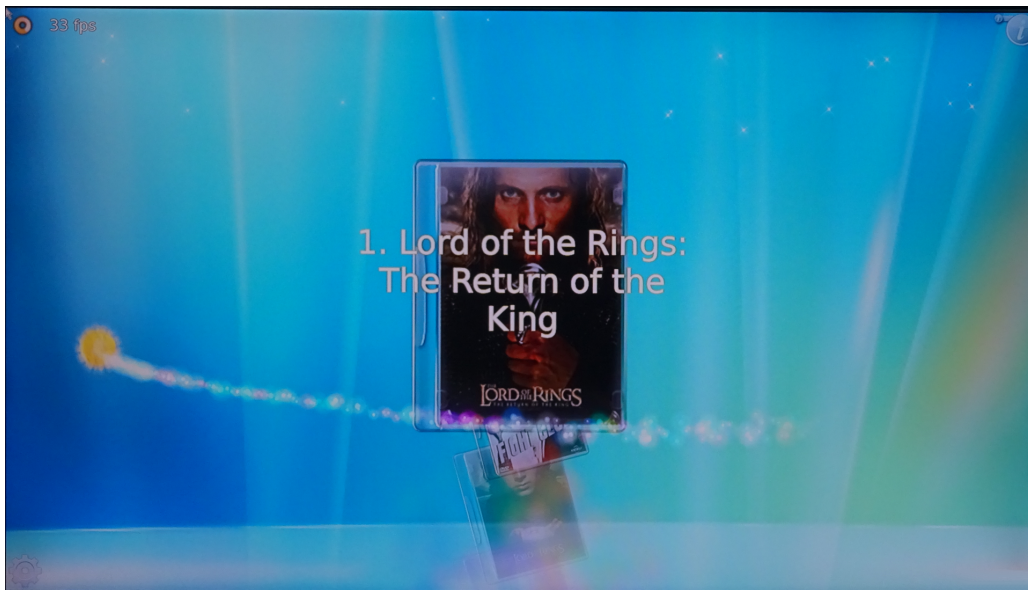


Figure E.2. – Cinematic Experience Demo

## E.3. Qt5\_EverywhereDemo

This Demo shows the versatility of QT and the spectrum of devices it can run.

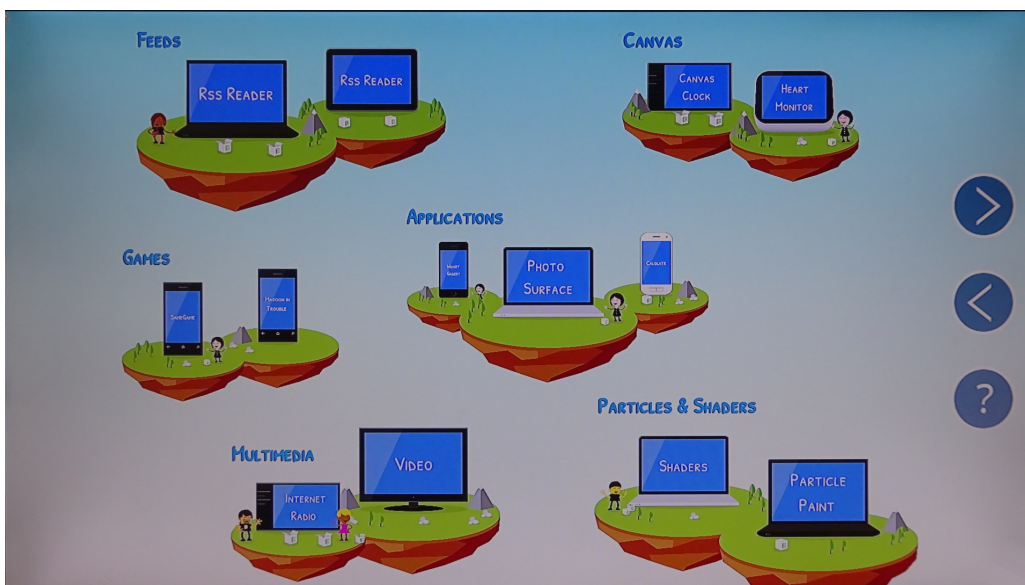


Figure E.3. – Everywhere Demo

## E.4. Qt5\_NMapperDemo

This Demo shows how elements can be manipulated and mapped.

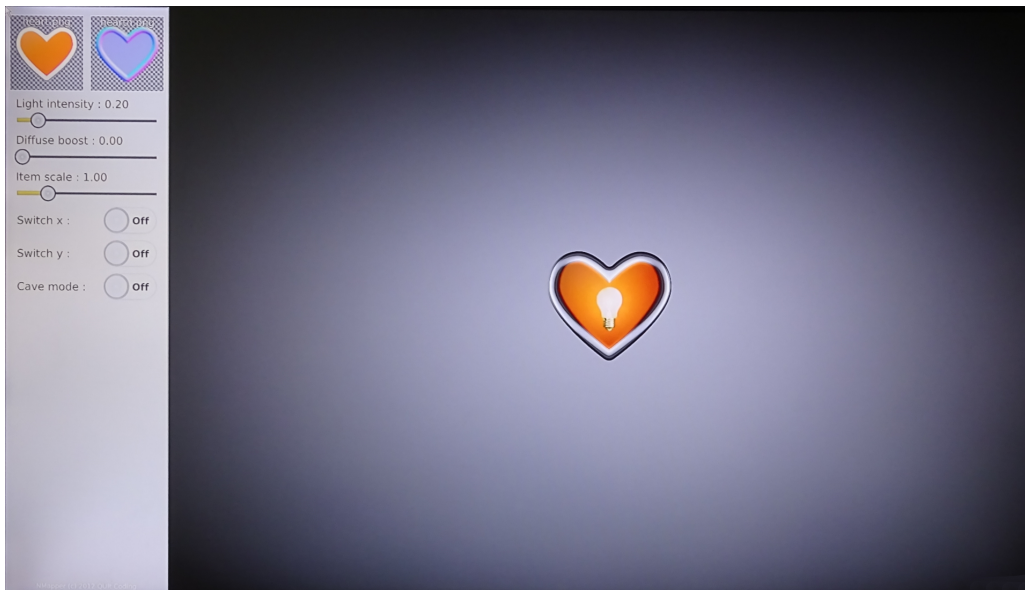


Figure E.4. – NMapper Demo

## E.5. Qt5\_NmapcarouseDemo

This Demo shows an carousel of items, which can be moved by keyboard cursor keys or mouse by clicking the arrows.

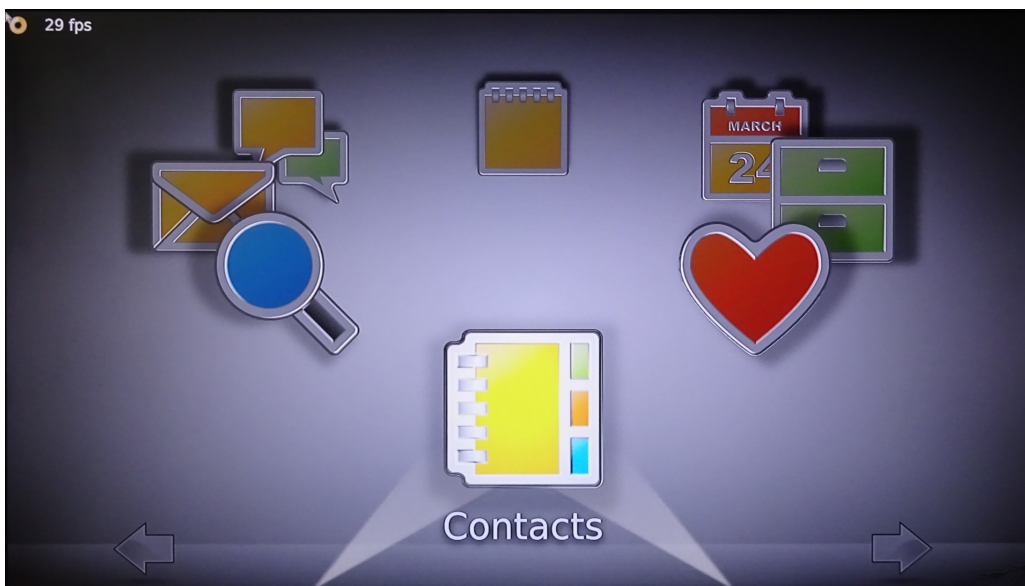


Figure E.5. – Nmapcarouse Demo

# E.6. Qt5\_QuitBatteryDemo

This Demo shows an usage of the QT5 Battery Component.



Figure E.6. – QuitBattery Demo

# E.7. Qt5\_SmartHomeDemo

This Demo shows a Smart Home control screen.



Figure E.7. – SmartHome Demo

## F. Supported Periphery

**Table F.1.** – Supported periphery

Periphery	Status	Comment
Audio	implemented	see <a href="#">Audio</a>
Beeper	implemented	
Ethernet	implemented	
eMMC	implemented	<code>/dev/mmcblk1</code> , see <a href="#">eMMC</a>
GPIOs	implemented	see <a href="#">GPIOs</a>
Graphic output	implemented	see <a href="#">Display Support</a>
Video encoding/decoding (H.264)	implemented	
Hardware sensors	implemented	
I2C	implemented	<a href="#">I2C</a>
PCIe	implemented	
RTC	implemented	
SATA	implemented	(only on i.MX6DL and i.MX6Q)
SDIO	implemented	<code>/dev/mmcblk0</code>
SPI	implemented	<a href="#">SPI</a>
TPM	t.b.d	
UART	implemented	<a href="#">UART</a>
USB-2.0	implemented	
USB-2.0 device	implemented	<a href="#">USB Device Support</a>

## F.1. eMMC

eMMC is available as MMC device `/dev/mmcblk2`

## F.2. GPIOs

GPIOs are available. Add the `gpio` base to the hardware GPIO port number. E.g. for `GPIO_2.15` add 32 to 15 to get the GPIO number 47. Then access this GPIO via `/sys/class/gpio`.

**Table F.2.** – GPIOs

gpio count	gpio base	gpio name
32	0	GPIO_1.x
32	32	GPIO_2.x
32	64	GPIO_3.x
32	96	GPIO_4.x
32	128	GPIO_5.x
32	160	GPIO_6.x
32	192	GPIO_7.x

Usable GPIOs are listed in `UM100584_MSC-NR-IMX6`, chapter 2.11 Signals and Pin Out. The GPIO-Number `n` for each GPIO is described in the column named Alternate PIN FUNCTION as `GPIO[n]`.

## F.3. Audio

Audio is supported by ALSA. Sound controls can be modified with `alsamixer`, which is very complex, there are above 100 controls for the used Audio Codec `TLV320AIC3107`. In an usual use case only a few will be used. There is a command line tool to set each setting named `amixer`. The available Controls can be listed with: `amixer controls`. The following chapters explain, how to set the mixer according to each action.

Note that all examples below use the predefined setting stored in the file `/var/lib/alsa/asound.state`.

### F.3.1. Playback on Headphone

To play sound on the headphone jack, do:

```
# restore defaults
alsactl restore
amixer sset 'PCM' 100%
amixer sset 'HP DAC' 100%
# test both channels alternating
speaker-test -c 2 -f 800 -t sine
# or play a file (the alsa sounds are all mono)
aplay /usr/share/sounds/alsa/Front_Center.wav
```

### F.3.2. Playback on Speaker

To play mono sound on the Speaker, do:

```
# restore defaults
alsactl restore
amixer sset 'PCM' 100%
amixer sset 'Line DAC' 100%
# test both channels alternating
speaker-test -c 2 -f 800 -t sine
# or play a file (the alsa sounds are all mono)
aplay /usr/share/sounds/alsa/Front_Center.wav
```

### F.3.3. Record from Mic

To record from the microphone, do:

```
# restore defaults
alsactl restore
# Set PGA gain
amixer sset 'PGA' 75%
# for check with headphone:
amixer sset 'HP PGA Bypass' 80%
# record 10 sec
arecord -c 2 -f S16_LE -r 22050 -d 10 record.wav
# Play back with:
aplay record.wav
```

### F.3.4. Record from LineIn

To record from LineIn, do:

```
# restore defaults
alsactl restore
# set Line2 routing
amixer sset 'Left PGA Mixer Line2L' on
amixer sset 'Right PGA Mixer Line2R' on
# Set PGA gain
amixer sset 'PGA' 75%
# for check with headphone:
amixer sset 'Left HP Mixer PGAL Bypass' on
amixer sset 'Right HP Mixer PGAR Bypass' on
amixer sset 'HP PGA Bypass' 80%
# record 10 sec
arecord -c 2 -f S16_LE -r 22050 -d 10 record.wav
# Play back with:
aplay record.wav
```

### F.3.5. 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
}
```

### F.3.6. Persistent Mixer Settings

The mixer settings can be saved to `/var/lib/alsa/asound.state` with `alsactl store` and restored with `alsactl restore`. Note: The standard settings are used as a base settings in chapter F.3. Depending on your changes being stored, the example setting may not work anymore.

## F.4. I2C

There are several I2C busses available as named `i2c-<n>`, configured to 100kHz (standard speed). The count of bus numbers can be different depending on the board, therefore run `i2cdetect -l` to list all busses and determine the I2C bus number. The devices on the bus `<n>` can be showed with `i2cdetect -y <n>`

## F.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).

### F.5.1. Testing Connection

To load the driver, do:

```
modprobe g_zero
```

A new USB device `Gadget Zero, version: Cinco de Mayo 2008` will appear.

To disable the device, run:

```
rmmod g_zero
```

### F.5.2. Setting Up Serial Gadget

To load the driver, do:

```
modprobe g_serial
```

A new USB device `Gadget Serial v2.4` 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
```

### F.5.3. Setting Up Ethernet Gadget

To load the driver, do:

```
modprobe g_ether
```

A new USB device `Ethernet Gadget, version: Memorial Day 2008` 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
```

## F.5.4. 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 Mass Storage Gadget, version: 2009/09/11 will appear.

It can be mounted and the contents modified.

To disable the device, run:

```
rmmod g_mass_storage
```

## F.6. UART

The kernel reserves 3 serial port devices (`/dev/ttymx0 ... /dev/ttymx2`) even when they are not connected.

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

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

A login shell is created on `/dev/ttymx0` running with 115200 baud, 8 databits, no parity.

## F.7. SPI

SPI is available as `/dev/spi0.0` for eeprom, and limited to 50 MHz and configured for SPI Mode 0.

## F.8. Graphic Output

### F.8.1. Display Support

The BSP supports output on multiple displays. There are devicetrees for several configurations: HDMI (named `*-hdmi.dtb`), RGB (named `*-800x480-lcd`), and LVDS (named `*-800x480-lvds.dtb`, `*-1024x768-lvds.dtb`) or `*-1280x1024-lvds.dtb`)

Up to 3 Displays (1x HDMI, 2x LVDS) are possible to connect, depending on hardware and used CPU.

LVDS display backlight may be controlled by software:

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

```
echo 3 >/sys/devices/soc0/lcd_backlight/backlight/lcd_backlight/brightness
```



# G. Troubleshooting

**Table G.1.** – Errors

Symptom	Solution
gitolite@msc-git02.msc-ge.com asks for password	Register as described here:  <a href="#">Registration On The MSC Git Server</a>
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 <code>hosts</code> line in <code>/etc/nsswitch.conf</code> on the target to this: <code>hosts: files</code>

# 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$ ./setup.py --bsp=0584 -layers-lxqt -layers-browser-chrome-instead-of-otter
```



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

## 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 --systohctc
```

# 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/0584/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>

**Canary 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](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 additional switch `--layers-security`. For example:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --layers-lxqt --layers-security
user@devhost:msc-ldk$ cd build/0584-security
user@devhost:msc-ldk/build/0584-security$ make msc-image-lxqt
```

# J. 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 J.1.** – CPU core states

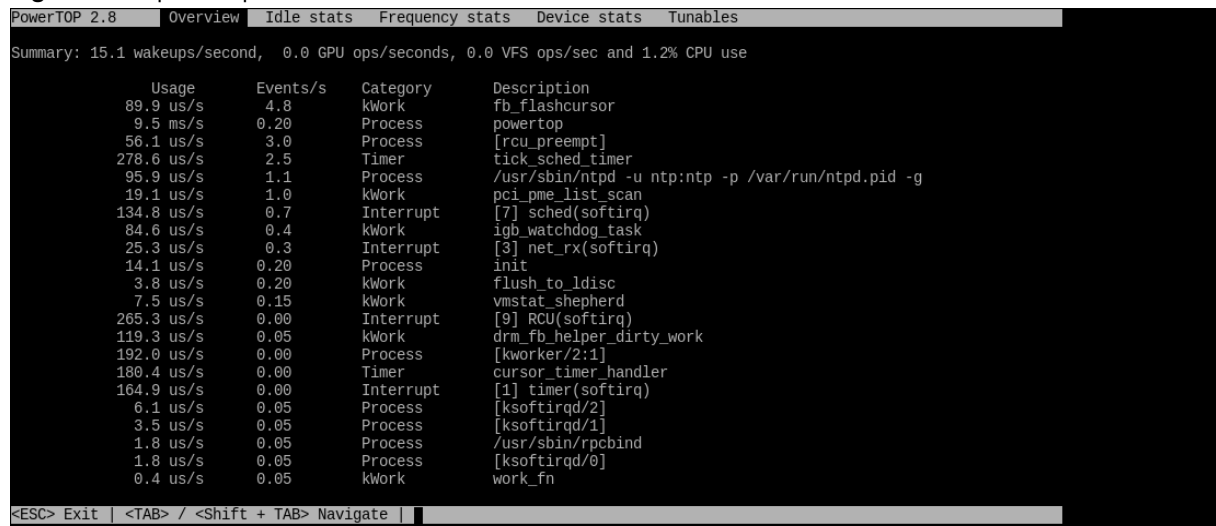
Mode	Description
C0	CPU fully turned on
C1	The main CPU clocks used for executing instructions are stopped, all others are still on
C3	All internal CPU clocks are stopped.
C6	All internal CPU clocks are stopped and internal voltage is reduced.

Depending on the CPU more modes might be available.

## J.1. 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 J.1.** – powertop overview



## K. Links

- MSC Technologies <http://www.msc-technologies.eu>
- Yocto project <https://www.yoctoproject.org>
- Yocto project documentation  
<http://www.yoctoproject.org/docs/3.0.1/mega-manual/mega-manual.html>
- LXQt <http://lxqt.org>
- Otter Browser <https://otter-browser.org>

# L. License Overview

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

Licenses		
(GPLv2+  LGPLv3)	AFL-2.1	Artistic-1.0
BSD	BSD-2-Clause	BSD-3-Clause
BSD-4-Clause	BSL-1.0	Firmware-atheros_firmware
Firmware-imx-sdma_firmware	GFDL-1.2	GPL-2+
GPL-2.0	GPL-2.0+	GPL-3.0-with-GCC-exception
GPLv1	GPLv2	GPLv2+
GPLv2.0+	GPLv3	GPLv3+
ICU	ISC	LGPL-2.1
LGPL-2.1+	LGPLv2	LGPLv2.1
LGPLv2.1+	LGPLv3	LGPLv3+
MIT	MIT-style	MPL-2.0
NTP	PD	PSFv2
Proprietary	Python-2.0	Zlib
bzip2	openssl	zsh

**Table L.2.** – License overview listing

Recipe	Version	Licenses
<b>alsa-lib</b> <i>alsa-conf</i>	1.1.9 <i>alsa-lib</i>	GPLv2.1, GPLv2+
<b>alsa-state</b> <i>alsa-state</i>	0.2.0 <i>alsa-states</i>	MIT
<b>alsa-utils</b> <i>alsa-utils</i> <i>alsa-utils-alsactl</i> <i>alsa-utils-alsamixer</i> <i>alsa-utils-alsaucm</i> <i>alsa-utils-aplay</i> <i>alsa-utils-aseqnet</i> <i>alsa-utils-midi</i>	1.1.9 <i>alsa-utils-aconnect</i> <i>alsa-utils-alsaloop</i> <i>alsa-utils-alsatplg</i> <i>alsa-utils-amixer</i> <i>alsa-utils-aseqdump</i> <i>alsa-utils-iecset</i> <i>alsa-utils-speakertest</i>	GPLv2+
<b>attr</b> <i>libattr</i>	2.4.47	GPLv2.1+
<b>avahi</b> <i>avahi-daemon</i> <i>libavahi-common</i>	0.7 <i>avahi-locale-en-gb</i> <i>libavahi-core</i>	GPLv2+, LGPLv2.1+
<b>base-files</b> <i>base-files</i>	3.0.14	GPLv2
<b>base-passwd</b> <i>base-passwd</i>	3.5.29	GPLv2+
<b>bash</b> <i>bash</i>	5.0	GPLv3+
<b>bc</b> <i>bc</i>	1.07.1	GPLv3+
<b>binutils</b> <i>binutils</i>	2.32.0 <i>libbfd</i>	GPLv3
<b>bluez5</b> <i>bluez5</i> <i>bluez5-obex</i>	5.50 <i>bluez5-noinst-tools</i>	GPLv2+, LGPLv2.1+
<b>bonnie++</b> <i>bonnie++</i>	1.03e	GPLv2
<b>boost</b>  <i>boost-filesystem</i> <i>boost-regex</i>	1.71.0  <i>boost-iostreams</i>	BSL-1.0, MIT, Python-2.0
<b>bootlog</b> <i>bootlog</i>	1.0	MIT
<b>busybox</b> <i>busybox</i> <i>busybox-syslog</i>	1.31.0 <i>busybox-hwclock</i> <i>busybox-udhcpc</i>	GPLv2, bzip2
<b>bzip2</b> <i>libbz2</i>	1.0.8	bzip2
<b>ca-certificates</b> <i>ca-certificates</i>	20190110	GPL-2.0+, MPL-2.0
<b>cifs-utils</b> <i>cifs-utils</i>	6.4	GPLv3, LGPLv3
<b>cmdline-keyboard</b> <i>cmdline-keyboard</i>	1.0	MIT
<b>cpufrequtils</b> <i>cpufrequtils</i>	008	GPLv2
<b>dbus</b> <i>dbus</i>	1.12.16 <i>dbus-lib</i>	AFL-2.1, GPLv2+
<b>devmem2</b> <i>devmem2</i>	1.0	GPLv2+



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

Recipe	Version	Licenses
<b>dialog</b> <i>dialog</i>	1.3-20190728	LGPL-2.1
<b>dosfstools</b> <i>dosfstools</i>	4.1	GPLv3
<b>e2fsprogs</b> <i>e2fsprogs-badblocks</i> <i>e2fsprogs-mke2fs</i> <i>libcomerr</i> <i>libext2fs</i>	1.45.3 <i>e2fsprogs-e2fsck</i> <i>e2fsprogs-tune2fs</i> <i>libe2p</i>	GPLv2
<b>elfutils</b> <i>elfutils</i> <i>libelf</i>	0.177 <i>libdw</i>	GPLv3+
<b>ell</b> <i>ell</i>	0.22	LGPLv2.1
<b>ethtool</b> <i>ethtool</i>	5.2	GPLv2+
<b>eudev</b> <i>eudev</i> <i>libudev</i>	3.2.8 <i>eudev-hwdb</i>	GPLv2.0+, LGPL-2.1+
<b>evtest</b> <i>evtest</i>	1.31	GPLv2
<b>expat</b> <i>expat</i>	2.2.8	MIT
<b>firmware-imx</b> <i>firmware-imx-vpu-imx6d</i>	8.1.1 <i>firmware-imx-vpu-imx6q</i>	Proprietary
<b>flac</b>  <i>libflac</i>	1.3.3	GFDL-1.2, GPLv2+, LGPLv2.1+, BSD
<b>gcc-runtime</b> <i>libstdc++</i>	9.2.0	GPL-3.0-with-GCC-exception
<b>gdb</b>  <i>gdb</i>	8.3.1	GPLv2, GPLv3, LGPLv2, LGPLv3
<b>gdbm</b> <i>gdbm</i>	1.18.1 <i>gdbm-compat</i>	GPLv3
<b>glib-2.0</b> <i>glib-2.0</i>	2.60.7 <i>glib-2.0-locale-en-gb</i>	LGPLv2.1+, BSD, PD
<b>glibc</b> <i>glibc</i>	2.30 <i>glibc-thread-db</i>	GPLv2, LGPLv2.1
<b>glibc-locale</b> <i>glibc-locale-en-gb</i> <i>locale-base-en-gb</i>	2.30 <i>locale-base-de-de</i> <i>locale-base-en-us</i>	GPLv2, LGPLv2.1
<b>gmp</b> <i>gmp</i>	6.1.2	GPLv2+, LGPLv3+
<b>gnutls</b> <i>gnutls</i>	3.6.8	LGPLv2.1+
<b>grep</b> <i>grep</i>	3.3	GPLv3
<b>hdparm</b> <i>hdparm</i>	9.58	BSD
<b>hostapd</b> <i>hostapd</i>	2.9	BSD-3-Clause

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

Recipe	Version	Licenses
<b>htpdate</b> <i>htpdate</i>	1.1.3	GPLv2
<b>i2c-ids</b> <i>i2c-ids</i>	1.0	MIT
<b>i2c-tools</b> <i>i2c-tools</i>	4.1	GPLv2+
<b>i2c-write-read</b> <i>i2c-write-read</i>	git	GPLv2
<b>icu</b> <i>libicudata</i> <i>libicuuc</i>	64.2 <i>libicut18n</i>	ICU
<b>init-ifupdown</b> <i>init-ifupdown</i>	1.0	GPLv2
<b>initscripts</b> <i>initscripts</i>	1.0 <i>initscripts-functions</i>	GPLv2
<b>iperf3</b> <i>iperf3</i>	3.7	BSD
<b>iw</b> <i>iw</i>	5.3	BSD-2-Clause
<b>kbd</b> <i>kbd-keymaps</i>	2.0.4	GPLv2+
<b>kexec-tools</b> <i>kexec</i>	2.0.19	GPLv2
<b>kmod</b> <i>kmod</i>	26 <i>libkmod</i>	GPL-2.0+, LGPL-2.1+
<b>libcap</b> <i>libcap</i>	2.27	BSD, GPLv2
<b>libdaemon</b> <i>libdaemon</i>	0.14	LGPLv2.1+
<b>libffi</b> <i>libffi</i>	3.3 rc0	MIT
<b>libgcc</b> <i>libgcc</i>	9.2.0	GPL-3.0-with-GCC-exception
<b>libgcrypt</b> <i>libgcrypt</i>	1.8.4	LGPLv2.1+
<b>libgpg-error</b> <i>libgpg-error</i>	1.36	GPLv2+, LGPLv2.1+
<b>libical</b> <i>libical</i>	3.0.6	LGPLv2.1, MPL-2.0
<b>libidn2</b> <i>libidn2</i>	2.2.0	(GPLv2+   LGPLv3)
<b>libjitterentropy</b> <i>libjitterentropy</i>	2.1.2	GPLv2+, BSD
<b>libmscboost</b> <i>libmscboost</i>	git	LGPLv2.1
<b>libmscboostpython</b> <i>libmscboostpython</i>	git	LGPLv2.1
<b>libnl</b> <i>libnl</i>	3.5.0 <i>libnl-genl</i>	LGPLv2.1
<b>libnsl2</b> <i>libnsl2</i>	1.2.0	LGPL-2.1
<b>libnss-mdns</b> <i>libnss-mdns</i>	0.10	LGPLv2.1+

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

Recipe	Version	Licenses
<b>libogg</b> <i>libogg</i>	1.3.4	BSD
<b>libpcap</b> <i>libpcap</i>	1.9.1	BSD
<b>libpcre</b> <i>libpcre</i>	8.43	BSD
<b>libsamplerate0</b> <i>libsamplerate0</i>	0.1.9	BSD-2-Clause
<b>libsndfile1</b> <i>libsndfile1</i>	1.0.28	LGPLv2.1
<b>libtirpc</b> <i>libtirpc</i>	1.1.4	BSD
<b>libunistring</b> <i>libunistring</i>	0.9.10	LGPLv3+, GPLv2
<b>libusb1</b> <i>libusb1</i>	1.0.22	LGPLv2.1+
<b>libvorbis</b> <i>libvorbis</i>	1.3.6	BSD
<b>libx11</b> <i>libx11</i>	1.6.8	MIT, MIT-style, BSD
<b>libxau</b> <i>libxau</i>	1.0.9	MIT-style
<b>libxcb</b> <i>libxcb</i>	1.13.1	MIT
<b>libxcrypt</b> <i>libxcrypt</i>	4.4.8	LGPLv2.1
<b>libxdrmcp</b> <i>libxdrmcp</i>	1.1.3	MIT-style
<b>libxml2</b> <i>libxml2</i>	2.9.9 <i>libxml2-utils</i>	MIT
<b>linux-firmware</b> <i>linux-firmware-ath6k</i> <i>linux-firmware-imx-sdma-imx6q</i> <i>linux-firmware-imx-sdma-license</i>	20190815	Firmware-atheros_firmware <i>linux-firmware-atheros-license</i> <i>linux-firmware-imx-sdma-imx7d</i>
<b>linux-tools-testusb</b> <i>linux-tools-testusb</i>	1.0	GPL-2.0
<b>linux-yocto-custom</b> <i>kernel-base</i> <i>kernel-image</i> <i>kernel-module-asix-5.4.8-yocto-standard</i> <i>kernel-module-at25-5.4.8-yocto-standard</i> <i>kernel-module-binfmt-misc-5.4.8-yocto-standard</i> <i>kernel-module-brcmutil-5.4.8-yocto-standard</i> <i>kernel-module-cdc-eem-5.4.8-yocto-standard</i> <i>kernel-module-cdc-ncm-5.4.8-yocto-standard</i> <i>kernel-module-coda-vpu-5.4.8-yocto-standard</i> <i>kernel-module-crc-ccitt-5.4.8-yocto-standard</i> <i>kernel-module-crc7-5.4.8-yocto-standard</i> <i>kernel-module-dw-hdmi-ahb-audio-5.4.8-yocto-standard</i> <i>kernel-module-ehset-5.4.8-yocto-standard</i> <i>kernel-module-ftdi-sio-5.4.8-yocto-standard</i> <i>kernel-module-g-audio-5.4.8-yocto-standard</i> <i>kernel-module-g-ether-5.4.8-yocto-standard</i> <i>kernel-module-g-hid-5.4.8-yocto-standard</i> <i>kernel-module-g-multi-5.4.8-yocto-standard</i> <i>kernel-module-g-serial-5.4.8-yocto-standard</i> <i>kernel-module-gadgetfs-5.4.8-yocto-standard</i> <i>kernel-module-i2c-algo-pca-5.4.8-yocto-standard</i>	5.4.8	GPLv2 <i>kernel-devicetree</i> <i>kernel-image-zimage</i> <i>kernel-module-at24-5.4.8-yocto-standard</i> <i>kernel-module-ax88179-178a-5.4.8-yocto-standard</i> <i>kernel-module-brcmfmac-5.4.8-yocto-standard</i> <i>kernel-module-cdc-acm-5.4.8-yocto-standard</i> <i>kernel-module-cdc-ether-5.4.8-yocto-standard</i> <i>kernel-module-cdc-subset-5.4.8-yocto-standard</i> <i>kernel-module-configfs-5.4.8-yocto-standard</i> <i>kernel-module-crc-itu-t-5.4.8-yocto-standard</i> <i>kernel-module-dummy-hcd-5.4.8-yocto-standard</i> <i>kernel-module-echainiv-5.4.8-yocto-standard</i> <i>kernel-module-evbug-5.4.8-yocto-standard</i> <i>kernel-module-g-acm-ms-5.4.8-yocto-standard</i> <i>kernel-module-g-cdc-5.4.8-yocto-standard</i> <i>kernel-module-g-ffs-5.4.8-yocto-standard</i> <i>kernel-module-g-mass-storage-5.4.8-yocto-standard</i> <i>kernel-module-g-ncm-5.4.8-yocto-standard</i> <i>kernel-module-g-zero-5.4.8-yocto-standard</i> <i>kernel-module-gspca-main-5.4.8-yocto-standard</i> <i>kernel-module-i2c-algo-pcf-5.4.8-yocto-standard</i>

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

Recipe	Version	Licenses
<i>kernel-module-imx-vdoa-5.4.8-yocto-standard</i>	<i>kernel-module-isofs-5.4.8-yocto-standard</i>	
<i>kernel-module-libcomposite-5.4.8-yocto-standard</i>	<i>kernel-module-libcrc32c-5.4.8-yocto-standard</i>	
<i>kernel-module-msdos-5.4.8-yocto-standard</i>	<i>kernel-module-net1080-5.4.8-yocto-standard</i>	
<i>kernel-module-net2280-5.4.8-yocto-standard</i>	<i>kernel-module-nls-iso8859-15-5.4.8-yocto-standard</i>	
<i>kernel-module-option-5.4.8-yocto-standard</i>	<i>kernel-module-ov5640-5.4.8-yocto-standard</i>	
<i>kernel-module-pegasus-5.4.8-yocto-standard</i>	<i>kernel-module-psmouse-5.4.8-yocto-standard</i>	
<i>kernel-module-r8152-5.4.8-yocto-standard</i>	<i>kernel-module-rtl8150-5.4.8-yocto-standard</i>	
<i>kernel-module-serport-5.4.8-yocto-standard</i>	<i>kernel-module-snd-hwdep-5.4.8-yocto-standard</i>	
<i>kernel-module-snd-rawmidi-5.4.8-yocto-standard</i>	<i>kernel-module-snd-usb-audio-5.4.8-yocto-standard</i>	
<i>kernel-module-snd-usbmidi-lib-5.4.8-yocto-standard</i>	<i>kernel-module-spidev-5.4.8-yocto-standard</i>	
<i>kernel-module-u-audio-5.4.8-yocto-standard</i>	<i>kernel-module-u-ether-5.4.8-yocto-standard</i>	
<i>kernel-module-u-serial-5.4.8-yocto-standard</i>	<i>kernel-module-udf-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-acm-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-ecm-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-ecm-subset-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-eem-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-fs-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-hid-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-mass-storage-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-midi-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-ncm-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-obex-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-printer-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-rndis-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-serial-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-ss-lb-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-uac1-5.4.8-yocto-standard</i>	<i>kernel-module-usb-f-uac2-5.4.8-yocto-standard</i>	
<i>kernel-module-usb-f-uvic-5.4.8-yocto-standard</i>	<i>kernel-module-usb-wwan-5.4.8-yocto-standard</i>	
<i>kernel-module-usbnet-5.4.8-yocto-standard</i>	<i>kernel-module-usbserial-5.4.8-yocto-standard</i>	
<i>kernel-module-usbtest-5.4.8-yocto-standard</i>	<i>kernel-module-uvcvideo-5.4.8-yocto-standard</i>	
<i>kernel-module-videobuf2-vmalloc-5.4.8-yocto-standard</i>	<i>kernel-module-wl12xx-5.4.8-yocto-standard</i>	
<i>kernel-module-wlcore-5.4.8-yocto-standard</i>	<i>kernel-module-wlcore-sdio-5.4.8-yocto-standard</i>	
<i>kernel-module-zaurus-5.4.8-yocto-standard</i>	<i>kernel-modules</i>	
<b>lrzsz</b>	0.12.20	GPLv2+
<i>lrzsz</i>		
<b>ltrace</b>	7.91	GPLv2
<i>ltrace</i>		
<b>maxtouch</b>	1.0	MIT
<i>maxtouch</i>		
<b>mem-edit</b>	git	GPLv2
<i>mem-edit</i>		
<b>memtester</b>	4.3.0	GPLv2
<i>memtester</i>		
<b>minicom</b>	2.7.1	GPLv2+
<i>minicom</i>		
<b>mmc-utils</b>	0.1	GPLv2
<i>mmc-utils</i>		
<b>mobile-broadband-provider-info</b>	20190618	PD
<i>mobile-broadband-provider-info</i>		
<b>modutils-initscripts</b>	1.0	PD
<i>modutils-initscripts</i>		
<b>msc-bug-report</b>	1.0	GPLv2
<i>msc-bug-report</i>		
<b>msc-completion</b>	1.0	MIT
<i>msc-completion</i>		
<b>msc-cpufreq</b>	1.0	GPLv2
<i>msc-cpufreq</i>		
<b>msc-init-script-early</b>	1.0	MIT
<i>msc-init-script-early</i>		
<b>msc-init-script-late</b>	1.0	MIT
<i>msc-init-script-late</i>		
<b>msc-ldk-benchmark</b>	git	GPLv2
<i>msc-ldk-benchmark</i>		
<b>msc-ldk-verification</b>	git	GPLv2
<i>msc-ldk-verification</i>		

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

Recipe	Version	Licenses
<b>msc-ldk-verification-apps</b> <i>msc-ldk-verification-apps</i>	git	GPLv2
<b>msc-linux-scripts</b> <i>msc-linux-scripts</i>	git	GPLv2
<b>mscio-cmd</b> <i>mscio-cmd</i>	git	LGPLv2.1
<b>mscio-drivers</b> <i>kernel-module-devreg-5.4.8-yocto-standard</i> <i>kernel-module-eapi-ec-bl-5.4.8-yocto-standard</i> <i>kernel-module-eapi-ec-running-time-5.4.8-yocto-standard</i> <i>kernel-module-i2c-ids-of-5.4.8-yocto-standard</i> <i>mscio-drivers</i>	git	GPLv2 <i>kernel-module-eapi-ec-5.4.8-yocto-standard</i> <i>kernel-module-eapi-ec-hwm-5.4.8-yocto-standard</i> <i>kernel-module-eapi-ec-wdt-5.4.8-yocto-standard</i> <i>kernel-module-user-gpios-5.4.8-yocto-standard</i>
<b>mscio-lib</b> <i>mscio-lib</i>	git	LGPLv2.1
<b>mtd-utils</b> <i>mtd-utils</i>	2.1.1	GPLv2+
<b>nano</b> <i>nano</i>	2.2.5	GPLv2
<b>ncurses</b> <i>ncurses-libformw</i> <i>ncurses-libncurses</i> <i>ncurses-libpanelw</i> <i>ncurses-terminfo</i>	6.1	MIT <i>ncurses-libmenuw</i> <i>ncurses-libncursesw</i> <i>ncurses-libtinfo</i> <i>ncurses-terminfo-base</i>
<b>neard</b> <i>neard</i>	0.16	GPLv2
<b>netbase</b> <i>netbase</i>	5.6	GPLv2
<b>netpipe</b> <i>netpipe</i>	3.7.2	GPLv1
<b>nettle</b> <i>nettle</i>	3.5.1	LGPLv3+, GPLv2+
<b>nfs-utils</b> <i>nfs-utils-client</i>	2.4.1 <i>nfs-utils-mount</i>	MIT, GPLv2+, BSD
<b>ntp</b> <i>ntp</i> <i>ntpdate</i>	4.2.8p13 <i>ntp-tickadj</i>	NTP
<b>ofono</b> <i>ofono</i>	1.30	GPLv2
<b>openssh</b> <i>openssh</i> <i>openssh-scp</i> <i>openssh-sshd</i>	8.0p1 <i>openssh-keygen</i> <i>openssh-ssh</i>	BSD, ISC, MIT
<b>openssl</b> <i>libcrypto</i> <i>openssl</i> <i>openssl-conf</i>	1.1.1d <i>libssl</i> <i>openssl-bin</i>	openssl
<b>opkg-utils</b> <i>update-alternatives-opkg</i>	0.4.1	GPLv2+
<b>packagegroup-base</b> <i>packagegroup-base</i> <i>packagegroup-base-alsa</i> <i>packagegroup-base-extended</i> <i>packagegroup-base-nfc</i> <i>packagegroup-base-pci</i> <i>packagegroup-base-usbhost</i> <i>packagegroup-base-zeroconf</i> <i>packagegroup-machine-base</i>	1.0 <i>packagegroup-base-3g</i> <i>packagegroup-base-bluetooth</i> <i>packagegroup-base-ipv6</i> <i>packagegroup-base-nfs</i> <i>packagegroup-base-usb gadget</i> <i>packagegroup-base-wifi</i> <i>packagegroup-distro-base</i>	MIT
<b>packagegroup-core-boot</b>	1.0	MIT

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

Recipe	Version	Licenses
<i>packagegroup-core-boot</i>		
<b>packagegroup-msc-ldk-core</b> <i>packagegroup-msc-ldk-core</i>	1.0	MIT
<b>packagegroup-mscio</b> <i>packagegroup-mscio</i>	1.0	MIT
<b>parted</b> <i>parted</i>	3.2	GPLv3+
<b>pci2uio</b> <i>kernel-module-pci2uio-5.4.8-yocto-standard</i>	git <i>pci2uio</i>	GPLv2
<b>pciutils</b> <i>libpci</i> <i>pciutils-ids</i>	3.6.2 <i>pciutils</i>	GPLv2+
<b>pm-utils</b> <i>pm-utils</i>	1.4.1	GPLv2
<b>powertop</b> <i>powertop</i> <i>powertop-locale-en-us</i>	2.10 <i>powertop-locale-en-gb</i>	GPLv2
<b>python3</b> <i>libpython3</i> <i>python3-asyncio</i> <i>python3-codecs</i> <i>python3-compression</i> <i>python3-crypt</i> <i>python3-curses</i> <i>python3-db</i> <i>python3-difflib</i> <i>python3-distutils-windows</i> <i>python3-email</i> <i>python3-html</i> <i>python3-image</i> <i>python3-json</i> <i>python3-mailbox</i> <i>python3-mime</i> <i>python3-mmap</i> <i>python3-multiprocessing</i> <i>python3-netserver</i> <i>python3-pickle</i> <i>python3-plistlib</i> <i>python3-profile</i> <i>python3-resource</i> <i>python3-smtpd</i> <i>python3-stringold</i> <i>python3-terminal</i> <i>python3-tkinter</i> <i>python3-unittest</i> <i>python3-venv</i> <i>python3-xmlrpc</i>	3.7.5 <i>python3-2to3</i> <i>python3-audio</i> <i>python3-compile</i> <i>python3-core</i> <i>python3-ctypes</i> <i>python3-datetime</i> <i>python3-debugger</i> <i>python3-distutils</i> <i>python3-doctest</i> <i>python3-fcntl</i> <i>python3-idle</i> <i>python3-io</i> <i>python3-logging</i> <i>python3-math</i> <i>python3-misc</i> <i>python3-modules</i> <i>python3-netclient</i> <i>python3-numbers</i> <i>python3-pkgutil</i> <i>python3-pprint</i> <i>python3-pydoc</i> <i>python3-shell</i> <i>python3-sqlite3</i> <i>python3-syslog</i> <i>python3-threading</i> <i>python3-typing</i> <i>python3-unixadmin</i> <i>python3-xml</i>	PSFv2
<b>readline</b> <i>readline</i>	8.0	GPLv3+
<b>rng-tools</b> <i>rng-tools</i>	6.7	GPLv2
<b>rpcbind</b> <i>rpcbind</i>	1.2.5	BSD
<b>run-postinsts</b> <i>run-postinsts</i>	1.0	MIT
<b>shadow</b> <i>shadow</i>	4.6 <i>shadow-base</i>	BSD, Artistic-1.0
<b>shadow-securetty</b> <i>shadow-securetty</i>	4.6	MIT

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

Recipe	Version	Licenses
<b>shared-mime-info</b> <i>shared-mime-info</i>	1.10	GPLv2
<b>smartmontools</b> <i>smartmontools-ctl</i>	6.3	GPLv2
<b>spi-register</b> <i>spi-register</i>	git	GPLv2
<b>sqlite3</b> <i>libsqlite3</i>	3.29.0	PD
<b>strace</b> <i>strace</i>	5.3	LGPL-2.1+, GPL-2+
<b>sudo</b> <i>sudo</i>	1.8.27	ISC, BSD, Zlib
<b>sysfsutils</b> <i>libsysfs</i>	2.1.0	LGPLv2.1
<b>system-test-controller-config</b> <i>system-test-controller-config</i>	1.0	MIT
<b>system-test-controller-v2</b> <i>system-test-controller-v2</i>	git	GPLv2
<b>sysvinit</b> <i>sysvinit</i>	2.88dsf <i>sysvinit-pidof</i>	GPLv2+
<b>sysvinit-inittab</b> <i>sysvinit-inittab</i>	2.88dsf	GPLv2
<b>tcp-wrappers</b> <i>libwrap</i>	7.6	BSD
<b>tcpdump</b> <i>tcpdump</i>	4.9.3	BSD
<b>tiny-shell</b> <i>tiny-shell</i>	git	GPLv2
<b>tzdata</b>  <i>tzdata</i> <i>tzdata-americas</i> <i>tzdata-arctic</i> <i>tzdata-atlantic</i> <i>tzdata-core</i> <i>tzdata-misc</i> <i>tzdata-posix</i>	2019c  <i>tzdata-africa</i> <i>tzdata-antarctica</i> <i>tzdata-asia</i> <i>tzdata-australia</i> <i>tzdata-europe</i> <i>tzdata-pacific</i> <i>tzdata-right</i>	PD, BSD, BSD-3-Clause
<b>u-boot-denix</b> <i>u-boot-denix</i>	1.0	GPLv2
<b>udev-extraconf</b> <i>udev-extraconf</i>	1.1	MIT
<b>udev-rules-imx</b> <i>udev-rules-imx</i>	1.0	MIT
<b>update-flash</b> <i>update-flash</i>	git	GPLv2
<b>update-rc.d</b> <i>update-rc.d</i>	0.8	GPLv2+
<b>usbutils</b> <i>usbutils</i>	012	GPLv2+
<b>useradd-msc</b> <i>useradd-msc</i>	1.0	MIT
<b>util-linux</b>  <i>util-linux-blkid</i>	2.34  <i>util-linux-dmmsg</i>	GPLv2+, LGPLv2.1+, BSD-3-Clause, BSD-4-Clause

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

Recipe	Version	Licenses
<i>util-linux-fstrim</i>	<i>util-linux-libblkid</i>	
<i>util-linux-libmount</i>	<i>util-linux-libsmartcols</i>	
<i>util-linux-libuuid</i>	<i>util-linux-lsblk</i>	
<i>util-linux-lscpu</i>	<i>util-linux-mount</i>	
<i>util-linux-sulogin</i>	<i>util-linux-swapoff</i>	
<i>util-linux-swapon</i>	<i>util-linux-swaponoff</i>	
<i>util-linux-umount</i>		
<b>wireless-regdb</b>	2019.06.03	ISC
<i>wireless-regdb-static</i>		
<b>wpa-supPLICANT</b>	2.9	BSD
<i>wpa-supPLICANT</i>	<i>wpa-supPLICANT-cli</i>	
<i>wpa-supPLICANT-passphrase</i>		
<b>xz</b>	5.2.4	PD
<i>liblzma</i>		
<b>zip</b>	3.0	BSD-3-Clause
<i>zip</i>		
<b>zlib</b>	1.2.11	Zlib
<i>zlib</i>		
<b>zsh</b>	5.4.2	zsh
<i>zsh</i>		
<b>zsh-config</b>	1.0	MIT
<i>zsh-config</i>		