Intelligent Solutions

msc technologies
an Avnet Company

Engineering Leadership

MSC-LDK Manual
BSP 0584_nanoRISC_imx6 V1.3.0

2018-06-15

Copyright Notice

Trademarks

Certification

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

Audience

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

Life-Cycle-Management

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

Disclaimer

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

Contents

List of Figures

List of Tables

Listings

1. History

1.1. Document

Revision	Changes
1.3.0	• Added chapter “Power Analysis” • Added chapter “MSC-IO” • Added section “Setup Optional Docker Container”
1.2.0	• Added chapter “Microsoft Azure” • Added chapter “Real-Time” • Added chapter “Security” • Added chapter “Tips And Tricks” • Added section “ConnMan Configuration” • Updated section “Deploying Images To The Hardware” (new image type .wic) • Updated section “Enhancing The Images” • Updated section “Installation Of MSC-LDK”
1.1.0	• Added section “Traceable and Reproducable Images” • Added section “Hotfixed and Updating MSC-LDK” • Updated section “Installation of MSC-LDK”
1.0.0	• Added section “Using LXQt” • Added section “Using the SDK images” • Added section “Bug Reporting”

1.2. MSC-LDK

Revision	Changes
1.3.0	• 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 reproducable images) • Added certified azure libraries to standard image • Supporting speaking names for BSPs, e.g. Q7-BT or Baytrail instead of C984 • Added fallback mirror server ftp4.ebv.com
1.0.0	• Updated to yocto 2.0 (jethro) • Added layer lxqt. • Simplified adding additional layers to the BSP builds. • Simplified adding new BSPs. • Moved recipes into own layers (e.g. meta-msc-ldk-core.git). • Added support for “develop” and feature branches of MSC-LDK development.

1.3. BSP 0584_nanoRISC_imx6

Revision	Changes
0.3.0	• Updated to kernel 4.1.15
0.2.0	• Added nanoRISC i.MX6 modules
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 2.4 project.

2.1. Scope

This document gives a hand in:

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

2.2. Out Of Scope

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

2.3. Features

Features of the MSC-LDK are:

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

2.4. Conventions

This section describes the conventions used in this manual.

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

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

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

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

Typographical conventions:

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

2.5. Product Support

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

3. Getting Started

3.1. Requirements

- Linux x86 development host (32bit or 64bit).
- Ubuntu 16.04 (LTS), but other distributions may also work.
- Internet access for downloading packages (HTTP, FTP, Git and SSH).
- Registration on the MSC Git server.
- Lots of free disc space for the initial build (>128 GB).
- Python3 with 'pip' installed (at least Python v3.3).
- To use the certified azure libraries, git v2.11 is required (otherwise an older git version works as well).

3.2. Registration On The MSC Git Server

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

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

3.2.1. Creating An SSH Key

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

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

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

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

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

3.2.2. Configuring For HTTP proxies

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

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

4. Using MSC-LDK

4.1. Yocto/MSC-LDK Terms

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

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

4.2. Setup Optional Docker Container

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

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

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

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

e.g. to:

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

To install the container:

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/docker-msc-ldk
user@devhost:$ cd docker-msc-ldk
user@devhost:$ git checkout v1.3.0
```

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

To use the container on an Ubuntu host:

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

To use the container on a CentOS host:

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

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

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

4.3. Installation Of MSC-LDK

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

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/msc-ldk
user@devhost:$ cd msc-ldk
user@devhost:$ git checkout v1.3.0
```

No files will be installed in other directories.

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

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

The question has to be answered with "no"

4.4. Directory Layout

Table 4.1. – MSC-LDK Directory Layout

Directory	Contents
`build/0584`	BSP build directory.
`doc`	This contains all the generated Yocto documentation.

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

Directory	Contents
	(only if build with: make doc)
downloads/	All downloaded sources are stored here. This directory can be shared with other MSC-LDK installations.
scripts/	Build helper scripts.
sources/0584/*.git	BSP specific layers. (will be created in chapter 4.5)
sources/meta-freescale.git/	Freescale specific layers, only on Freescale architectures.
sources/meta-intel-*.git/	Intel specific layers, only for Intel architectures.
sources/meta-msc-ldk-*.git/	MSC-LDK layers.
sources/meta-openembedded-*.git/	Additional useful tools that are not part of Yocto.
sources/meta-qt5*.git/	Qt5 layers used by msc-image-lxqt.
sources/yocto.git/	Yocto sources used by MSC-LDK.
sstate-cache/	All built files are stored here and reused on next build. This directory can be shared with other MSC-LDK installations.

4.4.1. MSC-LDK Layers

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

Figure 4.1. – MSC-LDK Layers

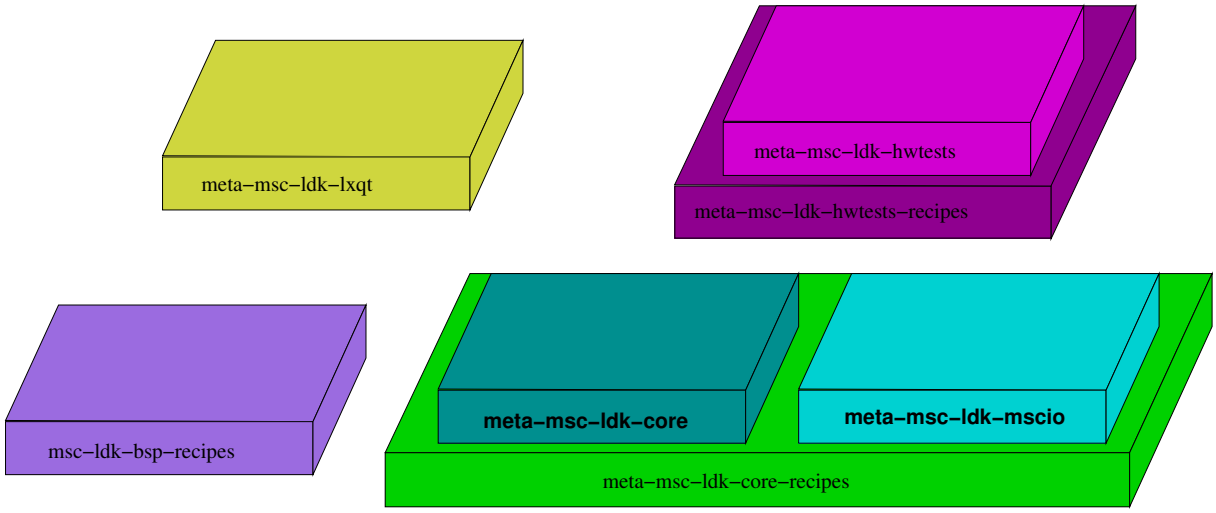


Table 4.2. – MSC-LDK Layer Description

Layer	Description
msc-ldk-bsp-recipes	BSP related layer, e.g. Kernel patches for a board. Mandatory for a given board.
meta-msc-ldk-core-recipes	Required recipes to build MSC specific applications.
meta-msc-ldk-core	MSC specific image recipes and package groups. Depends on meta-msc-ldk-core-recipes.
meta-msc-ldk-hwtests-recipes	Hardware test specific recipes.
meta-msc-ldk-hwtests	Board specific image msc-image-hwtests-<BOARD>. Depends on meta-msc-ldk-hwtests-recipes.
meta-msc-ldk-mscio	MSC-IO layer. Depends on meta-msc-ldk-core-recipes.

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

Layer	Description
meta-msc-ldk-lxqt	Support for the Lightweight Qt Desktop Environment LXQt (http://lxqt.org).

4.5. Installation Of MSC-LDK BSPs

This BSP can be built in a several variants

4.5.1. Setup for target

To setup MSC-LDK to generate images for this target, you have to select the cpu-class of your module, this is coded in the Boardmark. Refer to the following table:

mid digit	CPU-Type	CPU-Class
1	Solo	imx6dl
2	Dual	imx6q
3	DualLite	imx6dl
4	Quad	imx6q
6	Dual Plus	imx6q
8	Quad Plus	imx6q

So you command to setup for target is:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --variant=imx6dl
```

or

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --variant=imx6q
```

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-imx6dl
user@devhost:msc-ldk/build/0584-imx6dl$
```

or

```
user@devhost:msc-ldk$ cd build/0584-imx6q
user@devhost:msc-ldk/build/0584-imx6q$
```

This two variants are only for images using accelerated graphics, e.g. msc-image-qt5, other images can be build without –variant

Further steps, building a dedicated image, are described in section 4.10

4.5.2. 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 --variant=<same as above>  --re-create-conf
```

And then continue as usual:

```
user@devhost:msc-ldk$ cd build/0584[-imx6dl/imx6q]
user@devhost:msc-ldk/build/0584[-imx6dl/imx6q]$
```

4.6. BSP Aliases

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

4.7. Building Images

To build all supported images for all installed BSPs, 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
```

4.8. Traceable And Reproducable Images

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

```
Build Configuration:
BB_VERSION        = "1.32.0"
BUILD_SYS         = "x86_64-linux"
NATIVELSBSTRING   = "universal"
TARGET_SYS        = "arm-poky-linux-gnueabi"
MACHINE           = "msc-nr-imx6"
DISTRO            = "poky"
DISTRO_VERSION    = "2.2"
TUNE_FEATURES     = "arm armv7a vfp thumb neon       callconvention-hard
    cortexa9"
TARGET_FPU        = "hard"
meta-poky
meta-yocto-bsp    = "morty-msc:643a1bc96c5577a9334544a4a3b26e4242e689f5"
meta              = "master:4b55f4999ecee5aef498f2ee149b00fb93224206"
meta              = "morty-msc:643a1bc96c5577a9334544a4a3b26e4242e689f5"
meta-oe
meta-networking
meta-python       = "morty-msc:0bc4b2bc42363fcfde8d3a35490e60cf2880e46d"
meta-qt5.git      = "morty-msc:7128e252f83a77e04e4b53c90864f2e11677568c"
meta-msc-ldk-core-recipes.git = "master:3636c544dadbe617bc04fc8c06ee886980c95053"
meta-msc-ldk-core.git = "master:327dfb039c5a7195c4f3b78750ecc3dd66c22f97"
meta-msc-ldk-mscio.git = "master:2eef4699a0e9f509cfc67aeae87de44f87edbc89"
meta-freescale.git = "morty:38bd14cddf1ce325de43f1fb29f01551b48384bb"
meta-qt5-extra.git = "morty-msc:24d9416e425639925875fadafe3ec7b3403b2091"
```

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-gnuabi/<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.

4.9. 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.

4.10. Image Types

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

4.10.1. 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. `enp1s0`.

4.10.2. LXQt

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

It can be selectively built for Solo and DualLight CPUs by:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --layers-lxqt -variant=imx6dl
user@devhost:msc-ldk$ cd build/0584-imx6dl-lxqt
user@devhost:msc-ldk/build/0584-imx6dl-lxqt$ make msc-image-lxqt
```

and for Dual, Quad, DualPlus and QuadPlus CPUs:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --layers-lxqt -variant=imx6q
user@devhost:msc-ldk$ cd build/0584-imx6q-lxqt
user@devhost:msc-ldk/build/0584-imx6q-lxqt$ make msc-image-lxqt
```

4.10.3. LXQt SDK

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

It must be built for Solo and DualLight CPUs by:

```
user@devhost:msc-ldk$ cd build/0584-imx6dl-lxqt
user@devhost:msc-ldk/build/0584-imx6dl-lxqt$ make msc-image-lxqt-sdk
```

and for Dual, Quad, DualPlus and QuadPlus CPUs:

```
user@devhost:msc-ldk$ cd build/0584-imx6q-lxqt
user@devhost:msc-ldk/build/0584-imx6q-lxqt$ make msc-image-lxqt-sdk
```

4.10.4. 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

4.10.5. 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
```

4.11. 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).

4.11.1. SD card

To create a bootable SD Card with a root filesysystem, 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
```

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 2015.07-0584_nanoRISC_iMX6_develop+g4b22770 (Jan 31 2017 - 17:14:19)
Boardinfo:
  name ......... msc-nR-imx6
  variant ...... 123
  feature ...... 014
  serial ....... 1234567899
  revision ..... N/A
Booting from MMC1


U-Boot 2015.07-0584_nanoRISC_iMX6_develop+g4b22770 (Jan 31 2017 - 17:14:19 +0100)

CPU:   Freescale i.MX6D rev1.5 at 792MHz
CPU:   Industrial temperature grade (-40C to 105C) at 46C
Reset cause: POR
Board: MSC nanoRISC i.MX6
I2C:   ready
DRAM:  2 GiB
LCD, HDMI,
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
*** Warning - bad CRC, using default environment

Display: hdmi:1024x768M@60 (1024x768)
In:    serial
Out:   serial
Err:   serial
SF: Detected AT25SF041 with page size 256 Bytes, erase size 4 KiB, total 512 KiB
Net:   FEC [PRIME]
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... 1 USB Device(s) found
USB1:   USB EHCI 1.00
scanning bus 1 for devices... 1 USB Device(s) found
       scanning usb for storage devices... 0 Storage Device(s) found

USB device 0: unknown device
ERR: USB start failed
Attempting mmc boot...
switch to partitions #0, OK
mmc0 is current device
Loading environment (uEnv.txt) from MMC0 ...
reading uEnv.txt
10 bytes read in 10 ms (1000 Bytes/s)
Importing environment (uEnv.txt) ...
Loading linux image (boot/zImage) from MMC0 ...
```

```
5482800 bytes read in 443 ms (11.8 MiB/s)
Booting from MMC0 ...
Loading FDT image (boot/msc-nR-imx6-123-014-headless.dtb) from MMC0 ...
46607 bytes read in 652 ms (69.3 KiB/s)
Kernel image @ 0x12000000 [ 0x000000 - 0x53a930 ]
## Flattened Device Tree blob at 18000000
   Booting using the fdt blob at 0x18000000
   Using Device Tree in place at 18000000, end 1800e60e

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 4.1.15-yocto-standard (buildserver@destsm3ux05bs01.emea.avnet.com) (gcc version 6.2.0 (GCC) ) #1 SMP PREEMPT Sat Feb 11 04:55:11 CET 2017
[    0.000000] CPU: ARMv7 Processor [412fc09a] revision 10 (ARMv7), cr=10c53c7d
[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
[    0.000000] Machine model: MSC nanoRISC i.MX6Q
[    0.000000] Memory policy: Data cache writealloc
[    0.000000] PERCPU: Embedded 12 pages/cpu @ee7b7000 s16908 r8192 d24052 u49152
[    0.000000] Built 1 zonelists in Zone order, mobility grouping on.  Total pages: 520720
[    0.000000] Kernel command line: console=ttymxc1,115200 root=/dev/mmcblk0p2 rootwait rw consoleblank=0
[    0.000000] PID hash table entries: 4096 (order: 2, 16384 bytes)
[    0.000000] Dentry cache hash table entries: 262144 (order: 8, 1048576 bytes)
[    0.000000] Inode-cache hash table entries: 131072 (order: 7, 524288 bytes)
[    0.000000] Memory: 1740584K/2097152K available (7287K kernel code, 250K rwdata, 2320K rodata, 308K init, 429K bss, 356568K reserved, 0K cma-reserved, 270336K highmem)
[    0.000000] Virtual kernel memory layout:
[    0.000000]     vector  : 0xffff0000 - 0xffff1000   (   4 kB)
[    0.000000]     fixmap  : 0xffc00000 - 0xfff00000   (3072 kB)
[    0.000000]     vmalloc : 0xf0000000 - 0xff000000   ( 240 MB)
[    0.000000]     lowmem  : 0x80000000 - 0xef800000   (1784 MB)
[    0.000000]     pkmap   : 0x7fe00000 - 0x80000000   (   2 MB)
[    0.000000]     modules : 0x7f000000 - 0x7fe00000   (  14 MB)
[    0.000000]       .text : 0x80008000 - 0x8096a06c   (9609 kB)
[    0.000000]       .init : 0x8096b000 - 0x809b8000   ( 308 kB)
[    0.000000]       .data : 0x809b8000 - 0x809f6b60   ( 251 kB)
[    0.000000]        .bss : 0x809f6b60 - 0x80a620f8   ( 430 kB)
[    0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=2, Nodes=1
[    0.000000] Preemptible hierarchical RCU implementation.
[    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 16
[    0.000000] L2C-310 errata 752271 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, 1024 kB
[    0.000000] L2C-310: CACHE_ID 0x410000c7, AUX_CTRL 0x76470001
[    0.000000] mxc_clocksource_init 3000000
[    0.000000] Switching to timer-based delay loop, resolution 333ns
[    0.000006] 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
[    0.000964] Console: colour dummy device 80x30
[    0.000984] Calibrating delay loop (skipped), value calculated using timer frequency.. 6.00 BogoMIPS (lpj=30000)
[    0.000998] pid_max: default: 32768 minimum: 301
[    0.001090] Mount-cache hash table entries: 4096 (order: 2, 16384 bytes)
[    0.001103] Mountpoint-cache hash table entries: 4096 (order: 2, 16384 bytes)
[    0.001725] CPU: Testing write buffer coherency: ok
[    0.001988] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[    0.002087] Setting up static identity map for 0x10008280 - 0x100082d8
[    0.060157] CPU1: thread -1, cpu 1, socket 0, mpidr 80000001
[    0.060222] Brought up 2 CPUs
[    0.060240] SMP: Total of 2 processors activated (12.00 BogoMIPS).
[    0.060248] CPU: All CPU(s) started in SVC mode.
[    0.060690] devtmpfs: initialized
[    0.070959] VFP support v0.3: implementor 41 architecture 3 part 30 variant 9 rev 4
[    0.071350] clocksource jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns: 19112604462750000 ns
[    0.071749] pinctrl core: initialized pinctrl subsystem
[    0.072828] NET: Registered protocol family 16
[    0.074595] DMA: preallocated 256 KiB pool for atomic coherent allocations
[    0.099680] cpuidle: using governor ladder
[    0.129676] cpuidle: using governor menu
[    0.129792] CPU identified as i.MX6Q, silicon rev 1.5
[    0.153495] hw-breakpoint: found 5 (+1 reserved) breakpoint and 1 watchpoint registers.
[    0.153508] hw-breakpoint: maximum watchpoint size is 4 bytes.
[    0.154482] imx6q-pinctrl 20e0000.iomuxc: initialized IMX pinctrl driver
[    0.192564] mxs-dma 110000.dma-apbh: initialized
[    0.195596] SCSI subsystem initialized
[    0.196046] usbcore: registered new interface driver usbfs
[    0.196111] usbcore: registered new interface driver hub
[    0.196201] usbcore: registered new device driver usb
[    0.197088] i2c i2c-0: IMX I2C adapter registered
[    0.197106] i2c i2c-0: can't use DMA
[    0.197476] i2c i2c-1: IMX I2C adapter registered
```

```
[    0.197492] i2c i2c-1: can't use DMA
[    0.197788] Linux video capture interface: v2.00
[    0.197847] pps_core: LinuxPPS API ver. 1 registered
[    0.197856] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <giometti@linux.it>
[    0.197887] PTP clock support registered
[    0.209726] imx-ipuv3 2400000.ipu: IPU DMFC NORMAL mode: 1(0~1), 5B(4,5), 5F(6,7)
[    0.229724] imx-ipuv3 2800000.ipu: IPU DMFC NORMAL mode: 1(0~1), 5B(4,5), 5F(6,7)
[    0.230917] MIPI CSI2 driver module loaded
[    0.231053] Advanced Linux Sound Architecture Driver Initialized.
[    0.232419] Switched to clocksource mxc_timer1
[    0.244528] NET: Registered protocol family 2
[    0.245302] TCP established hash table entries: 16384 (order: 4, 65536 bytes)
[    0.245464] TCP bind hash table entries: 16384 (order: 5, 131072 bytes)
[    0.245750] TCP: Hash tables configured (established 16384 bind 16384)
[    0.245927] UDP hash table entries: 1024 (order: 3, 32768 bytes)
[    0.246005] UDP-Lite hash table entries: 1024 (order: 3, 32768 bytes)
[    0.246252] NET: Registered protocol family 1
[    0.246577] RPC: Registered named UNIX socket transport module.
[    0.246588] RPC: Registered udp transport module.
[    0.246595] RPC: Registered tcp transport module.
[    0.246602] RPC: Registered tcp NFSv4.1 backchannel transport module.
[    0.247212] CPU PMU: Failed to parse /soc/pmu/interrupt-affinity[0]
[    0.247261] hw perfevents: enabled with armv7_cortex_a9 PMU driver, 7 counters available
[    0.248738] Bus freq driver module loaded
[    0.249790] futex hash table entries: 512 (order: 3, 32768 bytes)
[    0.256737] VFS: Disk quotas dquot_6.6.0
[    0.256930] VFS: Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
[    0.258841] NFS: Registering the id_resolver key type
[    0.258877] Key type id_resolver registered
[    0.258886] Key type id_legacy registered
[    0.259187] fuse init (API version 7.23)
[    0.261030] bounce: pool size: 64 pages
[    0.261049] io scheduler noop registered
[    0.261066] io scheduler deadline registered
[    0.261112] io scheduler cfq registered (default)
[    0.261614] imx-weim 21b8000.weim: Driver registered.
[    0.263020] MIPI DSI driver module loaded
[    0.264680] imx-sdma 20ec000.sdma: no iram assigned, using external mem
[    0.264808] imx-sdma 20ec000.sdma: no event needs to be remapped
[    0.264985] imx-sdma 20ec000.sdma: loaded firmware 3.3
[    0.268297] imx-sdma 20ec000.sdma: initialized
[    0.269276] 2020000.serial: ttymxc0 at MMIO 0x2020000 (irq = 26, base_baud = 5000000) is a IMX
[    0.269953] 21e8000.serial: ttymxc1 at MMIO 0x21e8000 (irq = 295, base_baud = 5000000) is a IMX
[    0.944148] console [ttymxc1] enabled
[    0.948382] 21ec000.serial: ttymxc2 at MMIO 0x21ec000 (irq = 296, base_baud = 5000000) is a IMX
[    0.958993] [drm] Initialized drm 1.1.0 20060810
[    0.964153] [drm] Initialized vivante 1.0.0 20120216 on minor 0
[    0.979786] brd: module loaded
[    0.989039] loop: module loaded
[    0.993479] ahci-imx 2200000.sata: fsl,transmit-level-mV not specified, using 00000024
[    1.001419] ahci-imx 2200000.sata: fsl,transmit-boost-mdB not specified, using 00000480
[    1.009510] ahci-imx 2200000.sata: fsl,transmit-atten-16ths not specified, using 00002000
[    1.017721] ahci-imx 2200000.sata: fsl,receive-eq-mdB not specified, using 05000000
[    1.028741] ahci-imx 2200000.sata: SSS flag set, parallel bus scan disabled
[    1.035771] ahci-imx 2200000.sata: AHCI 0001.0300 32 slots 1 ports 3 Gbps 0x1 impl platform mode
[    1.044633] ahci-imx 2200000.sata: flags: ncq sntf stag pm led clo only pmp pio slum part ccc apst
[    1.055233] scsi host0: ahci-imx
[    1.058828] ata1: SATA max UDMA/133 mmio [mem 0x02200000-0x02203fff] port 0x100 irq 306
[    1.070270] nand: No NAND device found
[    1.075699] at25 spi0.0: 8 KByte at25 eeprom, pagesize 32
[    1.081121] spi_imx 2008000.ecspi: probed
[    1.085875] spi_imx 2014000.ecspi: probed
[    1.090925] m25p80 spi32766.0: found at25sf041, expected m25p16
[    1.096997] m25p80 spi32766.0: at25sf041 (512 Kbytes)
[    1.102082] 4 ofpart partitions found on MTD device spi32766.0
[    1.107940] Creating 4 MTD partitions on "spi32766.0":
[    1.113104] 0x000000000000-0x000000020000 : "SPL"
[    1.118938] 0x000000020000-0x000000040000 : "env"
[    1.124667] 0x000000040000-0x0000000c0000 : "uboot"
[    1.129559] mtd: partition "uboot" extends beyond the end of device "spi32766.0" -- size truncated to 0x40000
[    1.140456] 0x0000000c0000-0x000000080000 : "user"
[    1.145281] mtd: partition "user" is out of reach -- disabled
[    1.152234] at25 spi32766.1: 8 KByte at25 eeprom, pagesize 32
[    1.158469] CAN device driver interface
[    1.162564] 2090000.flexcan supply xceiver not found, using dummy regulator
[    1.170378] flexcan 2090000.flexcan: device registered (reg_base=f02e0000, irq=31)
[    1.179089] 2188000.ethernet supply phy not found, using dummy regulator
[    1.186358] pps pps0: new PPS source ptp0
[    1.194977] libphy: fec_enet_mii_bus: probed
[    1.199825] fec 2188000.ethernet eth0: registered PHC device 0
[    1.205928] e1000e: Intel(R) PRO/1000 Network Driver - 2.3.2-k
[    1.211774] e1000e: Copyright(c) 1999 - 2014 Intel Corporation.
```

```
[    1.217801] igb: Intel(R) Gigabit Ethernet Network Driver - version 5.2.15-k
[    1.224880] igb: Copyright (c) 2007-2014 Intel Corporation.
[    1.231706] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    1.238270] ehci-pci: EHCI PCI platform driver
[    1.243114] usbcore: registered new interface driver usb-storage
[    1.250196] 2184800.usbmisc supply vbus-wakeup not found, using dummy regulator
[    1.262166] ci_hdrc ci_hdrc.0: EHCI Host Controller
[    1.267100] ci_hdrc ci_hdrc.0: new USB bus registered, assigned bus number 1
[    1.292451] ci_hdrc ci_hdrc.0: USB 2.0 started, EHCI 1.00
[    1.298024] usb usb1: New USB device found, idVendor=1d6b, idProduct=0002
[    1.304847] usb usb1: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    1.312080] usb usb1: Product: EHCI Host Controller
[    1.316978] usb usb1: Manufacturer: Linux 4.1.15-yocto-standard ehci_hcd
[    1.323702] usb usb1: SerialNumber: ci_hdrc.0
[    1.328795] hub 1-0:1.0: USB hub found
[    1.332640] hub 1-0:1.0: 1 port detected
[    1.340655] ci_hdrc ci_hdrc.1: EHCI Host Controller
[    1.345615] ci_hdrc ci_hdrc.1: new USB bus registered, assigned bus number 2
[    1.372462] ci_hdrc ci_hdrc.1: USB 2.0 started, EHCI 1.00
[    1.378031] usb usb2: New USB device found, idVendor=1d6b, idProduct=0002
[    1.384853] usb usb2: New USB device strings: Mfr=3, Product=2, SerialNumber=1
[    1.392087] usb usb2: Product: EHCI Host Controller
[    1.397005] usb usb2: Manufacturer: Linux 4.1.15-yocto-standard ehci_hcd
[    1.403732] usb usb2: SerialNumber: ci_hdrc.1
[    1.408750] hub 2-0:1.0: USB hub found
[    1.412568] ata1: SATA link down (SStatus 0 SControl 300)
[    1.418009] ahci-imx 2200000.sata: no device found, disabling link.
[    1.424353] hub 2-0:1.0: 1 port detected
[    1.429450] mousedev: PS/2 mouse device common for all mice
[    1.435080] ahci-imx 2200000.sata: pass ahci_imx..hotplug=1 to enable hotplug
[    1.442582] usbcore: registered new interface driver usbtouchscreen
[    1.449884] snvs_rtc 20cc000.snvs:snvs-rtc-lp: rtc core: registered 20cc000.snvs:snvs-r as
     rtc0
[    1.458735] i2c /dev entries driver
[    1.463634] at24 3-0054: 512 byte 24c04 EEPROM, writable, 16 bytes/write
[    1.472969] da9063 3-0058: Device detected (chip-ID: 0x61, var-ID: 0x50)
[    1.480704] da9063 3-0058: Fault-log content is 0x00.
[    1.526278] input: da9063-onkey as /devices/soc0/soc/2100000.aips-bus/21a4000.i2c/i2c-1/
    i2c-3/3-0058/da9063-onkey/input/input0
[    1.549423] da9063-rtc da9063-rtc: rtc core: registered da9063-rtc as rtc1
[    1.557886] i2c i2c-1: Added multiplexed i2c bus 3
[    1.563233] i2c i2c-1: Added multiplexed i2c bus 4
[    1.568040] i2c-mux-gpio 2100000.aips-bus:i2cmux: 2 port mux on 21a4000.i2c adapter
[    1.577658] sdhci: Secure Digital Host Controller Interface driver
[    1.583874] sdhci: Copyright(c) Pierre Ossman
[    1.588275] sdhci-pltfm: SDHCI platform and OF driver helper
[    1.594831] /soc/aips-bus@02100000/usdhc@02190000: voltage-ranges unspecified
[    1.602149] sdhci-esdhc-imx 2190000.usdhc: No vmmc regulator found
[    1.608367] sdhci-esdhc-imx 2190000.usdhc: No vqmmc regulator found
[    1.652451] usb 1-1: new low-speed USB device number 2 using ci_hdrc
[    1.653485] mmc0: SDHCI controller on 2190000.usdhc [2190000.usdhc] using ADMA
[    1.655985] /soc/aips-bus@02100000/usdhc@02194000: voltage-ranges unspecified
[    1.656014] sdhci-esdhc-imx 2194000.usdhc: Got CD GPIO
[    1.656034] sdhci-esdhc-imx 2194000.usdhc: Got WP GPIO
[    1.661324] sdhci-esdhc-imx 2194000.usdhc: No vmmc regulator found
[    1.661330] sdhci-esdhc-imx 2194000.usdhc: No vqmmc regulator found
[    1.702250] mmc0: host does not support reading read-only switch, assuming write-enable
[    1.715445] mmc1: SDHCI controller on 2194000.usdhc [2194000.usdhc] using ADMA
[    1.723388] /soc/aips-bus@02100000/usdhc@02198000: voltage-ranges unspecified
[    1.730550] sdhci-esdhc-imx 2198000.usdhc: could not get ultra high speed state, work on
    normal mode
[    1.740739] mmc0: new high speed SDHC card at address 1234
[    1.746690] mmcblk0: mmc0:1234 SA08G 7.24 GiB
[    1.752592]  mmcblk0: p1 p2
[    1.752605] sdhci-esdhc-imx 2198000.usdhc: No vmmc regulator found
[    1.752612] sdhci-esdhc-imx 2198000.usdhc: No vqmmc regulator found
[    1.794234] mmc2: SDHCI controller on 2198000.usdhc [2198000.usdhc] using ADMA
[    1.806835] mxc_vpu 2040000.vpu_fsl: VPU initialized
[    1.812370] mxc_vdoa 21e4000.vdoa: i.MX Video Data Order Adapter(VDOA) driver probed
[    1.821223] Galcore version 5.0.11.41671
[    1.846632] mmc1: new high speed SDHC card at address 1234
[    1.853162] mmcblk1: mmc1:1234 SA08G 7.24 GiB (ro)
[    1.860210]  mmcblk1: p1 p2
[    1.864409] usb 1-1: New USB device found, idVendor=046d, idProduct=c050
[    1.871157] usb 1-1: New USB device strings: Mfr=1, Product=2, SerialNumber=0
[    1.879681] usb 1-1: Product: USB-PS/2 Optical Mouse
[    1.887994] usb 1-1: Manufacturer: Logitech
[    1.960421] mmc2: MAN_BKOPS_EN bit is not set
[    1.989671] mmc2: new DDR MMC card at address 0001
[    2.002473] mmcblk2: mmc2:0001 SEM04G 3.68 GiB
[    2.007194] mmcblk2boot0: mmc2:0001 SEM04G partition 1 2.00 MiB
[    2.013351] mmcblk2boot1: mmc2:0001 SEM04G partition 2 2.00 MiB
[    2.057668] ledtrig-cpu: registered to indicate activity on CPUs
[    2.070380] input: Logitech USB-PS/2 Optical Mouse as /devices/soc0/soc/2100000.aips-bus
    /2184000.usb/ci_hdrc.0/usb1/1-1/1-1:1.0/0003:046D:C050.0001/input/input1
[    2.085175] hid-generic 0003:046D:C050.0001: input: USB HID v1.10 Mouse [Logitech USB-PS/2
     Optical Mouse] on usb-ci_hdrc.0-1/input0
[    2.097138] usbcore: registered new interface driver usbhid
[    2.102737] usbhid: USB HID core driver
```

```
[    2.109375] fsl-asrc 2034000.asrc: driver registered
[    2.117907] imx-tlv320aic3107 sound: tlv320aic3x-hifi <-> 2028000.ssi mapping ok
[    2.128229] NET: Registered protocol family 10
[    2.133743] sit: IPv6 over IPv4 tunneling driver
[    2.139101] NET: Registered protocol family 17
[    2.143595] can: controller area network core (rev 20120528 abi 9)
[    2.149860] NET: Registered protocol family 29
[    2.154376] can: raw protocol (rev 20120528)
[    2.158664] can: broadcast manager protocol (rev 20120528 t)
[    2.164370] can: netlink gateway (rev 20130117) max_hops=1
[    2.170058] 8021q: 802.1Q VLAN Support v1.8
[    2.174320] Key type dns_resolver registered
[    2.192194] 1ffc000.pcie supply pcie-bus not found, using dummy regulator
[    2.531908] imx6q-pcie 1ffc000.pcie: phy link never came up
[    2.537531] imx6q-pcie 1ffc000.pcie: failed to initialize host
[    2.543486] imx6q-pcie: probe of 1ffc000.pcie failed with error -22
[    2.553554] input: keypad as /devices/soc0/keypad/input/input2
[    2.561572] da9063-rtc da9063-rtc: setting system clock to 2017-02-11 04:41:51 UTC
    (1486788111)
[    2.583470] ALSA device list:
[    2.586458]   #0: imx-tlv320aic3107
[    2.599470] EXT3-fs (mmcblk0p2): error: couldn't mount because of unsupported optional
    features (240)
[    2.609886] EXT2-fs (mmcblk0p2): error: couldn't mount because of unsupported optional
    features (244)
[    2.765313] EXT4-fs (mmcblk0p2): recovery complete
[    2.772219] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
[    2.780380] VFS: Mounted root (ext4 filesystem) on device 179:2.
[    2.788707] devtmpfs: mounted
[    2.792340] Freeing unused kernel memory: 308K (8096b000 - 809b8000)
INIT: version 2.88 booting
Starting udev
[    3.413099] udevd[110]: starting version 3.2
[    3.426645] random: udevd urandom read with 24 bits of entropy available
[    3.474335] udevd[111]: starting eudev-3.2
[    3.697014] EXT4-fs (mmcblk0p2): re-mounted. Opts: data=ordered
[    4.874992] ERROR: v4l2 capture: slave not found!
[    5.069908] EXT4-fs (mmcblk1p2): mounted filesystem with ordered data mode. Opts: (null)
[    5.107754] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be
    corrupt. Please run fsck.
ALSA: Restoring mixer settings...
No state is present for card imxtlv320aic310
Found hardware: "imx-tlv320aic31" "" "" "" ""
Hardware is initialized using a generic method
No state is present for card imxtlv320aic310
INIT: Entering runlevel: 5
Configuring network interfaces... [    5.623142] fec 2188000.ethernet eth0: Freescale FEC PHY
     driver [Micrel KSZ9031 Gigabit PHY] (mii_bus:phy_addr=2188000.ethernet:00, irq=-1)
[    5.635803] IPv6: ADDRCONF(NETDEV_UP): eth0: link is not ready
udhcpc (v1.24.1) started
Sending discover...
Sending discover...
[    9.623122] fec 2188000.ethernet eth0: Link is Up - 1Gbps/Full - flow control rx/tx
[    9.630824] IPv6: ADDRCONF(NETDEV_CHANGE): eth0: link becomes ready
Sending discover...
Sending select for 192.168.150.101...
Lease of 192.168.150.101 obtained, lease time 172800
/etc/udhcpc.d/50default: Adding DNS 192.168.150.250
done.
Starting system message bus: dbus.
Starting rpcbind daemon...done.
starting statd: done
Starting bluetooth
bluetoothd
Starting ntpd: done
Starting syslogd/klogd: done
 * Starting Avahi mDNS/DNS-SD Daemon: avahi-daemon
   ...done.
Starting Telephony daemon
Loading MSC-IO: Done
Starting Linux NFC daemon
Starting internet superserver: inetd.

MSC-LDK LC984_20161116_V1_1_0-38-ge9e4de4 built on Sat Feb 11 04:57:50 CET 2017 by
    buildserver@destsm3ux05bs01.emea.avnet.com
Poky (Yocto Project Reference Distro) 2.2 msc-nr-imx6 /dev/ttymxc1

msc-nr-imx6 login:
```

4.12. 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	Sato GUI requires no login. No password is necessary for hardware test image.
msc	msc	Standard user with sudo permissions.

4.13. Enabled Services

Enabled services are:

- telnetd
- Serial console on /dev/ttymxc1 with 115200 baud/8 bits/no parity

4.14. Enhancing The Images

4.14.1. 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/2.4/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
```

4.14.2. 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/2.4/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
```

4.14.3. 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 = ""
```

4.14.4. Time Zone Setup

The default time zone is set to `Europe/Berlin`. The time zone can customized by setting the Yocto variable `DEFAULT_TIMEZONE` in `build/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`.

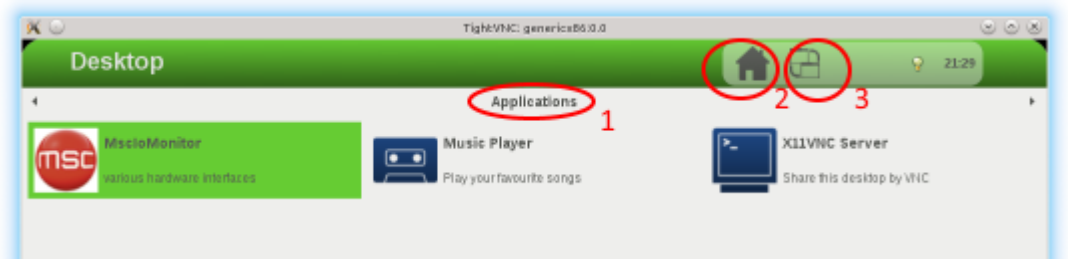
The time zone is entered in `/etc/timezone` and there is a symbolic link: `/etc/localtime->/usr/share/zoneinfo/Europe/Berlin`.

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

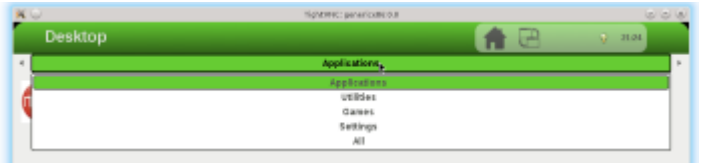
```
# rm -f /etc/localtime
# ln -s /usr/share/zoneinfo/America/New_York /etc/localtime
# echo "America/New_York" > /etc/timezone
```

4.15. Using Sato

Sato is a window manager in which applications run full-screen, optimized for use with touch-displays.
Figure 4.2. – Sato - Mainwindow



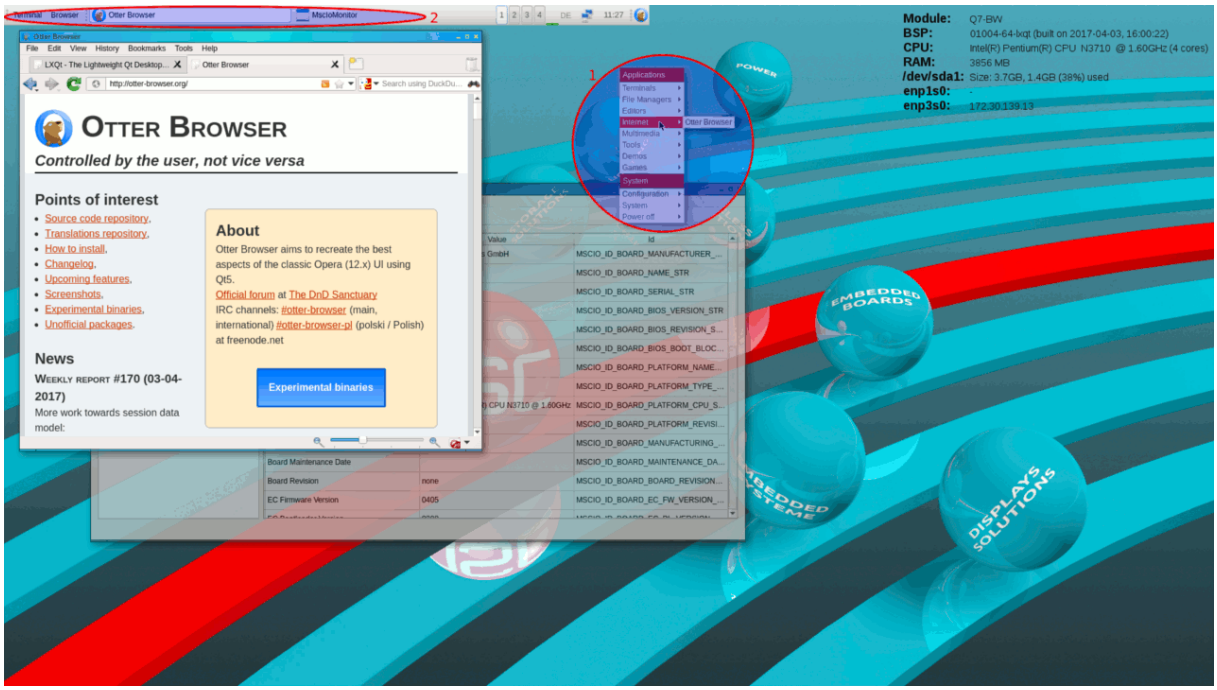
- Dropdown Box (1) - This lists the various application groups.
  Figure 4.3. – Sato - Groups



- Button (2) - This displays the application group.
- Dropdown Box (3) - This lists the running GUI applications.
  Figure 4.4. – Sato - Selecting Running Apps

4.16. Using LXQt

LXQt is a lightweight desktop for Qt based systems (http://lxqt.org).
Figure 4.5. – LXQt - Mainwindow



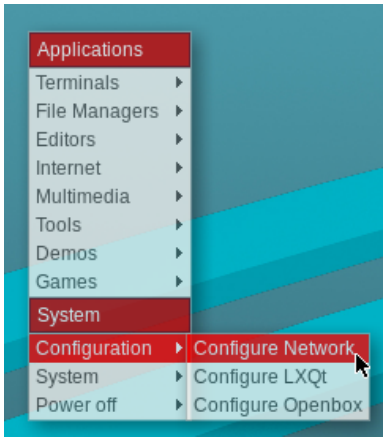
The pre-configured applications are accessible using the right mouse button.

- 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

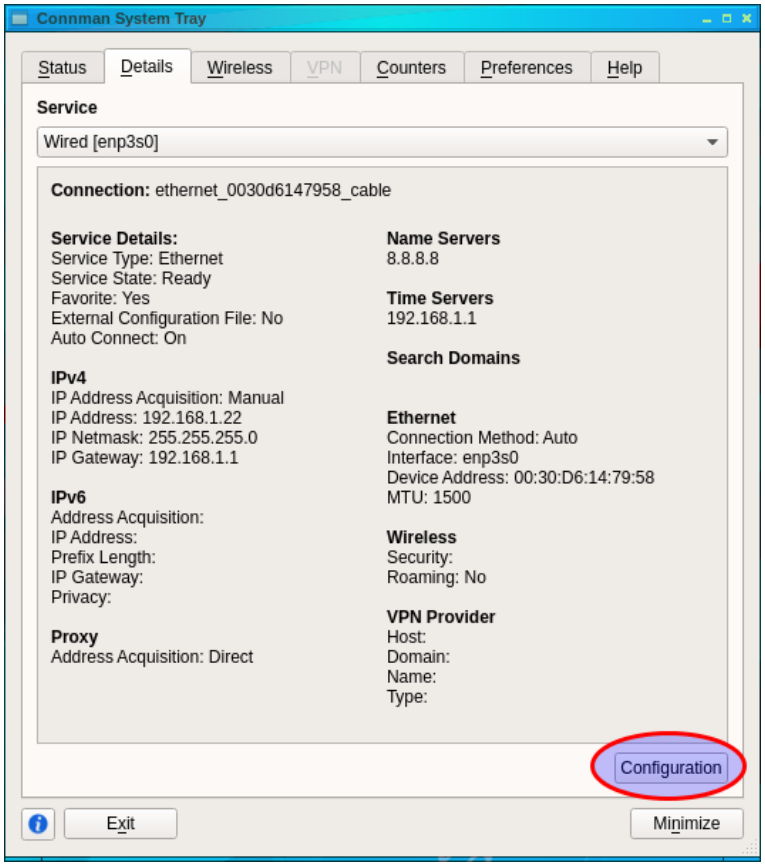
The internet connection daemon ConnMan (https://01.org/connman) is used in LXQt images to manage the internet connections.
The ConnMan Property Editor can be launched using a right-click on the LXQt desktop:

Figure 4.6. – Start ConnMan Property Editor



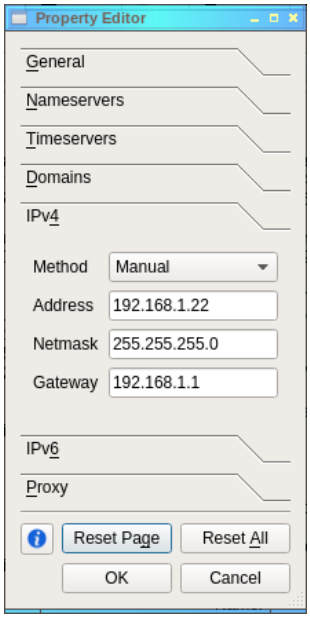
There is a detailed overview for each connection when the Details tab is selected:

Figure 4.7. – 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 4.8. – Manual IP Assignment

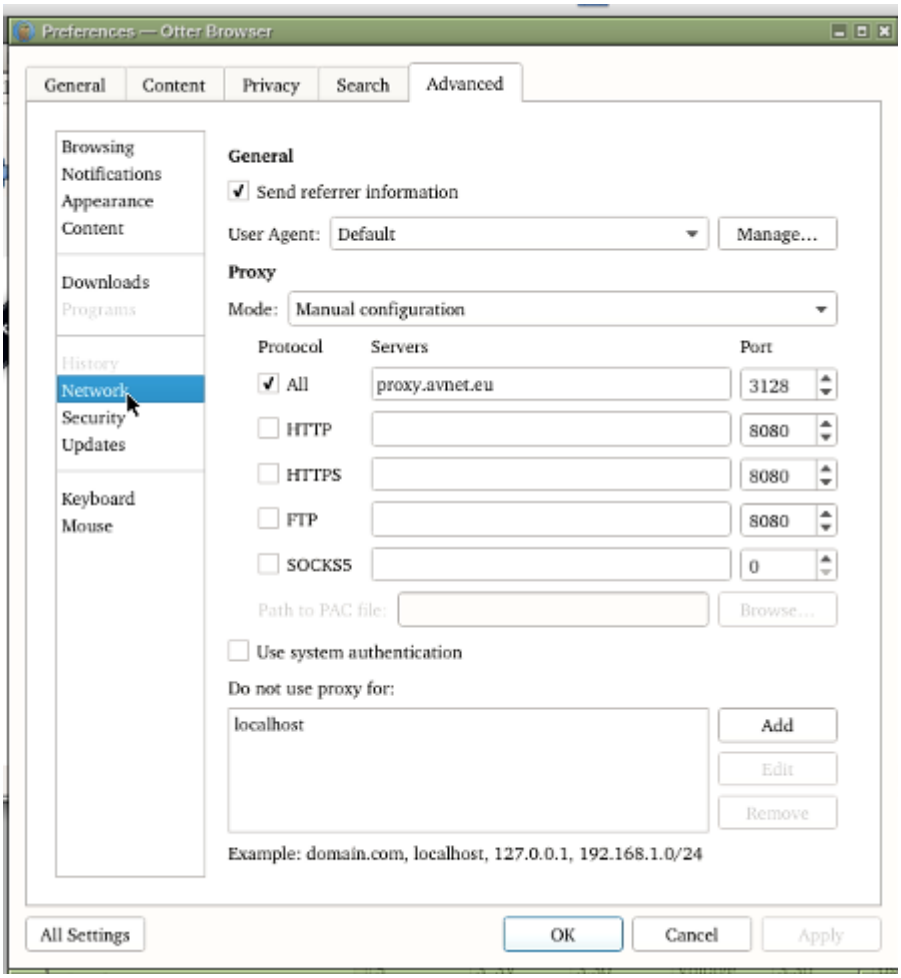


The tool `connman-info` can be used to show retrieve network interface details:

```
# 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 4.14.3).
Figure 4.9. – Otter Browser - Proxy Setup



4.18. 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.

4.18.1. 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
```

4.18.2. 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 /home/develop/src
exit


mount /home/develop/src
```

4.18.3. 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
```

4.18.4. 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
```

4.18.5. Building Qt Based hello_world (Only On LXQt Images)

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

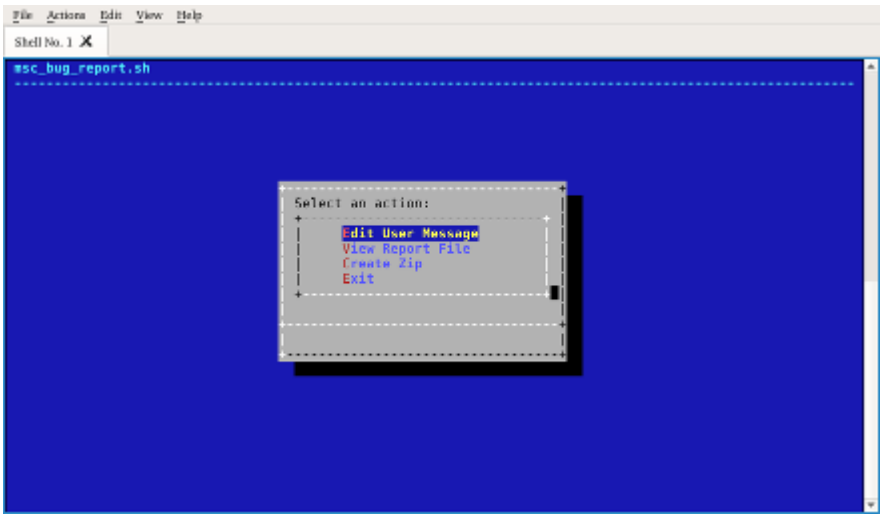
Build and run it with:

```
/usr/bin/qt5/qmake hello_world.pro && make && ./hello_world
```

4.19. Bug Reporting

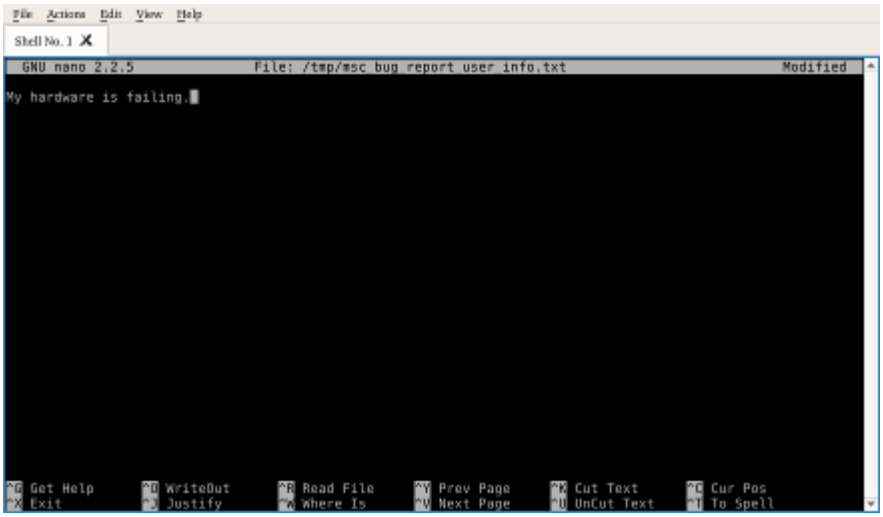
To simplify collecting information necessary for effectively responding to bug reports, use the tool `msc_bug_report.sh` to generate bug report message. It will collect all necessary information like hardware, kernel logs etc.

- Run `msc_bug_report.sh`.
  Figure 4.10. – Bug Reporting - Start

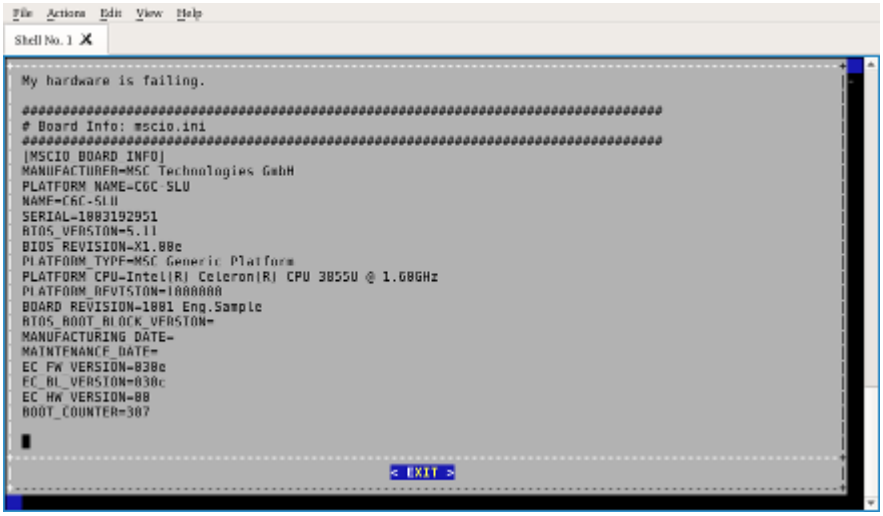


- Select “Edit User Message”.
- Enter bug report message and press Ctrl-O and Ctrl-X.

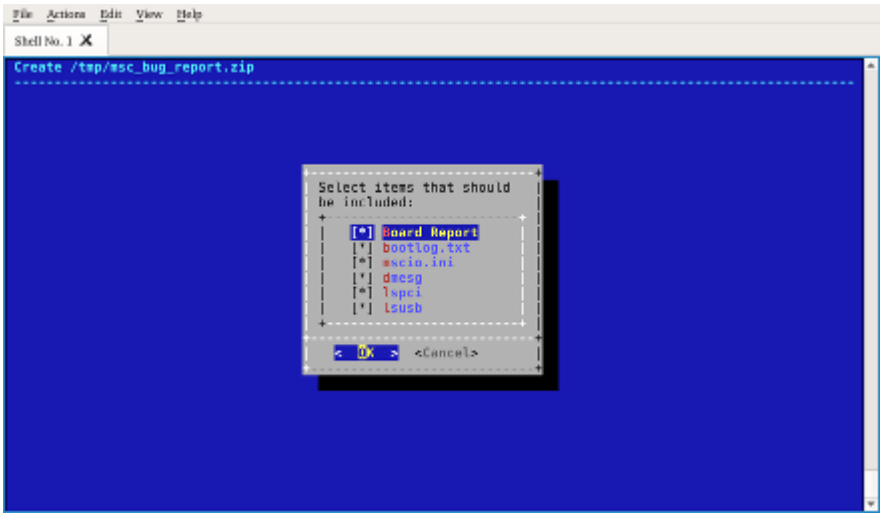
Figure 4.11. – Bug Reporting - Start



- Optionally you can then view the message with the board report (hardware information).
  Figure 4.12. – Bug Reporting - View

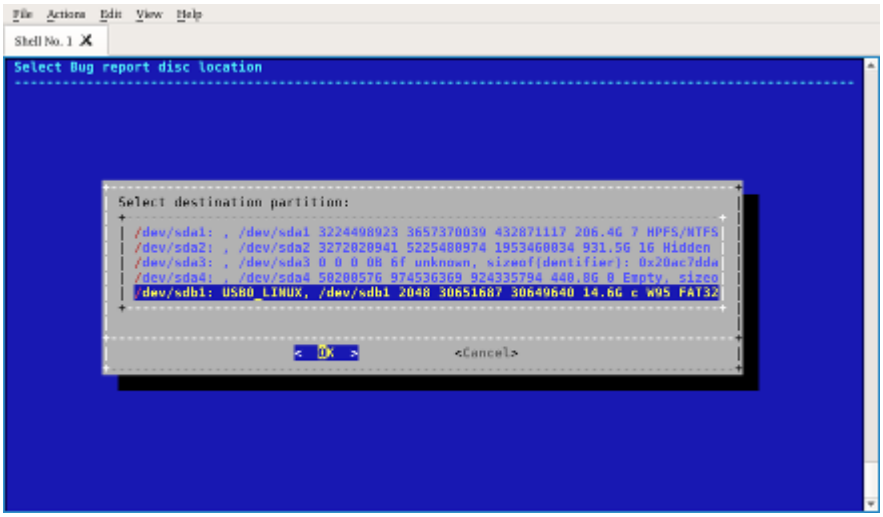


- Press “Create a zip file” and select the components you want to send (e.g. bootlog, mscio.ini, last kernel logs (dmesg) or the installed hardware).
  Figure 4.13. – Bug Reporting - Create Zip



- 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)

4.20. 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 v1.3.0`) as described above.

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

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

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

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

5. Package List

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

5.1. eapi-so

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

Dependencies: mscio.

5.2. mscio

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

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

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

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

5.3. system-test-controller

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

A. Glossary

AAOT Avnet Azure On Boarding Tool. 89, 91–95

BSP Board Support Package. 17

EAPI Embedded Application Programmer Interface. 41

MSC-LDK MSC Linux Development Kit. 13, 17

B. License Overview

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

Licenses		
((GPLv2+ & LGPLv2.1+) |(GPLv3+ & LGPLv3+))	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial)	(GPL-3.0 & The-Qt-Company-GPL-Exception-1.0 |The-Qt-Company-Commercial)
(GPLv3 & Elfutils-Exception)	(LGPL-2.1 & The-Qt-Company-Qt-LGPL-Exception-1.1 |LGPL-3.0)	(LGPLv2.1 |MPL-1.1 |BSD)
(MPL-2.0 & GPL-2.0+)	(MPL-2.0 & LGPL-2.1+)	AFL-2
Apache-2.0	Artistic-1.0	BSD
BSD-2-Clause	BSD-3-Clause	BSL-1.0
CC-BY-SA-3.0	CPL-1.0	Firmware-atheros_firmware
Firmware-imx-sdma_firmware	FreeType	GFDL-1.2
GFDL-1.3	GPL-1.0+	GPL-2.0
GPL-2.0+	GPL-3.0-with-GCC-exception	GPLv1
GPLv2	GPLv2+	GPLv2.0+
GPLv3	GPLv3+	ICU
ISC	LGPL-2.0+	LGPL-2.1
LGPL-2.1+	LGPL-3.0	LGPLv2
LGPLv2+	LGPLv2.0+	LGPLv2.1
LGPLv2.1+	LGPLv3	LGPLv3+
Libpng	MIT	MIT-X
MIT-style	MPL-1	MPL-2.0
MSCv1	NTP	OFL-1.1
PD	PSF	PSFv2
Proprietary	Python-2.0	SGIv1
Sleepycat	The-Qt-Company-Commercial	Zlib
bzip2	openssl	zsh

Table B.2. – License overview listing

Recipe	Version	Licenses
acl *acl*	2.2.52 *libacl*	GPLv2+
adwaita-icon-theme *adwaita-icon-theme-symbolic*	3.24.0	LGPL-3.0, CC-BY-SA-3.0
alsa-lib *alsa-conf*	1.1.4.1 *libasound*	LGPLv2.1, GPLv2+
alsa-plugins *alsa-plugins-pulseaudio-conf* *libasound-module-ctl-pulse*	1.1.4 *libasound-module-conf-pulse* *libasound-module-pcm-pulse*	LGPLv2.1, GPLv2+
alsa-state *alsa-state*	0.2.0 *alsa-states*	MIT
alsa-utils *alsa-utils* *alsa-utils-alsactl* *alsa-utils-alsamixer* *alsa-utils-alsaucm* *alsa-utils-aplay* *alsa-utils-aseqnet* *alsa-utils-midi*	1.1.4 *alsa-utils-aconnect* *alsa-utils-alsaloop* *alsa-utils-alsatplg* *alsa-utils-amixer* *alsa-utils-aseqdump* *alsa-utils-iecset* *alsa-utils-speakertest*	GPLv2+
at-spi2-atk *at-spi2-atk*	2.24.1	LGPLv2
at-spi2-core	2.24.1	LGPLv2

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

Recipe	Version	Licenses
at-spi2-core	*at-spi2-core-locale-en-gb*	
atk *atk*	2.24.0 *atk-locale-en-gb*	GPLv2+, LGPLv2+
attr *libattr*	2.4.47	LGPLv2.1+
avahi *avahi-daemon* *libavahi-common*	0.6.32 *avahi-locale-en-gb* *libavahi-core*	GPLv2+, LGPLv2.1+
azure-iot-sdk *azure-iot-sdk*	1.1.23	MIT
azure-uamqp-c *azure-uamqp-c*	1.0.43	MIT
azure-umqtt-c *azure-umqtt-c*	1.0.43	MIT
base-files *base-files*	3.0.14	GPLv2
base-passwd *base-passwd*	3.5.29	GPLv2+
bash *bash*	4.4	GPLv3+
bc *bc*	1.06	GPLv2+, LGPLv2.1
bluez5 *bluez5* *bluez5-obex*	5.46 *bluez5-noinst-tools*	GPLv2+, LGPLv2.1+
bonnie++ *bonnie++*	1.03e *bonnie++-scripts*	GPLv2
boost *boost-filesystem* *boost-python* *boost-system*	1.64.0 *boost-iostreams* *boost-regex*	BSL-1.0, MIT, Python-2.0
bootlog *bootlog*	1.0	MIT
busybox *busybox* *busybox-syslog*	1.24.1 *busybox-hwclock* *busybox-udhcpc*	GPLv2, bzip2
bzip2 *libbz2*	1.0.6	bzip2
ca-certificates *ca-certificates*	20170717	GPL-2.0+, MPL-2.0
cairo *cairo*	1.14.10 *cairo-gobject*	MPL-1, LGPLv2.1
cifs-utils *cifs-utils*	6.4	GPLv3, LGPLv3
cmdline-keyboard *cmdline-keyboard*	1.0	MIT
cmdline-keyboard-xorg *cmdline-keyboard-xorg*	1.0	MIT
cmst *cmst*	2016.10.03	MIT
compton *compton*	v0.1-beta2	NTP, MIT
compton-conf *compton-conf*	0.3.0	LGPLv2.1
connman *connman*	1.34 *connman-client*	GPLv2
connman-info *connman-info*	1.0	GPLv2
consolekit	0.4.6	GPLv2+

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

Recipe	Version	Licenses
consolekit		
cpufrequtils *cpufrequtils*	008	GPLv2
curl *libcurl*	7.54.1	MIT
db *db*	5.3.28	Sleepycat
dbus *dbus*	1.10.20 *dbus-lib*	AFL-2, GPLv2+
dbus-glib *dbus-glib*	0.108	AFL-2, GPLv2+
demoaaot *demoaaot*	1.1.0	GPLv2
dialog *dialog*	1.3-20160828	LGPL-2.1
dnf *dnf*	2.6.3 *dnf-locale-en-gb*	GPLv2
dosfstools *dosfstools*	4.1	GPLv3
dropbear *dropbear*	2017.75	MIT, BSD-3-Clause, BSD-2-Clause, PD
e2fsprogs *e2fsprogs-badblocks* *e2fsprogs-mke2fs* *libcomerr* *libext2fs*	1.43.5 *e2fsprogs-e2fsck* *e2fsprogs-tune2fs* *libe2p*	GPLv2
eapi-so *eapi-so*	2.1.0	LGPLv2.1
efi-var *efi-var*	0.1.1	GPLv2
eject *eject*	2.1.5	GPLv2
elfutils *libdw*	0.170 *libelf*	(GPLv3 & Elfutils-Exception)
ethtool *ethtool*	4.11	GPLv2+
eudev *eudev* *libudev*	3.2.2 *eudev-hwdb*	GPLv2.0+, LGPL-2.1+
expat *expat*	2.2.3	MIT
file *file*	5.31	BSD
firmware-imx *firmware-imx-vpu-imx6q*	5.4	Proprietary
flac *libflac*	1.3.2	GFDL-1.2, GPLv2+, LGPLv2.1+, BSD
fontconfig *fontconfig*	2.12.4 *fontconfig-utils*	MIT-style, MIT, PD
formfactor *formfactor*	0.0	MIT
freetype *freetype*	2.8	FreeType, GPLv2+
gcc-runtime *libgomp*	7.3.0 *libstdc++*	GPL-3.0-with-GCC-exception
gdbm *gdbm*	1.13 *gdbm-compat*	GPLv3
gdk-pixbuf *gdk-pixbuf*	2.36.8 *gdk-pixbuf-loader-gif*	LGPLv2

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

Recipe	Version	Licenses
gdk-pixbuf-loader-jpeg	*gdk-pixbuf-loader-png*	
gdk-pixbuf-loader-xpm	*gdk-pixbuf-locale-en-gb*	
glib-2.0	2.52.3	LGPLv2+, BSD, PD
glib-2.0	*glib-2.0-locale-en-gb*	
glib-2.0-utils		
glib-networking	2.50.0	LGPLv2
glib-networking	*glib-networking-locale-en-gb*	
glibc	2.26	GPLv2, LGPLv2.1
glibc		
glibc-locale	2.26	GPLv2, LGPLv2.1
glibc-binary-localedata-de-de	*glibc-binary-localedata-en-gb*	
glibc-binary-localedata-en-us	*glibc-gconv*	
glibc-gconv-iso8859-1	*glibc-locale-en-gb*	
locale-base-de-de	*locale-base-en-gb*	
locale-base-en-us		
glmark2	2017.07	GPLv3+, SGIv1
glmark2		
gmp	6.1.2	GPLv2+, LGPLv3+
gmp		
gnome-themes-standard	3.22.3	LGPL-2.1
gnome-theme-adwaita	*gnome-themes-standard-locale-en-gb*	
gnupg	2.2.0	GPLv3, LGPLv3
gnupg		
gnutls	3.5.13	LGPLv2.1+
gnutls		
gobject-introspection	1.52.1	LGPLv2+, GPLv2+
gobject-introspection		
gottet	1.1.5	GPLv3
gottet	*gottet-locale-en*	
gpgme	1.9.0	GPLv2+, LGPLv2.1+
gpgme	*python3-gpg*	
grep	3.1	GPLv3
grep		
gst-player	0.0.1	LGPL-2.0+
gst-player		
gstreamer1.0	1.12.2	LGPLv2+
gstreamer1.0	*gstreamer1.0-locale-en-gb*	
gstreamer1.0-libav	1.12.2	((GPLv2+ & LGPLv2.1+) |(GPLv3+ & LGPLv3+)), GPL
gstreamer1.0-libav		
gstreamer1.0-plugins-bad	1.12.2	GPLv2+, LGPLv2+, LGPLv2.1+
gstreamer1.0-plugins-bad-accurip	*gstreamer1.0-plugins-bad-adpcmdec*	
gstreamer1.0-plugins-bad-adpcmenc	*gstreamer1.0-plugins-bad-aiff*	
gstreamer1.0-plugins-bad-asfmux	*gstreamer1.0-plugins-bad-audiobuffersplit*	
gstreamer1.0-plugins-bad-audiofxbad	*gstreamer1.0-plugins-bad-audiomixer*	
gstreamer1.0-plugins-bad-audiomixmatrix	*gstreamer1.0-plugins-bad-audiovisualizers*	
gstreamer1.0-plugins-bad-autoconvert	*gstreamer1.0-plugins-bad-bayer*	
gstreamer1.0-plugins-bad-bluez	*gstreamer1.0-plugins-bad-bz2*	
gstreamer1.0-plugins-bad-camerabin	*gstreamer1.0-plugins-bad-coloreffects*	
gstreamer1.0-plugins-bad-compositor	*gstreamer1.0-plugins-bad-curl*	
gstreamer1.0-plugins-bad-dashdemux	*gstreamer1.0-plugins-bad-debugutilsbad*	
gstreamer1.0-plugins-bad-decklink	*gstreamer1.0-plugins-bad-dtls*	
gstreamer1.0-plugins-bad-dvb	*gstreamer1.0-plugins-bad-dvbsuboverlay*	
gstreamer1.0-plugins-bad-dvdspu	*gstreamer1.0-plugins-bad-faceoverlay*	
gstreamer1.0-plugins-bad-fbdevsink	*gstreamer1.0-plugins-bad-festival*	
gstreamer1.0-plugins-bad-fieldanalysis	*gstreamer1.0-plugins-bad-freeverb*	
gstreamer1.0-plugins-bad-frei0r	*gstreamer1.0-plugins-bad-gaudieffects*	
gstreamer1.0-plugins-bad-gdp	*gstreamer1.0-plugins-bad-geometrictransform*	
gstreamer1.0-plugins-bad-hls	*gstreamer1.0-plugins-bad-id3tag*	
gstreamer1.0-plugins-bad-inter	*gstreamer1.0-plugins-bad-interlace*	
gstreamer1.0-plugins-bad-ivfparse	*gstreamer1.0-plugins-bad-ivtc*	
gstreamer1.0-plugins-bad-jp2kdecimator	*gstreamer1.0-plugins-bad-jpegformat*	
gstreamer1.0-plugins-bad-legacyrawparse	*gstreamer1.0-plugins-bad-locale-en-gb*	
gstreamer1.0-plugins-bad-meta	*gstreamer1.0-plugins-bad-midi*	
gstreamer1.0-plugins-bad-mpegpsdemux	*gstreamer1.0-plugins-bad-mpegpsmux*	

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

Recipe	Version	Licenses
gstreamer1.0-plugins-bad-mpegtsdemux	*gstreamer1.0-plugins-bad-mpegtsmux*	
gstreamer1.0-plugins-bad-mxf	*gstreamer1.0-plugins-bad-neonhttpsrc*	
gstreamer1.0-plugins-bad-netsim	*gstreamer1.0-plugins-bad-pcapparse*	
gstreamer1.0-plugins-bad-pnm	*gstreamer1.0-plugins-bad-removesilence*	
gstreamer1.0-plugins-bad-rfbsrc	*gstreamer1.0-plugins-bad-rsvg*	
gstreamer1.0-plugins-bad-rtponvif	*gstreamer1.0-plugins-bad-sbc*	
gstreamer1.0-plugins-bad-sdpelem	*gstreamer1.0-plugins-bad-segmentclip*	
gstreamer1.0-plugins-bad-shm	*gstreamer1.0-plugins-bad-siren*	
gstreamer1.0-plugins-bad-smooth	*gstreamer1.0-plugins-bad-smoothstreaming*	
gstreamer1.0-plugins-bad-sndfile	*gstreamer1.0-plugins-bad-speed*	
gstreamer1.0-plugins-bad-stereo	*gstreamer1.0-plugins-bad-subenc*	
gstreamer1.0-plugins-bad-timecode	*gstreamer1.0-plugins-bad-ttmlsubs*	
gstreamer1.0-plugins-bad-uvch264	*gstreamer1.0-plugins-bad-vcdsrc*	
gstreamer1.0-plugins-bad-videofiltersbad	*gstreamer1.0-plugins-bad-videoframe-audiolevel*	
gstreamer1.0-plugins-bad-videoparsersbad	*gstreamer1.0-plugins-bad-videosignal*	
gstreamer1.0-plugins-bad-vmnc	*gstreamer1.0-plugins-bad-vulkan*	
gstreamer1.0-plugins-bad-webp	*gstreamer1.0-plugins-bad-y4mdec*	
gstreamer1.0-plugins-bad-yadif	*libgstadaptivedemux-1.0*	
libgstbadallocators-1.0	*libgstbadaudio-1.0*	
libgstbadbase-1.0	*libgstbadvideo-1.0*	
libgstbasecamerabinsrc-1.0	*libgstcodecparsers-1.0*	
libgstinsertbin-1.0	*libgstmpegts-1.0*	
libgstphotography-1.0	*libgstplayer-1.0*	
libgsturidownloader-1.0		
gstreamer1.0-plugins-base	1.12.2	GPLv2+, LGPLv2+
gstreamer1.0-plugins-base-adder	*gstreamer1.0-plugins-base-alsa*	
gstreamer1.0-plugins-base-app	*gstreamer1.0-plugins-base-apps*	
gstreamer1.0-plugins-base-audioconvert	*gstreamer1.0-plugins-base-audiorate*	
gstreamer1.0-plugins-base-audioresample	*gstreamer1.0-plugins-base-audiotestsrc*	
gstreamer1.0-plugins-base-encoding	*gstreamer1.0-plugins-base-gio*	
gstreamer1.0-plugins-base-locale-en-gb	*gstreamer1.0-plugins-base-meta*	
gstreamer1.0-plugins-base-ogg	*gstreamer1.0-plugins-base-pango*	
gstreamer1.0-plugins-base-pbtypes	*gstreamer1.0-plugins-base-playback*	
gstreamer1.0-plugins-base-rawparse	*gstreamer1.0-plugins-base-subparse*	
gstreamer1.0-plugins-base-tcp	*gstreamer1.0-plugins-base-theora*	
gstreamer1.0-plugins-base-typefindfunctions	*gstreamer1.0-plugins-base-videoconvert*	
gstreamer1.0-plugins-base-videorate	*gstreamer1.0-plugins-base-videoscale*	
gstreamer1.0-plugins-base-videotestsrc	*gstreamer1.0-plugins-base-volume*	
gstreamer1.0-plugins-base-vorbis	*gstreamer1.0-plugins-base-ximagesink*	
gstreamer1.0-plugins-base-xvimagesink	*libgstallocators-1.0*	
libgstapp-1.0	*libgstaudio-1.0*	
libgstfft-1.0	*libgstpbutils-1.0*	
libgstriff-1.0	*libgstrtp-1.0*	
libgstrtsp-1.0	*libgstsdp-1.0*	
libgsttag-1.0	*libgstvideo-1.0*	
gstreamer1.0-plugins-good	1.12.2	GPLv2+, LGPLv2.1+
gstreamer1.0-plugins-good	*gstreamer1.0-plugins-good-alaw*	
gstreamer1.0-plugins-good-alpha	*gstreamer1.0-plugins-good-alphacolor*	
gstreamer1.0-plugins-good-apetag	*gstreamer1.0-plugins-good-audiofx*	
gstreamer1.0-plugins-good-audioparsers	*gstreamer1.0-plugins-good-auparse*	
gstreamer1.0-plugins-good-autodetect	*gstreamer1.0-plugins-good-avi*	
gstreamer1.0-plugins-good-cairo	*gstreamer1.0-plugins-good-cutter*	
gstreamer1.0-plugins-good-debug	*gstreamer1.0-plugins-good-deinterlace*	
gstreamer1.0-plugins-good-dtmf	*gstreamer1.0-plugins-good-effectv*	
gstreamer1.0-plugins-good-equalizer	*gstreamer1.0-plugins-good-flac*	
gstreamer1.0-plugins-good-flv	*gstreamer1.0-plugins-good-flxdec*	
gstreamer1.0-plugins-good-gdkpixbuf	*gstreamer1.0-plugins-good-goom*	
gstreamer1.0-plugins-good-goom2k1	*gstreamer1.0-plugins-good-icydemux*	
gstreamer1.0-plugins-good-id3demux	*gstreamer1.0-plugins-good-imagefreeze*	
gstreamer1.0-plugins-good-interleave	*gstreamer1.0-plugins-good-isomp4*	
gstreamer1.0-plugins-good-jpeg	*gstreamer1.0-plugins-good-level*	
gstreamer1.0-plugins-good-locale-en-gb	*gstreamer1.0-plugins-good-matroska*	
gstreamer1.0-plugins-good-meta	*gstreamer1.0-plugins-good-mulaw*	
gstreamer1.0-plugins-good-multifile	*gstreamer1.0-plugins-good-multipart*	
gstreamer1.0-plugins-good-navigationtest	*gstreamer1.0-plugins-good-ossaudio*	
gstreamer1.0-plugins-good-png	*gstreamer1.0-plugins-good-pulseaudio*	
gstreamer1.0-plugins-good-replaygain	*gstreamer1.0-plugins-good-rtp*	
gstreamer1.0-plugins-good-rtpmanager	*gstreamer1.0-plugins-good-rtsp*	
gstreamer1.0-plugins-good-shapewipe	*gstreamer1.0-plugins-good-smpte*	
gstreamer1.0-plugins-good-soup	*gstreamer1.0-plugins-good-spectrum*	
gstreamer1.0-plugins-good-speex	*gstreamer1.0-plugins-good-taglib*	

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

Recipe	Version	Licenses
gstreamer1.0-plugins-good-udp	*gstreamer1.0-plugins-good-video4linux2*	
gstreamer1.0-plugins-good-videobox	*gstreamer1.0-plugins-good-videocrop*	
gstreamer1.0-plugins-good-videofilter	*gstreamer1.0-plugins-good-videomixer*	
gstreamer1.0-plugins-good-wavenc	*gstreamer1.0-plugins-good-wavparse*	
gstreamer1.0-plugins-good-ximagesrc	*gstreamer1.0-plugins-good-y4menc*	
gstreamer1.0-plugins-ugly	1.12.2	GPLv2+, LGPLv2.1+, LGPLv2+
gstreamer1.0-plugins-ugly-a52dec	*gstreamer1.0-plugins-ugly-asf*	
gstreamer1.0-plugins-ugly-dvdlpcmdec	*gstreamer1.0-plugins-ugly-dvdsub*	
gstreamer1.0-plugins-ugly-lame	*gstreamer1.0-plugins-ugly-locale-en-gb*	
gstreamer1.0-plugins-ugly-meta	*gstreamer1.0-plugins-ugly-mpeg2dec*	
gstreamer1.0-plugins-ugly-mpg123	*gstreamer1.0-plugins-ugly-realmedia*	
gstreamer1.0-plugins-ugly-xingmux		
gtk+	2.24.31	LGPLv2, LGPLv2+, LGPLv2.1+
gtk+	*gtk+-locale-en-gb*	
gtk+3	3.22.17	LGPLv2, LGPLv2+, LGPLv2.1+
gtk+3	*gtk+3-locale-en*	
gtk+3-locale-en-gb		
gvfs	1.32.1	LGPLv2
gvfs	*gvfs-locale-en-gb*	
gvfsd-trash		
harfbuzz	1.4.8	MIT
harfbuzz		
hdparm	9.52	BSD
hdparm		
hicolor-icon-theme	0.15	GPLv2
hicolor-icon-theme		
hostap-utils	0.4.7	GPLv2
hostap-utils		
hostapd	2.6	GPLv2, BSD
hostapd		
hsetroot	1.0.2	GPLv2
hsetroot		
htpdate	1.1.3	GPLv2
htpdate		
hunspell	1.6.1	GPLv3, LGPLv3
hunspell		
i2c-tools	3.1.2	GPLv2+
i2c-tools		
i2c-write-read	1.0.6	GPLv2
i2c-write-read		
icu	59.1	ICU
libicudata	*libicui18n*	
libicuuc		
imlib2	1.4.6	MIT, BSD
imlib2	*imlib2-loaders*	
imx-alsa-plugins	1.0.26	GPLv2
imx-alsa-plugins		
imx-gpu-viv	5.0.11.p8.6-hfp	Proprietary
libegl-mx6	*libgal-mx6*	
libgl-mx6	*libgles-mx6*	
libgles2-mx6	*libglslc-mx6*	
libopenvg-mx6	*libvivante-dri-mx6*	
libvsc-mx6		
init-ifupdown	1.0	GPLv2
init-ifupdown		
initscripts	1.0	GPLv2
initscripts	*initscripts-functions*	
iperf3	3.2	BSD
iperf3		
iptables	1.6.1	GPLv2+
iptables		

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

Recipe	Version	Licenses
iw *iw*	4.9	BSD
kbd *kbd-keymaps*	2.0.4	GPLv2+
kernel-module-imx-gpu-viv *kernel-module-galcore-4.1.15-yocto-standard*	5.0.11.p8.6 *kernel-module-imx-gpu-viv*	GPLv2
kexec-tools *kexec*	2.0.14	GPLv2
kidletime *kidletime*	5.41.0	GPLv2, LGPLv2.1
kmod *kmod*	24 *libkmod*	GPL-2.0+, LGPL-2.1+
kwayland *kwayland*	5.41.0	LGPLv2.1
kwindowsystem *kwindowsystem*	5.41.0 *kwindowsystem-locale-en-gb*	GPLv2, LGPLv2.1
l3afpad *l3afpad*	0.8.18.1.11	GPLv2+
lame *libmp3lame*	3.99.5	LGPLv2+
liba52 *liba52*	0.7.4	GPLv2+
libarchive *libarchive*	3.3.2	BSD
libassuan *libassuan*	2.4.3	LGPLv2.1+
libatasmart *libatasmart*	0.19	LGPLv2.1
libcap *libcap*	2.25	BSD, GPLv2
libcomps *libcomps*	0.1.8	GPLv2
libconfig *libconfig*	1.5	LGPLv2.1
libcroco *libcroco*	0.6.12	LGPLv2, LGPLv2.1
libdaemon *libdaemon*	0.14	LGPLv2.1+
libdbusmenu-qt5 *libdbusmenu-qt5*	0.9.3	GPLv2
libdmx *libdmx*	1.1.3	MIT
libdnf *libdnf*	0.9.3	LGPLv2.1
libdrm *libdrm*	2.4.83	MIT
libepoxy *libepoxy*	1.4.3	MIT
liberation-fonts *liberation-fonts*	2.00.1	OFL-1.1
libevdev *libevdev*	1.5.7	MIT-X
libevent *libevent*	2.1.8	BSD, MIT
libexif *libexif*	0.6.21 *libexif-locale-en-gb*	LGPLv2.1
libffi *libffi*	3.2.1	MIT

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

Recipe	Version	Licenses
libfm *libfm*	1.2.5 *libfm-locale-en-gb*	GPLv2+, LGPLv2+
libfm-extra *libfm-extra*	1.2.5	LGPLv2+
libfm-qt *libfm-qt*	0.12.0	LGPLv2.1
libfontenc *libfontenc*	1.1.3	MIT
libgcc *libgcc*	7.3.0	GPL-3.0-with-GCC-exception
libgcrypt *libgcrypt*	1.8.0	LGPLv2.1+
libgpg-error *libgpg-error*	1.27	GPLv2+, LGPLv2.1+
libgphoto2 *libgphoto2* *libgphotoport*	2.5.14 *libgphoto2-camlibs*	LGPLv2.1
libgudev *libgudev*	231	LGPLv2.1
libical *libical*	2.0.0	LGPLv2.1, MPL-1
libice *libice*	1.0.9	MIT-style
libidn *libidn*	1.33	LGPLv2.1+, LGPLv3
libinput *libinput*	1.8.1	MIT
libjpeg-turbo *libjpeg-turbo*	1.5.2	BSD-3-Clause
libksba *libksba*	1.3.5	GPLv2+, LGPLv3+, GPLv3+
libkscreen *libkscreen*	5.11.5	GPLv2
liblxqt *liblxqt*	0.12.0	LGPLv2.1
libmscboost *libmscboost*	4.1.0	LGPLv2.1
libmscboostpython *libmscboostpython*	1.0.1	LGPLv2.1
libnfsidmap *libnfsidmap*	0.25	BSD
libnl *libnl*	3.2.29 *libnl-genl*	LGPLv2.1
libnotify *libnotify*	0.7.7	LGPLv2.1
libnss-mdns *libnss-mdns*	0.10	LGPLv2.1+
libogg *libogg*	1.3.2	BSD
libpcap *libpcap*	1.8.1	BSD
libpciaccess *libpciaccess*	0.13.5	MIT, MIT-style
libpcre *libpcre*	8.41	BSD
libpng *libpng*	1.6.31	Libpng
libproxy	0.4.14	LGPLv2.1+

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

Recipe	Version	Licenses
libproxy		
libqtxdg *libqtxdg*	3.1.0	LGPLv2.1
librepo *librepo*	1.7.20	LGPLv2.1
librsvg *librsvg*	2.40.18	LGPLv2+
libsamplerate0 *libsamplerate0*	0.1.9	BSD-2-Clause
libsdl *libsdl*	1.2.15	LGPLv2.1
libsecret *libsecret*	0.18.5	LGPLv2.1
libsm *libsm*	1.2.2	MIT-style
libsndfile1 *libsndfile1*	1.0.28	LGPLv2.1
libsolv *libsolv*	0.6.28 *libsolvext*	BSD-3-Clause
libsoup-2.4 *libsoup-2.4*	2.58.2 *libsoup-2.4-locale-en-gb*	LGPLv2
libstatgrab *libstatgrab*	0.91	MIT
libsysstat *libsysstat*	0.4.0	LGPLv2.1
libtheora *libtheora*	1.1.1	BSD
libtirpc *libtirpc*	1.0.2	BSD
libtool *libltdl*	2.4.6	GPLv2, LGPLv2.1
libunistring *libunistring*	0.9.7	LGPLv3+, GPLv2
libusb1 *libusb1*	1.0.21	LGPLv2.1+
libva *libva*	1.8.3 *libva-x11*	MIT
libva-utils *libva-utils*	1.8.3	MIT
libvorbis *libvorbis*	1.3.5	BSD
libwebp *libwebp*	0.6.0	BSD
libx11 *libx11* *libx11-xcb*	1.6.5 *libx11-locale*	MIT, MIT-style, BSD
libxau *libxau*	1.0.8	MIT-style
libxaw *libxaw7*	1.0.13	MIT-X
libxcb *libxcb* *libxcb-render* *libxcb-shm* *libxcb-xfixes* *libxcb-xkb*	1.12 *libxcb-randr* *libxcb-shape* *libxcb-sync* *libxcb-xinerama*	MIT
libxcomposite *libxcomposite*	0.4.4	MIT-style
libxcursor	1.1.14	MIT-style

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

Recipe	Version	Licenses
libxcursor		
libxdamage *libxdamage*	1.1.4	MIT
libxdmcp *libxdmcp*	1.1.2	MIT-style
libxext *libxext*	1.3.3	MIT-style
libxfce4ui *libxfce4ui* *libxfce4ui-gtk3*	4.12.1 *libxfce4ui-gtk2* *libxfce4ui-locale-en-gb*	GPLv2
libxfce4util *libxfce4util*	4.12.1 *libxfce4util-locale-en-gb*	GPLv2
libxfixes *libxfixes*	5.0.3	MIT-style
libxfont2 *libxfont2*	2.0.1	MIT, MIT-style, BSD
libxft *libxft*	2.3.2	MIT
libxi *libxi*	1.7.9	MIT, MIT-style
libxinerama *libxinerama*	1.1.3	MIT
libxkbcommon *libxkbcommon*	0.7.1	MIT, MIT-style
libxkbfile *libxkbfile*	1.0.9	MIT-style
libxml2 *libxml2*	2.9.4	MIT
libxmu *libxmu*	1.1.2 *libxmuu*	MIT, MIT-style
libxpm *libxpm*	3.5.12	BSD
libxrandr *libxrandr*	1.5.1	MIT-style
libxrender *libxrender*	0.9.10	MIT-style
libxscrnsaver *libxscrnsaver*	1.2.2	MIT
libxslt *libxslt*	1.1.29	MIT
libxt *libxt*	1.1.5	MIT, MIT-style
libxtst *libxtst*	1.2.3	MIT-style
libxv *libxv*	1.0.11	MIT-style
libxxf86dga *libxxf86dga*	1.1.4	MIT-X
libxxf86misc *libxxf86misc*	1.0.3	MIT
libxxf86vm *libxxf86vm*	1.1.4	MIT
linux-firmware *linux-firmware-ath6k* *linux-firmware-imx-sdma-imx6q*	0.0 *linux-firmware-atheros-license* *linux-firmware-imx-sdma-license*	Firmware-atheros_firmware
linux-tools-testusb *linux-tools-testusb*	1.0	GPL-2.0
linux-yocto-custom	4.1.15	GPLv2

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

Recipe	Version	Licenses
kernel-base	*kernel-devicetree*	
kernel-image	*kernel-image-zimage*	
kernel-module-at24-4.1.15-yocto-standard	*kernel-module-at25-4.1.15-yocto-standard*	
kernel-module-cdc-acm-4.1.15-yocto-standard	*kernel-module-crc-itu-t-4.1.15-yocto-standard*	
kernel-module-crc7-4.1.15-yocto-standard	*kernel-module-dummy-hcd-4.1.15-yocto-standard*	
kernel-module-ftdi-sio-4.1.15-yocto-standard	*kernel-module-g-acm-ms-4.1.15-yocto-standard*	
kernel-module-g-cdc-4.1.15-yocto-standard	*kernel-module-g-ether-4.1.15-yocto-standard*	
kernel-module-g-hid-4.1.15-yocto-standard	*kernel-module-g-mass-storage-4.1.15-yocto-standard*	
kernel-module-g-multi-4.1.15-yocto-standard	*kernel-module-g-ncm-4.1.15-yocto-standard*	
kernel-module-g-serial-4.1.15-yocto-standard	*kernel-module-g-zero-4.1.15-yocto-standard*	
kernel-module-gadgetfs-4.1.15-yocto-standard	*kernel-module-gspca-main-4.1.15-yocto-standard*	
kernel-module-i2c-algo-pca-4.1.15-yocto-standard	*kernel-module-i2c-algo-pcf-4.1.15-yocto-standard*	
kernel-module-ipu-bg-overlay-sdc-4.1.15-yocto-standard	*kernel-module-ipu-csi-enc-4.1.15-yocto-standard*	
kernel-module-ipu-fg-overlay-sdc-4.1.15-yocto-standard	*kernel-module-ipu-prp-enc-4.1.15-yocto-standard*	
kernel-module-ipu-still-4.1.15-yocto-standard	*kernel-module-isofs-4.1.15-yocto-standard*	
kernel-module-libcomposite-4.1.15-yocto-standard	*kernel-module-libcrc32c-4.1.15-yocto-standard*	
kernel-module-msdos-4.1.15-yocto-standard	*kernel-module-mx6s-capture-4.1.15-yocto-standard*	
kernel-module-mxc-v4l2-capture-4.1.15-yocto-standard	*kernel-module-net2280-4.1.15-yocto-standard*	
kernel-module-nls-iso8859-15-4.1.15-yocto-standard	*kernel-module-ov5640-camera-4.1.15-yocto-standard*	
kernel-module-ov5640-camera-int-4.1.15-yocto-standard	*kernel-module-ov5640-camera-mipi-4.1.15-yocto-standard*	
kernel-module-ov5640-camera-mipi-int-4.1.15-yocto-standard	*kernel-module-ov5642-camera-4.1.15-yocto-standard*	
kernel-module-tw9910-4.1.15-yocto-standard	*kernel-module-u-ether-4.1.15-yocto-standard*	
kernel-module-u-serial-4.1.15-yocto-standard	*kernel-module-udf-4.1.15-yocto-standard*	
kernel-module-usb-f-acm-4.1.15-yocto-standard	*kernel-module-usb-f-ecm-4.1.15-yocto-standard*	
kernel-module-usb-f-ecm-subset-4.1.15-yocto-standard	*kernel-module-usb-f-hid-4.1.15-yocto-standard*	
kernel-module-usb-f-mass-storage-4.1.15-yocto-standard	*kernel-module-usb-f-ncm-4.1.15-yocto-standard*	
kernel-module-usb-f-obex-4.1.15-yocto-standard	*kernel-module-usb-f-rndis-4.1.15-yocto-standard*	
kernel-module-usb-f-serial-4.1.15-yocto-standard	*kernel-module-usb-f-ss-lb-4.1.15-yocto-standard*	
kernel-module-usbserial-4.1.15-yocto-standard	*kernel-module-usbtest-4.1.15-yocto-standard*	
kernel-module-uvcvideo-4.1.15-yocto-standard	*kernel-module-v4l2-int-device-4.1.15-yocto-standard*	
kernel-module-videobuf2-vmalloc-4.1.15-yocto-standard	*kernel-modules*	
lmsensors *lmsensors-libsensors*	3.4.0	GPLv2+, LGPLv2.1+
lmsensors-config *lmsensors-config-libsensors*	1.0	MIT-X
lximage-qt *lximage-qt*	0.6.0	GPLv2
lxmenu-data *lxmenu-data*	0.1.5	LGPLv2.1
lxqt-about *lxqt-about*	0.12.0	LGPLv2.1
lxqt-admin *lxqt-admin*	0.12.0	LGPLv2.1
lxqt-config *lxqt-config*	0.12.0	LGPLv2.1
lxqt-globalkeys *lxqt-globalkeys*	0.12.0	LGPLv2.1
lxqt-l10n *lxqt-l10n*	0.12.0	LGPLv2.1
lxqt-notificationd *lxqt-notificationd*	0.12.0	LGPLv2.1
lxqt-openssh-askpass *lxqt-openssh-askpass*	0.12.0	LGPLv2.1
lxqt-panel *lxqt-panel*	0.12.0	LGPLv2.1
lxqt-policykit *lxqt-policykit*	0.12.0	LGPLv2.1
lxqt-powermanagement *lxqt-powermanagement*	0.12.0	LGPLv2.1
lxqt-qtplugin *lxqt-qtplugin*	0.12.0	LGPLv2.1
lxqt-runner *lxqt-runner*	0.12.0	LGPLv2.1
lxqt-session	0.12.0	LGPLv2.1

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

Recipe	Version	Licenses
lxqt-session		
lxqt-sudo *lxqt-sudo*	0.12.0	LGPLv2.1
lxqt-themes *lxqt-themes*	0.12.0	LGPLv2.1
lxqt-world *lxqt-world*	1.0	MIT
lzo *lzo*	2.10	GPLv2+
mem-edit *mem-edit*	1.5.0	GPLv2
menu-cache *menu-cache*	1.0.2	LGPLv2.1+
mesa *libgbm* *mesa-megadriver*	17.1.7 *libglapi*	MIT
mesa-demos *mesa-demos*	8.3.0	MIT, PD
mini-x-session *mini-x-session*	0.1	GPLv2
minicom *minicom*	2.7.1	GPLv2+
mmc-utils *mmc-utils*	0.1	GPLv2
mobile-broadband-provider-info *mobile-broadband-provider-info*	20170310	PD
modutils-initscripts *modutils-initscripts*	1.0	PD
mozjs *libmozjs*	17.0.0	MPL-2.0
mpeg2dec *libmpeg2*	0.5.1	GPLv2+
mpg123 *mpg123*	1.25.6	LGPLv2.1
msc-bug-report *msc-bug-report*	1.0	GPLv2
msc-completion *msc-completion*	1.0	MIT
msc-cpufreq *msc-cpufreq*	1.0	GPLv2
msc-init-script-early *msc-init-script-early*	1.0	MIT
msc-init-script-late *msc-init-script-late*	1.0	MIT
msc-ldk-benchmark *msc-ldk-benchmark*	0.4.0	MSCv1
msc-ldk-exhibition-safe *msc-ldk-exhibition-safe*	0.2.1	GPLv2
msc-ldk-verification *msc-ldk-verification*	3.0.1	MSCv1
msc-ldk-verification-apps *msc-ldk-verification-apps*	0.5.0	GPLv2
msc-linux-scripts *msc-linux-scripts*	2.0.0	MSCv1
msc-lxqt-config *msc-lxqt-config*	1.0	MIT
msc-screensaver *msc-screensaver*	1.0.0	MIT

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

Recipe	Version	Licenses
msc-wallpaper *msc-wallpaper*	1.0.0	GPLv2
mscio-cmd *mscio-cmd*	1.5.0	LGPLv2.1
mscio-drivers *kernel-module-devreg-4.1.15-yocto-standard* *kernel-module-eapi-ec-bl-4.1.15-yocto-standard* *kernel-module-eapi-ec-running-time-4.1.15-yocto-standard* *kernel-module-user-gpios-4.1.15-yocto-standard*	4.0.0 *kernel-module-eapi-ec-4.1.15-yocto-standard* *kernel-module-eapi-ec-hwm-4.1.15-yocto-standard* *kernel-module-eapi-ec-wdt-4.1.15-yocto-standard* *mscio-drivers*	GPLv2
mscio-lib *mscio-lib*	3.4.0	LGPLv2.1
mscio-monitor *mscio-monitor*	2.3.0	GPLv2
mtd-utils *mtd-utils*	2.0.0	GPLv2+
mtdev *mtdev*	1.1.5	MIT
muparser *muparser*	2.2.5	MIT
nano *nano*	2.2.5	GPLv2
ncurses *ncurses-libformw* *ncurses-libncurses* *ncurses-libpanel* *ncurses-libtinfo* *ncurses-terminfo-base*	6.0 *ncurses-libmenuw* *ncurses-libncursesw* *ncurses-libpanelw* *ncurses-terminfo*	MIT
neard *neard*	0.16	GPLv2
neon *neon*	0.30.2	LGPLv2+
netbase *netbase*	5.4	GPLv2
netpipe *netpipe*	3.7.2	GPLv1
nettle *nettle*	3.3	LGPLv3+, GPLv2+
nfs-utils *nfs-utils-client*	2.1.1 *nfs-utils-mount*	MIT, GPLv2+, BSD
npth *npth*	1.5	LGPLv2+
nspr *nspr*	4.16	GPL-2.0, MPL-2.0, LGPL-2.1
nss *nss*	3.31.1	(MPL-2.0 & LGPL-2.1+), (MPL-2.0 & GPL-2.0+), MPL-2
ntp *ntp* *ntpdate*	4.2.8p10 *ntp-tickadj*	NTP
obconf-qt *obconf-qt*	0.12.0	LGPLv2.1
ofono *ofono*	1.20	GPLv2
openbox *openbox* *openbox-core* *openbox-theme-bear2* *openbox-theme-clearlooks-3.4* *openbox-theme-mikachu* *openbox-theme-onyx* *openbox-theme-orang*	3.6.1 *openbox-config* *openbox-theme-artwiz-boxed* *openbox-theme-clearlooks* *openbox-theme-clearlooks-olive* *openbox-theme-natura* *openbox-theme-onyx-citrus* *openbox-theme-syscrash*	GPLv2+

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

Recipe	Version	Licenses
openssl *libcrypto* *openssl*	1.0.2m *libssl* *openssl-conf*	openssl
opkg-utils *update-alternatives-opkg*	0.3.5	GPLv2+
orc *liborc-0.4*	0.4.27	BSD-2-Clause, BSD-3-Clause
otter-browser *otter-browser*	0.9.12-beta12	GPLv3
oxygen-icons *oxygen-icons*	15.04.3	LGPLv3
packagegroup-base *packagegroup-base* *packagegroup-base-alsa* *packagegroup-base-extended* *packagegroup-base-nfc* *packagegroup-base-pci* *packagegroup-base-usbhost* *packagegroup-base-zeroconf* *packagegroup-machine-base*	1.0 *packagegroup-base-3g* *packagegroup-base-bluetooth* *packagegroup-base-ipv6* *packagegroup-base-nfs* *packagegroup-base-usbgadget* *packagegroup-base-wifi* *packagegroup-distro-base*	MIT
packagegroup-core-boot *packagegroup-core-boot*	1.0	MIT
packagegroup-core-ssh-dropbear *packagegroup-core-ssh-dropbear*	1.0	MIT
packagegroup-core-x11 *packagegroup-core-x11-utils*	1.0	MIT
packagegroup-core-x11-base *packagegroup-core-x11-base*	1.0	MIT
packagegroup-core-x11-xserver *packagegroup-core-x11-xserver*	1.0	MIT
packagegroup-lxqt-base *packagegroup-lxqt-base*	1.0	MIT
packagegroup-msc-ldk-core *packagegroup-msc-ldk-core*	1.0	MSCv1
packagegroup-msc-lxqt *packagegroup-msc-lxqt* *packagegroup-msc-lxqt-base*	1.0 *packagegroup-msc-lxqt-apps* *packagegroup-msc-lxqt-games*	MIT
packagegroup-mscio *packagegroup-mscio*	1.0	GPLv2, MSCv1
pango *pango*	1.40.6	LGPLv2.0+
parole *parole*	0.8.1 *parole-locale-en-gb*	GPLv2
parted *parted*	3.2	GPLv3+
pavucontrol-qt *pavucontrol-qt*	0.3.0	GPLv2
pci2uio *kernel-module-pci2uio-4.1.15-yocto-standard*	2.0.1 *pci2uio*	GPLv2
pciutils *libpci* *pciutils-ids*	3.5.5 *pciutils*	GPLv2+
pcmanfm-qt *pcmanfm-qt*	0.12.0	GPLv2
perl *perl*	5.24.1 *perl-lib*	Artistic-1.0, GPL-1.0+
pinentry *pinentry*	1.0.0	GPLv2
pixman *pixman*	0.34.0	MIT, MIT-style, PD

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

Recipe	Version	Licenses
pm-utils	1.4.1	GPLv2
pm-utils		
pointercal	0.0	MIT
pointercal		
pointercal-xinput	0.0	MIT
pointercal-xinput		
polkit	0.113	LGPLv2+
polkit		
polkit-group-rule-datetime	1.0	MIT
polkit-group-rule-datetime		
polkit-qt-1	0.112.0	LGPLv2.1
polkit-qt-1		
popt	1.16	MIT
popt		
powertop	2.8	GPLv2
powertop	*powertop-locale-en-gb*	
powertop-locale-en-us		
proxy-config	1.0	MIT
proxy-config		
psplash	0.1	GPLv2+
psplash	*psplash-default*	
pulseaudio	10.0	LGPLv2.1+, MIT, BSD-3-Clause
libpulse	*libpulse-mainloop-glib*	
libpulse-simple	*libpulsecommon*	
libpulsecore	*pulseaudio-lib-alsa-util*	
pulseaudio-lib-protocol-native	*pulseaudio-module-alsa-card*	
pulseaudio-module-alsa-sink	*pulseaudio-module-alsa-source*	
pulseaudio-module-always-sink	*pulseaudio-module-augment-properties*	
pulseaudio-module-card-restore	*pulseaudio-module-console-kit*	
pulseaudio-module-default-device-restore	*pulseaudio-module-detect*	
pulseaudio-module-device-manager	*pulseaudio-module-device-restore*	
pulseaudio-module-filter-apply	*pulseaudio-module-filter-heuristics*	
pulseaudio-module-intended-roles	*pulseaudio-module-native-protocol-unix*	
pulseaudio-module-null-sink	*pulseaudio-module-position-event-sounds*	
pulseaudio-module-rescue-streams	*pulseaudio-module-role-cork*	
pulseaudio-module-stream-restore	*pulseaudio-module-suspend-on-idle*	
pulseaudio-module-switch-on-port-available	*pulseaudio-module-udev-detect*	
pulseaudio-module-x11-cork-request	*pulseaudio-module-x11-publish*	
pulseaudio-module-x11-xsmp	*pulseaudio-server*	
python	2.7.13	PSFv2
libpython2	*python-codecs*	
python-contextlib	*python-core*	
python-crypt	*python-datetime*	
python-io	*python-lang*	
python-logging	*python-math*	
python-mime	*python-netclient*	
python-pickle	*python-re*	
python-readline	*python-stringold*	
python-textutils		
python-imaging	1.1.7	MIT
python-imaging		
python3	3.5.3	PSFv2
libpython3	*python3-2to3*	
python3-argparse	*python3-asyncio*	
python3-audio	*python3-codecs*	
python3-compile	*python3-compression*	
python3-core	*python3-crypt*	
python3-ctypes	*python3-curses*	
python3-datetime	*python3-db*	
python3-debugger	*python3-difflib*	
python3-distutils	*python3-doctest*	
python3-email	*python3-enum*	
python3-fcntl	*python3-gdbm*	
python3-html	*python3-idle*	
python3-image	*python3-importlib*	
python3-io	*python3-json*	

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

Recipe	Version	Licenses
python3-lang	*python3-logging*	
python3-mailbox	*python3-math*	
python3-mime	*python3-misc*	
python3-mmap	*python3-modules*	
python3-multiprocessing	*python3-netclient*	
python3-netserver	*python3-numbers*	
python3-pickle	*python3-pkgutil*	
python3-pprint	*python3-profile*	
python3-pydoc	*python3-re*	
python3-readline	*python3-reprlib*	
python3-resource	*python3-selectors*	
python3-shell	*python3-signal*	
python3-smtpd	*python3-sqlite3*	
python3-sqlite3-tests	*python3-stringold*	
python3-subprocess	*python3-syslog*	
python3-terminal	*python3-textutils*	
python3-threading	*python3-tkinter*	
python3-typing	*python3-unittest*	
python3-unixadmin	*python3-xml*	
python3-xmlrpc		
python3-dbus *python3-dbus*	1.2.4	MIT
python3-ewmh *python3-ewmh*	0.1.5	LGPL-3.0
python3-iniparse *python3-iniparse*	0.4	MIT, PSF
python3-pycairo *python3-pycairo*	1.10.0	LGPLv3
python3-pygobject *python3-pygobject*	3.24.1	LGPLv2.1
python3-setuptools *python3-setuptools*	36.2.7	MIT
python3-six *python3-six*	1.10.0	MIT
python3-xlib *python3-xlib*	0.17	LGPL-2.1
qtbase *qtbase*	5.9.4 *qtbase-plugins*	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
qtdeclarative *qtdeclarative* *qtdeclarative-qmlplugins*	5.9.4 *qtdeclarative-plugins*	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
qterminal *qterminal*	0.8.0	GPLv2
qtermwidget *qtermwidget*	0.8.0	GPLv2
qtimageformats *qtimageformats-plugins*	5.9.4	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
qtlocation *qtlocation* *qtlocation-qmlplugins*	5.9.4 *qtlocation-plugins*	(LGPL-2.1 & The-Qt-Company-Qt-LGPL-Exception-1.1
qtmultimedia *qtmultimedia* *qtmultimedia-qmlplugins*	5.9.4 *qtmultimedia-plugins*	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
qtsensors *qtsensors* *qtsensors-qmlplugins*	5.9.4 *qtsensors-plugins*	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
qtsvg *qtsvg*	5.9.4 *qtsvg-plugins*	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
qtwebkit *qtwebkit*	5.9.4 *qtwebkit-qmlplugins*	BSD, LGPLv2+, GPL-2.0
qtx11extras *qtx11extras*	5.9.4	(LGPL-2.1 & The-Qt-Company-Qt-LGPL-Exception-1.1

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

Recipe	Version	Licenses
qtxmlpatterns *qtxmlpatterns*	5.9.4	(GPL-2.0+ |LGPL-3.0 |The-Qt-Company-Commercial),
readline *readline*	7.0	GPLv3+
rgb *rgb*	1.0.6	MIT-X
rpcbind *rpcbind*	0.2.4	BSD
rpm *python3-rpm*	4.13.90 *rpm*	GPL-2.0
run-postinsts *run-postinsts*	1.0	MIT
sbc *sbc*	1.3	LGPLv2.1+
shadow *shadow*	4.2.1 *shadow-base*	BSD, Artistic-1.0
shadow-securetty *shadow-securetty*	4.2.1	MIT
shared-mime-info *shared-mime-info*	1.8 *shared-mime-info-data*	LGPLv2+
smartmontools *smartmontools-ctl*	6.3	GPLv2
solid *solid*	5.41.0 *solid-locale-en-gb*	LGPLv2.1
speex *speex*	1.2.0	BSD
speexdsp *speexdsp*	1.2rc3	BSD
spi-register *spi-register*	1.0.8	GPLv2
sqlite3 *libsqlite3*	3.20.0	PD
strace *strace*	4.18	BSD
stress *stress*	1.0.4	GPLv2
sudo *sudo*	1.8.20p2	ISC, BSD, Zlib
sysfsutils *libsysfs*	2.1.0 *sysfsutils*	LGPLv2.1
system-test-controller-config *system-test-controller-config*	1.0	MSCv1
system-test-controller-v2 *system-test-controller-v2*	2.22.1	GPLv2
sysvinit *sysvinit*	2.88dsf *sysvinit-pidof*	GPLv2+
sysvinit-inittab *sysvinit-inittab*	2.88dsf	GPLv2
taglib *taglib*	1.11.1 *taglib-c*	LGPLv2.1, MPL-1
tcp-wrappers *libwrap*	7.6	BSD
tcpdump *tcpdump*	4.7.4	BSD
tiff *tiff*	4.0.8	BSD-2-Clause
tiny-shell *tiny-shell*	0.2.4	GPLv2

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

Recipe	Version	Licenses
tpm-tools *tpm-tools*	1.3.9.1	CPL-1.0
tpm2-abrmd *tpm2-abrmd*	1.2.0	BSD-2-Clause
tpm2-tools *tpm2-tools*	3.0.3	BSD
tpm2-tss *libtctidevice* *libtss2*	1.3.0 *libtctisocket*	BSD-2-Clause
trace-cmd *trace-cmd*	2.6.1	GPLv2, LGPLv2.1
trousers *libtspi*	0.3.14 *trousers*	BSD
tslib *tslib*	1.1 *tslib-conf*	LGPLv2
tzdata *tzdata*	2018c *tzdata-europe*	PD, BSD, BSD-3-Clause
u-boot-denx *u-boot-denx*	1.0	GPLv2
udev-extraconf *udev-extraconf*	1.1	MIT
udev-rules-imx *udev-rules-imx*	1.0	MIT
udisks2 *udisks2* *udisks2-locale-en-gb*	2.1.8 *udisks2-libs*	GPLv2+, LGPLv2+
update-flash *update-flash*	0.2.1	GPLv2
update-rc.d *update-rc.d*	0.7	GPLv2+
usbutils *usbutils*	008	GPLv2+
useradd-msc *useradd-msc*	1.0	MIT
util-linux *util-linux-blkid* *util-linux-fstrim* *util-linux-libmount* *util-linux-libuuid* *util-linux-mcookie* *util-linux-sulogin* *util-linux-umount*	2.30 *util-linux-dmesg* *util-linux-libblkid* *util-linux-libsmartcols* *util-linux-lsblk* *util-linux-mount* *util-linux-swaponoff*	GPLv2+, LGPLv2.1+, BSD
vulkan *vulkan*	1.0.51.0	Apache-2.0
wayland *wayland*	1.13.0	MIT
wireless-tools *wireless-tools*	30.pre9	(LGPLv2.1 |MPL-1.1 |BSD), GPLv2
wpa-supplicant *wpa-supplicant* *wpa-supplicant-passphrase*	2.6 *wpa-supplicant-cli*	BSD
xauth *xauth*	1.0.10	MIT-X
xcb-util *xcb-util*	0.4.0	MIT
xcb-util-image *xcb-util-image*	0.4.0	MIT
xcb-util-keysyms *xcb-util-keysyms*	0.4.0	MIT
xcb-util-renderutil	0.3.9	MIT

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

Recipe	Version	Licenses
xcb-util-renderutil		
xcb-util-wm *xcb-util-wm*	0.4.1	MIT
xdg-user-dirs *xdg-user-dirs*	0.15	GPLv2
xdpyinfo *xdpyinfo*	1.3.2	MIT-X
xf86-input-evdev *xf86-input-evdev*	2.10.5	MIT-X
xf86-input-libinput *xf86-input-libinput*	0.25.1	MIT-X
xf86-video-imxfb-vivante *xf86-video-imxfb-vivante*	5.0.11.p8.6	MIT-X
xfconf *xfconf*	4.12.1 *xfconf-locale-en-gb*	GPLv2
xhost *xhost*	1.0.7	MIT-X
xinit *xinit*	1.3.4	MIT-X
xinput *xinput*	1.6.2	MIT-X
xinput-calibrator *xinput-calibrator*	0.7.5	MIT-X
xkbcomp *xkbcomp*	1.4.0	MIT-X
xkeyboard-config *xkeyboard-config*	2.21 *xkeyboard-config-locale-en-gb*	MIT, MIT-style
xmodmap *xmodmap*	1.0.9	MIT
xrandr *xrandr*	1.5.0	MIT
xserver-nodm-init *xserver-nodm-init*	3.0	GPLv2
xserver-xf86-config *xserver-xf86-config*	0.1	MIT-X
xserver-xorg *xserver-xorg* *xserver-xorg-module-exa*	1.19.3 *xserver-xorg-extension-glx*	MIT-X
xset *xset*	1.2.3	MIT
xssstate *xssstate*	1.1	MIT
xterm *xterm*	330	MIT-X
xvinfo *xvinfo*	1.1.3	MIT-X
xz *liblzma*	5.2.3	PD
zip *zip*	3.0	BSD-3-Clause
zlib *zlib*	1.2.11	Zlib
zsh *zsh*	5.3.1	zsh
zsh-config *zsh-config*	1.0	MSCv1

C. Bootloader Configuration

C.1. Booting

It is possible to install the Linux images on a µSD-Card and to boot from.

Advantages are:

- The µSD 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 µSD-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 appropiate .sdcard-file from:
`msc-ldk/build/0584/tmp/work/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
```

After inserting the µSD card into the modules µSD Slot and powering the system it will boot:

```
This is an example booting an i.MX6Dual (BoardMark: IMX6D-123_014)

U-Boot SPL 2015.07-0584_nanoRISC_iMX6_develop+g4b22770 (Jan 31 2017 - 17:14:19)
Boardinfo:
  name ......... msc-nR-imx6
  variant ...... 123
  feature ...... 014
  serial ....... 1234567899
  revision ..... N/A
Booting from MMC1


U-Boot 2015.07-0584_nanoRISC_iMX6_develop+g4b22770 (Jan 31 2017 - 17:14:19 +0100)

CPU:   Freescale i.MX6D rev1.5 at 792MHz
CPU:   Industrial temperature grade (-40C to 105C) at 46C
Reset cause: POR
Board: MSC nanoRISC i.MX6
I2C:   ready
DRAM:  2 GiB
LCD, HDMI,
MMC:   FSL_SDHC: 0, FSL_SDHC: 1, FSL_SDHC: 2
*** Warning - bad CRC, using default environment

Display: hdmi:1024x768M@60 (1024x768)
In:    serial
Out:   serial
Err:   serial
SF: Detected AT25SF041 with page size 256 Bytes, erase size 4 KiB, total 512 KiB
Net:   FEC [PRIME]
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... 1 USB Device(s) found
USB1:   USB EHCI 1.00
scanning bus 1 for devices... 1 USB Device(s) found
       scanning usb for storage devices... 0 Storage Device(s) found

USB device 0: unknown device
ERR: USB start failed
Attempting mmc boot...
switch to partitions #0, OK
mmc0 is current device
Loading environment (uEnv.txt) from MMC0 ...
reading uEnv.txt
10 bytes read in 10 ms (1000 Bytes/s)
Importing environment (uEnv.txt) ...
Loading linux image (boot/zImage) from MMC0 ...
5482800 bytes read in 443 ms (11.8 MiB/s)
Booting from MMC0 ...
Loading FDT image (boot/msc-nR-imx6-123-014-headless.dtb) from MMC0 ...
46607 bytes read in 652 ms (69.3 KiB/s)
Kernel image @ 0x12000000 [ 0x000000 - 0x53a930 ]
## Flattened Device Tree blob at 18000000
   Booting using the fdt blob at 0x18000000
   Using Device Tree in place at 18000000, end 1800e60e

Starting kernel ...
```

C.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 ...... 143
  feature ...... 014
  serial ....... 1003065655
  revision ..... N/A
```

C.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-imx6-143-014-800x480-lcd.dtb
  => saveenv
  Saving Environment to MMC...
  Writing to MMC(0)... done
  =>
```

C.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 .sdcard image.
- generate U-Boot separated from yocto, update µSD 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 acessible 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 loctated
- # 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

C.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
```

C.6. Change Boot order

The universal Bootlodader 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 emm
```

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 dertermine the mmc to boot from. Hence here is also another option to change the boot device. On default, the µMMC-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.

D. 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.

D.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

D.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 loacated in `/sys/class/boarddata`:

```
root@msc-nr-imx6:/sys/class/boarddata# ls -1
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
```

E. Devicetree Configuration

E.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 hierachically. The .dts (Device Tree Source) files are the topmost ones. They include the .dtsi (Device Tree Source Include) files, which can include another .dtsi.

E.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 sheme 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-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:

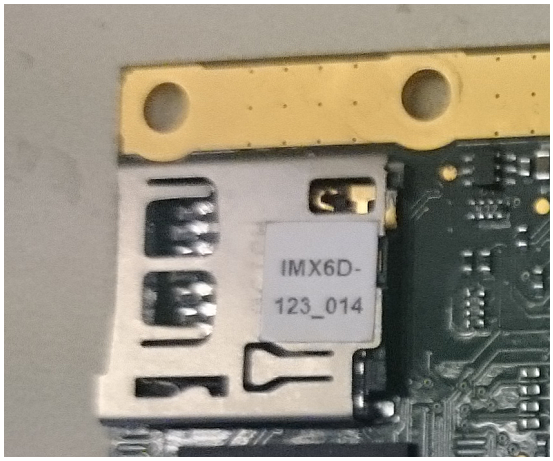


Figure E.1. – example BoardMark

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		bitwise
	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-imx6-143-014-800x480-lcd.dts:**

```
#include "msc-nR-imx6-143-014.dtsi"
#include "msc-nR-imx6-mb2-maxtouch.dtsi"

&mxcfb1 {
        compatible = "fsl,mxc_sdc_fb";
        disp_dev = "lcd";
        interface_pix_fmt = "RGB24";
        mode_str ="AMPIRE-WVGA";
        default_bpp = <32>;
        int_clk = <0>;
        late_init = <0>;
        status = "okay";
};

&lcd_panel {
        status = "okay";
};

&pwm1 {
        status = "okay";
};

&lcd_bl {
        status = "okay";
};
```

In the first lines two other dtsi files are included:
`msc-nR-imx6-143-014.dtsi` and `msc-nR-imx6-mb2-maxtouch.dtsi`,
in the file itself the framebuffer is set up for 24bit LCD. Panel power, PWM and Backlight are enabled. The include of `msc-nR-imx6-mb2-maxtouch.dtsi` enables the touchscreen.
The include of `msc-nR-imx6-143-014.dtsi` will setup the other stuff, which will be explained below, it is the common part for all `msc-nR-imx6-143-014-*` files.

Taking a look to the head of the file **msc-nR-imx6-143-014.dtsi:**

```
#include "msc-nR-imx6q.dtsi"
```

and the first lines of **msc-nR-imx6q.dtsi**:

```
#include "imx6q-pinfunc.h"
#include "imx6q.dtsi"
#include "msc-nR-imx6qd.dtsi"
```

The file `imx6q.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 furthermore the headlines of **msc-nR-imx6qd.dtsi**:

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

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

```
#include "skeleton.dtsi"
#include <dt-bindings/gpio/gpio.h>
#include "imx6qdl.dtsi"
```

The file `imx6qdl.dtsi` is a generic file included in all i.MX6 designs - also in Freescale reference board sabresd - and may not be modified.

E.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. To use this layer it has to be added in the setup phase of BSP:

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 --add-layer ssh://gitolite@msc-git02.msc-ge.com
    :9418/msc_ol99/meta-msc-arm-extensions.git
```

Afterwards there is a subdirectory `addons/meta-msc-arm-extensions.git` created. This is the location where setup has checked out the layer.

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/addons/meta-msc-arm-extension$ 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's designed as base to be copied/expanded by your needs. Copying it, you have to add the name in
`.../meta-msc-arm-extensions.git/recipes-kernel/device-tree/linux-yocto-custom_%.bbappend`

There are two display interfaces as example mentioned, the third one page before, but commented out to be inactive. You may activate one ore more of them, the example uses different framebuffers for each interface to ensure simultaneous usage. Here is the base version of this file:

```
/*
 * Copyright (C) 2017 MSC Technologies, Design Center Aachen
 *
 * This program is free software; you can redistribute it and/or modify
 * it under the terms of the GNU General Public License as published by
 * the Free Software Foundation; either version 2 of the License, or
 * (at your option) any later version.

 * This program is distributed in the hope that it will be useful,
 * but WITHOUT ANY WARRANTY; without even the implied warranty of
 * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
 * GNU General Public License for more details.

 * You should have received a copy of the GNU General Public License along
 * with this program; if not, write to the Free Software Foundation, Inc.,
 * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
 */
/dts-v1/;

/*****************************************************************************
 *                                                                           *
 * Example Devicetree                                                        *
 *                                                                           *
 * please select the dtsi file according to your board.                      *
 *                                                                           *
 *****************************************************************************/
/{};
/* enable by changing #if 0 to #if 1 */
#if 0
#include "msc-<modul>-<variant>.dtsi"
#endif

/* example HDMI Display setting using fb2: */
/* activate by changing #if 0 to #if 1 */

#if 0

&mxcfb2 {
        status = "okay";
};

&hdmi_display {
        status = "okay";
};

&hdmi_core {
        status = "okay";
};

&hdmi_video {
        status = "okay";
};

#endif

/* example LVDS Display setting using fb1: */
/* activate by changing #if 0 to #if 1 */

#if 0

&mxcfb1 {
        interface_pix_fmt = "RGB666";
```

```
63          status = "okay";
64  };
65
66  &lvds0_timings {
67          clock-frequency = <65000000>;
68          hactive = <1024>;
69          vactive = <768>;
70          hback-porch = <220>;
71          hfront-porch = <40>;
72          vback-porch = <21>;
73          vfront-porch = <7>;
74          hsync-len = <60>;
75          vsync-len = <10>;
76  };
77
78  &ldb {
79          status = "okay";
80
81          lvds-channel@0 {
82                  status = "okay";
83          };
84  };
85
86  #endif
87
88  /* add your setting below */
```

You can see an example base devicetree include in line 28, 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
```

Additional settings can be added below the last line, which is highlited in green in the above listing. Some information can be found in chapter E. 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 µSD-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
```

F. 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:

F.1. QML_BillboardDemo

This Demo shows an animated Text on a LED screen:

Figure F.1. – Billboard Demo

F.2. Qt5_CinematicExperienceDemo

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

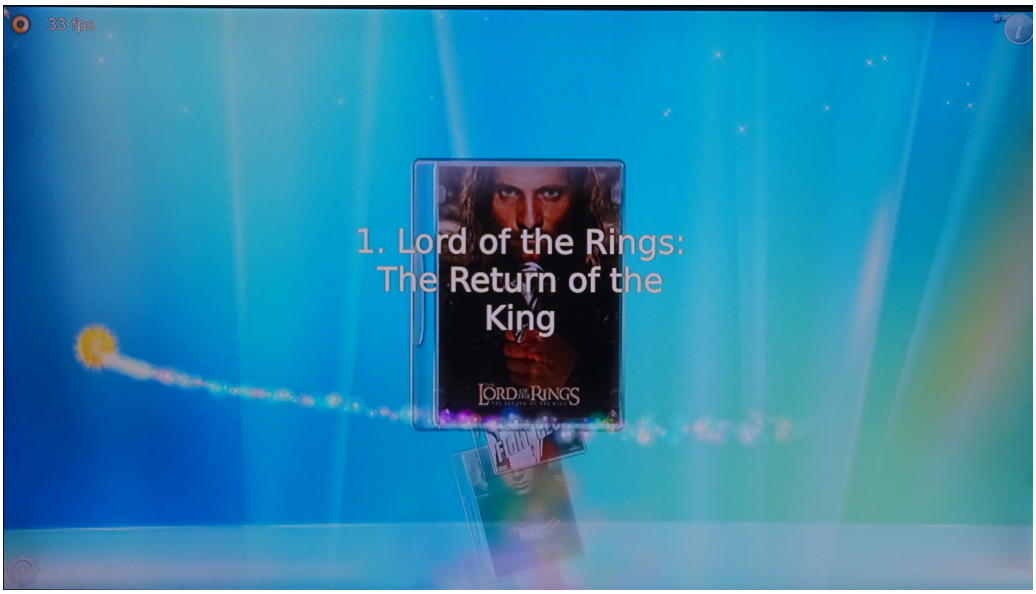


Figure F.2. – Cinematic Experience Demo

F.3. Qt5_EverywhereDemo

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

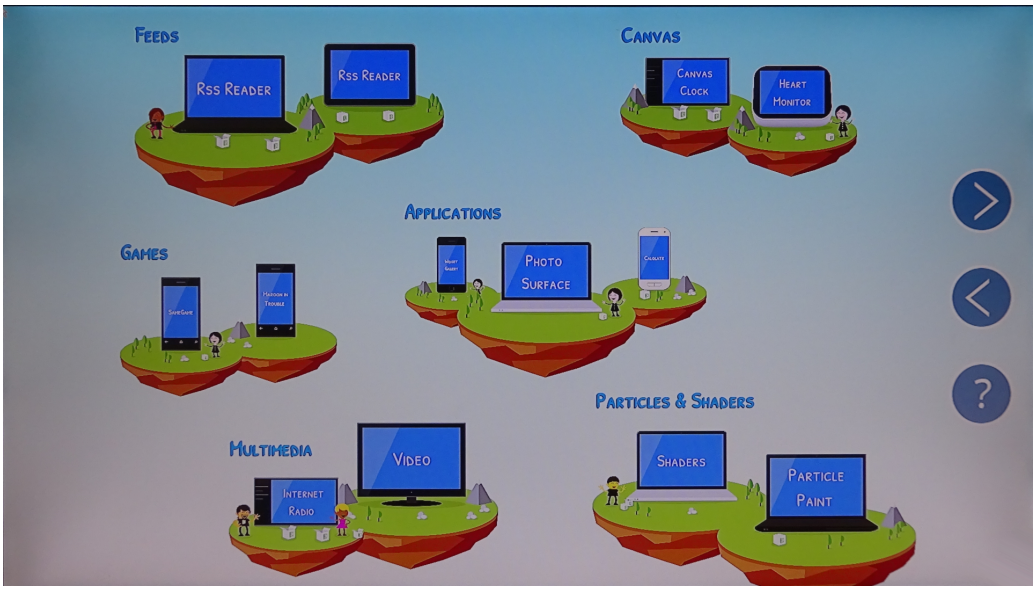


Figure F.3. – Everywhere Demo

F.4. Qt5_NMapperDemo

This Demo shows how elements can be manipulated and mapped.

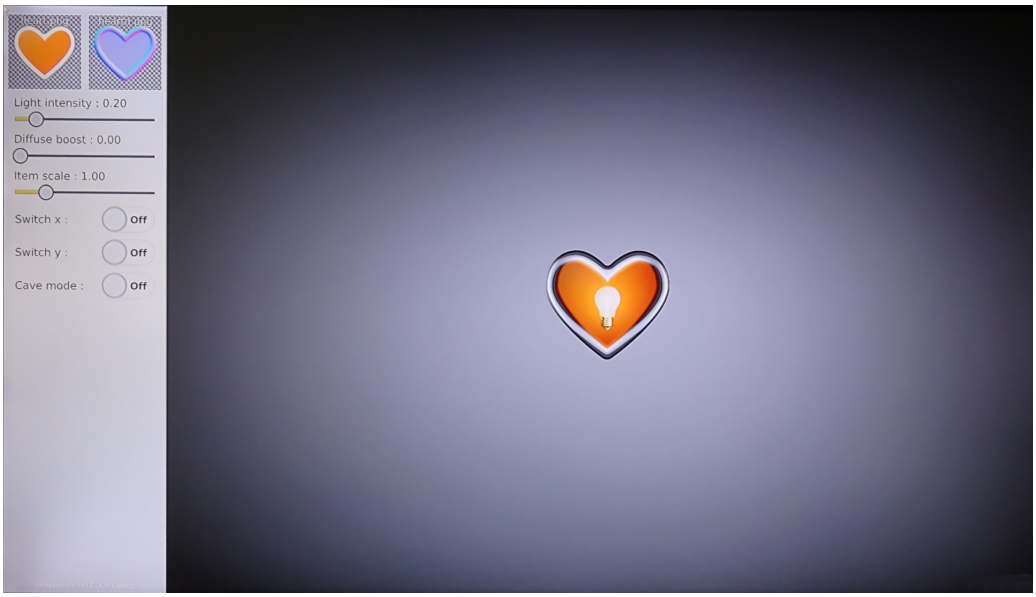


Figure F.4. – NMapper Demo

F.5. Qt5_NmapcarouseDemo

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

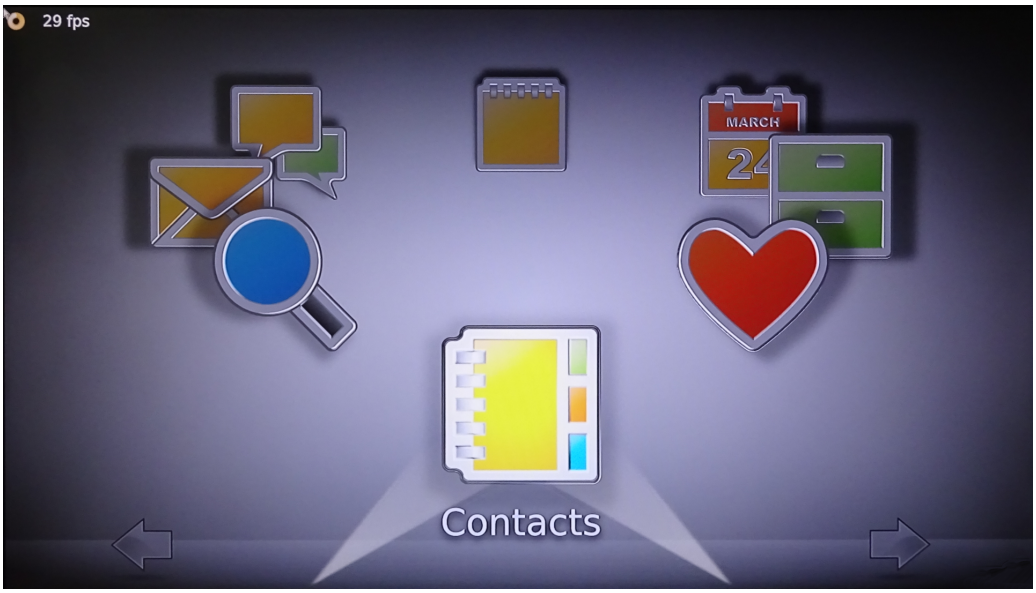


Figure F.5. – Nmapcarouse Demo

F.6. Qt5_QuitBatteryDemo

This Demo shows an usage of the QT5 Battery Component.

Figure F.6. – QuitBattery Demo

F.7. Qt5_SmartHomeDemo

This Demo shows a Smart Home control screen.

Figure F.7. – SmartHome Demo

G. Supported Periphery

Table G.1. – Supported periphery

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

G.1. eMMC

eMMC is available as MMC device /dev/mmcblk2

G.2. GPIOs

GPIOs are available. Add the `Base GPIO#` 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 G.2. – GPIOs

Base GPIO#	gpio prefix	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].

G.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`.

G.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
```

G.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
```

G.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
```

G.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
```

G.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
}
```

G.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 G.3. Depending on your changes beeing stored, the example setting may not work anymore.

G.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>`

G.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).

G.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
```

G.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
```

G.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
```

G.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
```

G.6. UART

The kernel reserves 3 serial port devices (`/dev/ttymxc0` ... `/dev/ttymxc2`) even when they are not connected.

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

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

A login shell is created on `/dev/ttymxc0` running with 115200 baud, 8 databits, no parity.

G.7. SPI

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

G.8. Graphic Output

G.8.1. Display Support

The BSP supports output on multiple displays. There are devicetrees for seeral 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.

- To disable a display on frambuffer `<nr>`, run:

  ```
  echo 1 > /sys/class/graphics/fb<nr>/blank
  ```

  e.g. to disable the LVDS display, run:

  ```
  echo 1 > /sys/class/graphics/fb<nr>/blank
  ```

- To enable a display, run:

  ```
  echo 0 > /sys/class/graphics/fb<nr>/blank
  ```

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

```
echo 3 >/sys/devices/soc0/lcd_backlight.25/backlight/lcd_backlight.25/brightness
```

H. Troubleshooting

Table H.1. – Errors

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

I. Tips And Tricks

I.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`.

I.1.1. Using The google-chrome Browser

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

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

user@devhost:msc-ldk$ ./setup.py --bsp=0584 \
   --local-conf-append=chrome.append \
   --add-layer=https://github.com/OSSystems/meta-browser
```

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

```
user@devhost:msc-ldk$ ./setup.py --bsp=0584 \
   --add-layer=https://github.com/OSSystems/meta-browser \
   --add-layer=ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/meta-chrome-instead-of-otter
```

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

I.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.

I.2.1. htpdate

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

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

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

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

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

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

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

To update the RTC, do:

```
hwclock --systohtc
```

J. Security

J.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.

J.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
- **Canyry stack protection** : https://lwn.net/Articles/584225/
- **Position Independent Executable** : https://securityblog.redhat.com/2012/11/28/position-independent-executables-pie/
- **Memory Protection Extensions (MPX)** : https://software.intel.com/sites/default/files/managed/9d/f6/Intel_MPX_EnablingGuide.pdf

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

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

KCA : Kernel settings

LA : Undesired recipe licenses

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

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

J.3. Setting Up A Project

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

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

K. Power Analysis

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

The most import CPU core states are:

Table K.1. – CPU core states

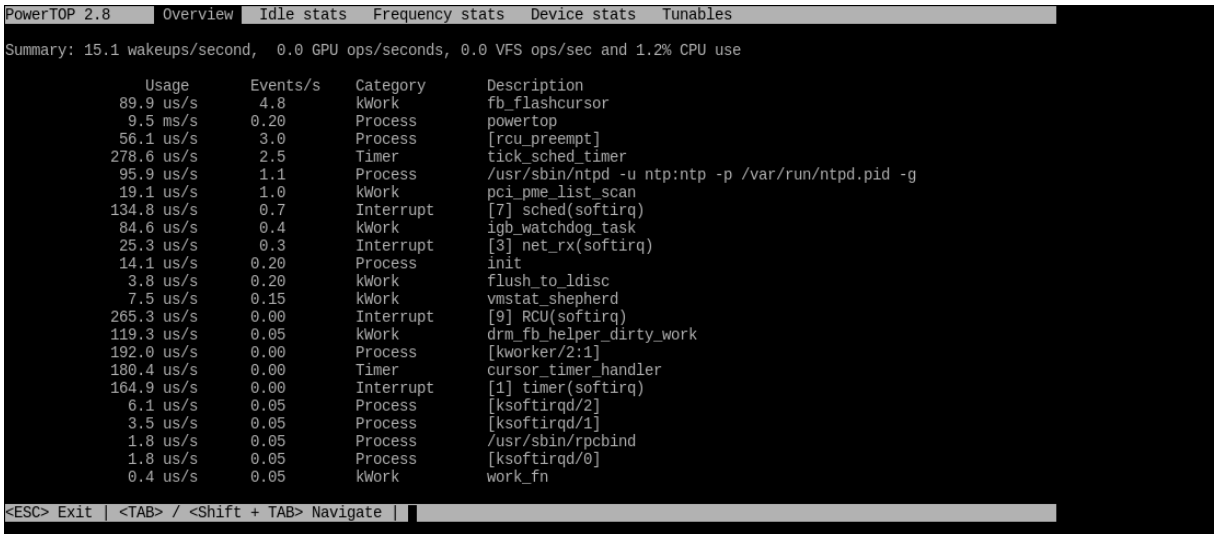
Mode	Description
C0	CPU fully turned on
C1	The main CPU clocks used for executing instructions are stopped, all others are still on
C3	All internal CPU clocks are stopped.
C6	All internal CPU clocks are stopped and internal voltage is reduced.

Depending on the CPU more modes might be available.

K.1. powertop

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

Figure K.1. – powertop overview



L. Microsoft Azure

L.1. What Is Microsoft Azure?

Microsoft Azure is a growing collection of integrated cloud services that developers and IT professionals use to build, deploy and manage applications through a global network of data centres. Featuring over 50 compliance offerings Microsoft Azure is the most trusted cloud offering for US Government institutions and is robust and secure for any commercial implication.

Various MSC CPU modules have been certified under Linux and Windows for Microsoft Azure.

L.2. How Does Our Solution Work?

The **A**vnet **A**zure **o**n **b**oarding **t**ool (AOOT) makes use of a number of attributes of the Microsoft Azure IoT Hub to provide customers with a secure means to measure, monitor and interact dynamically with their target hardware. The Avnet Azure on boarding tool give customers the opportunity to experience plug and play connectivity and has been designed to be scalable and customizable Platform as a Service that can be built upon to allow customers to create their own bespoke look, feel and functionality within their own secure single tenant Avnet Azure account.

L.3. How To Setup The Board For Azure

L.3.1. Proxy Setting

If the device is behind a proxy, the proxy settings in `/etc/profile.d/proxy.sh` must be adapted to the network in order for the device being able to communicate with the internet.

```
export no_proxy=localhost,127.0.0.0/8,::1
export https_proxy=proxy.avnet.eu:3128
export http_proxy=proxy.avnet.eu:3128
export ftp_proxy=proxy.avnet.eu:3128
export socks_server=proxy.avnet.eu:3128
```

File L.1 – Example /etc/profile.d/proxy.sh

These settings take effect only after a login, a reboot or parsing them manually with:

```
. /etc/profile.d/proxy.sh
```

When using the `otter-browser`, the proxy must be changed as described in Proxy Configuration For The otter-browser, too. This is not necessary for the google-chrome.

L.4. How To Use The AAOT Web Frontent

L.4.1. Use Conditions & Costs

There are no costs to use Easy IoT. Easy IoT is subject to fair use policies. Each user is allowed to register up to 3 boards, and 2 users per company are allowed. 3 months from the date of the first connection to the AAOT web front end the board will be disconnected. To extend your trial period please contact `mailto://support@msc-technologies.eu`

L.4.2. How To Get Access To The Portal

To gain access to the portal your device needs to be registered on the portal. Please email mailto://support@msc-technologies.eu with the unique 10 digit serial number for each board you wish to register. **The serial number for the module is visible in the BIOS setup under the main menu and is also printed on a label on the module (also with bar code)**. Please note this is named "Board Id" in the management dashboard.

L.4.3. How To Build Your Own Application

Easy IoT has been designed so that customers who have the knowledge can build upon the device level software to create their own functionality and reporting algorithms. This is also true of the dashboard. Please contact mailto://support@msc-technologies.eu if you wish to do this.

Alternatively Avnet can offer training and/or a bespoke solution service encompassing device and cloud development. Please contact us for more details on our offerings.

L.4.4. Overview

Avnet provides a web frontend for the AAOT for monitoring devices as a multi-tenant cloud solution. This solution is meant for getting familiar with Azure.

The features of the cloud solution are:

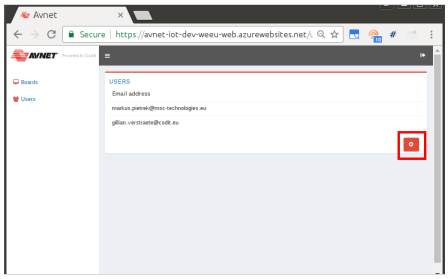
1. Display health graphs over time of the devices of the customer
2. Monitoring device health and sending alerts when thresholds are reached.
3. Sending text messages to the device.
4. Customers have their own partition with separate user management.

L.4.5. User Management

Using the AAOT requires a Microsoft Live login. If not already done an account must be created at Microsoft Live.

Two additional users can be registered that can manage the boards and view the device data:

Figure L.1. – AAOT - User List



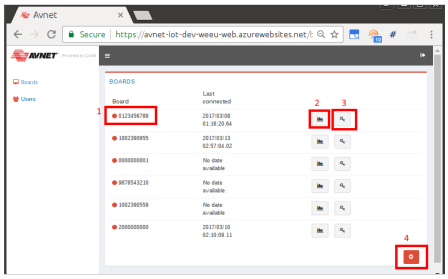
Press the "+" button on the bottom right window and enter the email address of a registered Microsoft Live account.

L.4.6. Device Management

To view the list of registered boards press "Boards" in the left menu:

1. The devices board ID is displayed with the last time it was connected.
2. The health page with device details can be opened.
3. The device key for the board can be viewed, necessary in case the software on the device has been reinstalled.
4. New devices can be added.

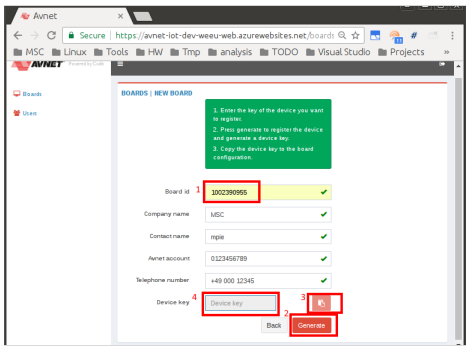
Figure L.2. – AAOT - User List



When adding a new device:

1. The board ID / serial number of the device must be entered besides the account information.
2. A device key will be generated and displayed in (4)
3. The key can be copied to the clipboard.

Figure L.3. – AAOT - Add New Device

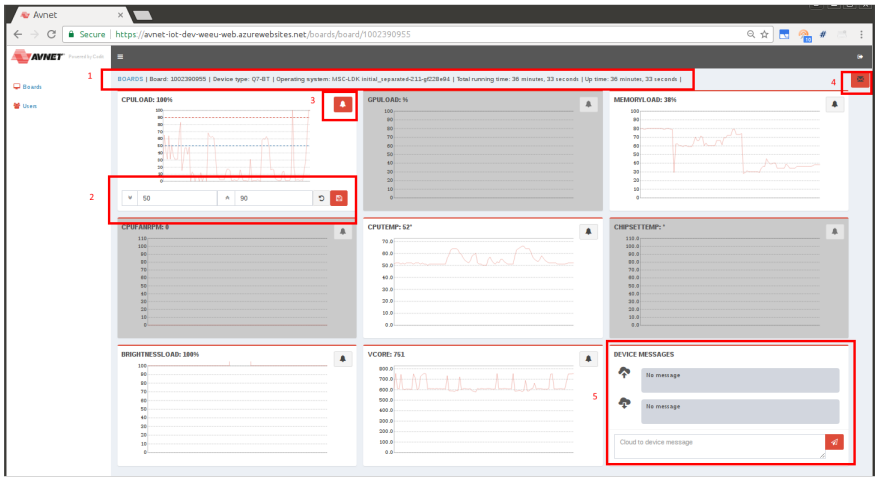


L.4.7. Displaying Health Data

The health page displays board information and various sensor values. The availability of displayed sensor types depends on the actual hardware.

1. Some board information with the operating system running when it was last connected and the total running time as well as the up time is displayed.
2. Thresholds for email alerts can be set. An email will be sent to the users when the measured value is below or above the thresholds. This analysis is done in the cloud and not the device.
3. A sensor must be individually enabled for alerts.
4. Sending email alerts can be disabled.
5. Two way communication to the device can be performed. (See How To Send Messages and How To Receive Messages

Figure L.4. – AAOT - Health Charts



L.5. Demo Application DemoAAOT

L.5.1. Overview

The image `msc-image-lxqt` includes the application `DemoAAOT` for demonstrating the use of the AAOT and extracting health data on the device. It makes use of Microsofts `azure-iot-sdk`.

Features of the demo application are:

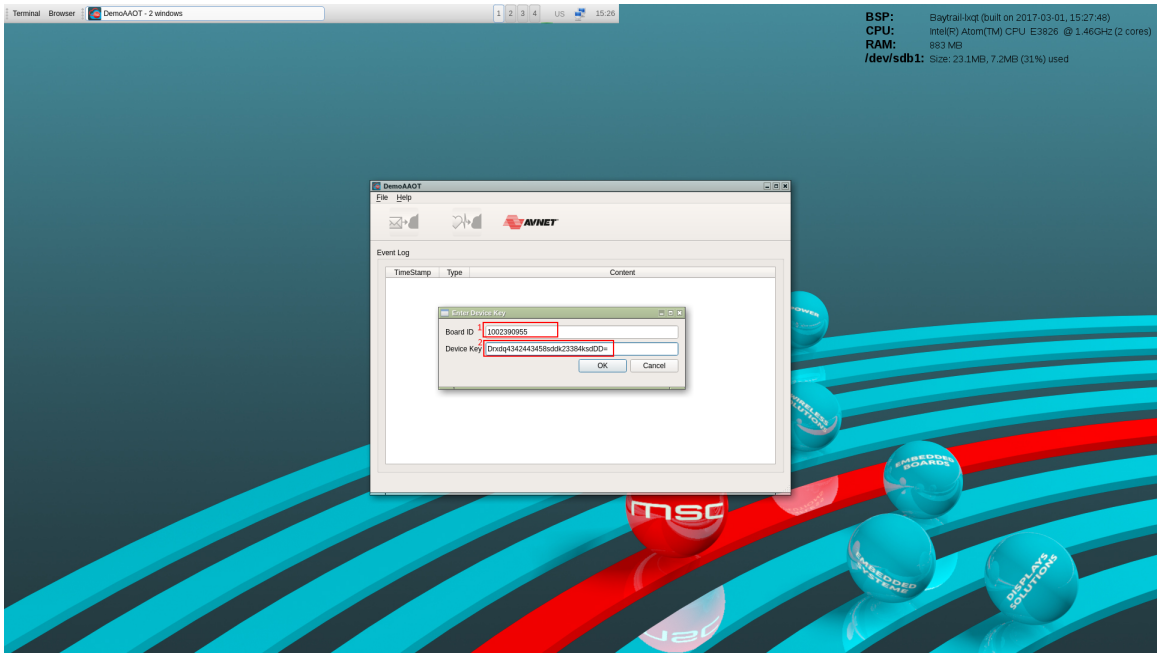
1. Connecting to the AAOT cloud services and sending board identification once per connection
2. Sending health data like CPU temperature, FAN speed, CPU load and other values periodically to the cloud.
3. Sending text messages to the cloud. These messages could be self-defined alert or status messages.
4. Displaying incoming messages from the cloud.

L.5.2. Startup

The application can be launched by the shell command `DemoAAOT` or via the desktop's right mouse button context menu entry `Demos/Azure DemoAAOT`. On first launch it will query the unique Azure device key.

1. The devices board ID / serial number is displayed in the device key dialog
2. The board must be registered on the AAOT. The device key can then be copied & pasted into the dialog. If the software was re-installed on the same hardware the existing device key should be used.

Figure L.5. – DemoAAOT - Startup



After entering the device key the device connects to the AAOT and sends an identification telegram consisting of board ID/serial number, board type (e.g. Q7-BT oder C6C-BT) as well as the operating system.

The device key is only stored on proper application exit.

L.5.3. Health Data

The application will send health data every minute. Only values supported by the board and the operating system will be included in the telegram.

Figure L.6. – DemoAAOT - Health Data sent

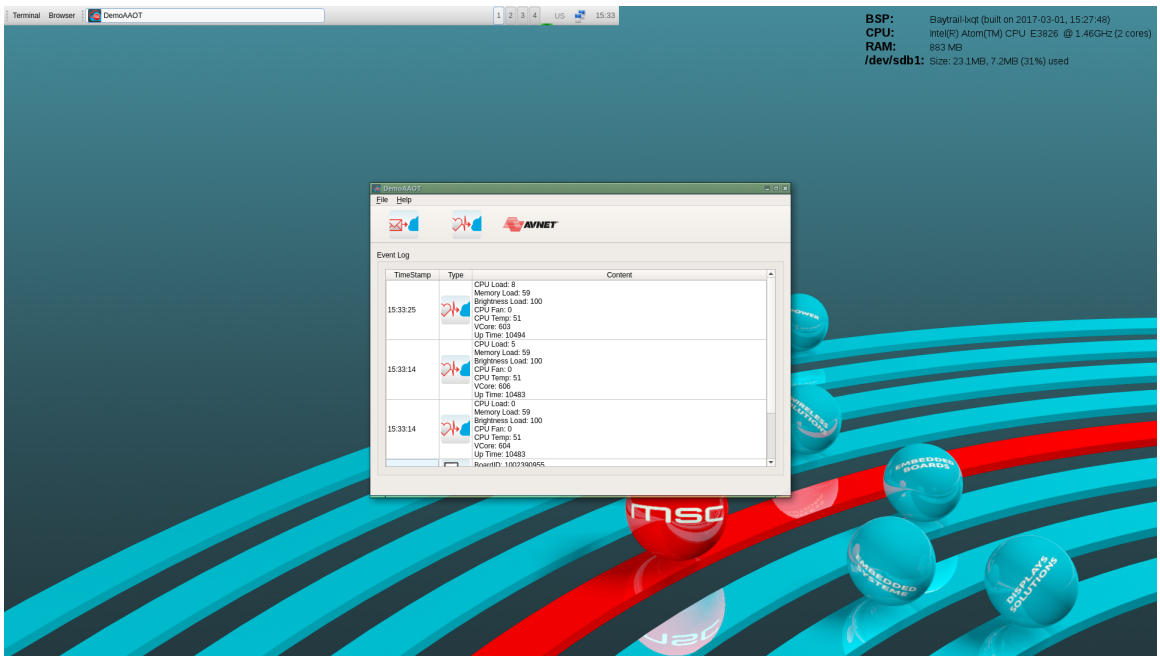


Table L.1. – DemoAAOT's Health Values

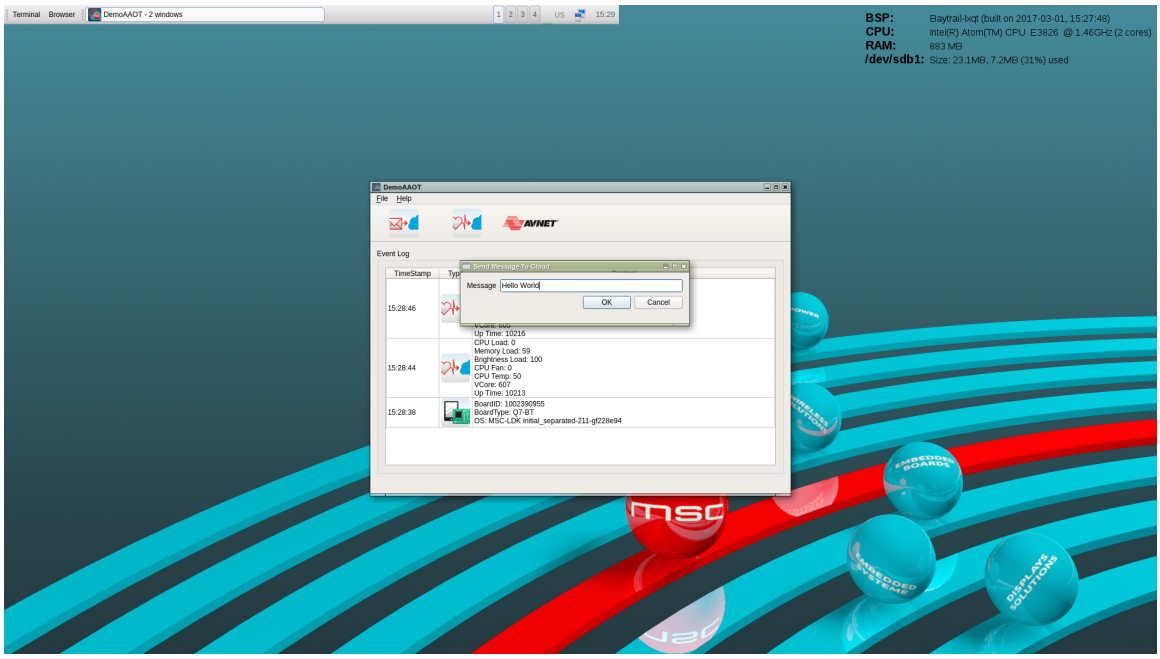
Entry	Unit	Description
CPU Load	Percentage	The average CPU load since the last health telegram. The first one will be 0.
GPU Load	Percentage	The average CPU load since the last health telegram. The first one will be 0.
Memory Load	Percentage	The used or non-available memory.
CPU Fan	Rotations per minute	The rotation speed of the FAN attached to the CPU module. When it is 0 the FAN is either not connected or stuck.
CPU Temperature	Celsius	The temperature measured at the CPU.
Chipset Temperature	Celsius	The temperature measured at the CPU's companion chipset.
Brightness	Percentage	The brightness of the LVDS display. This value might be displayed even when there is no display attached because the system can't detect this condition.
VCore	Voltage	The CPU's core voltage. The typical value depends on actual CPU.
Total	Seconds	Total seconds the system has been powered on.
UpTime	Seconds	Seconds the operating system has been running since last reset.

A health telegram will be also sent when the toolbar button is pressed.

L.5.4. How To Send Messages

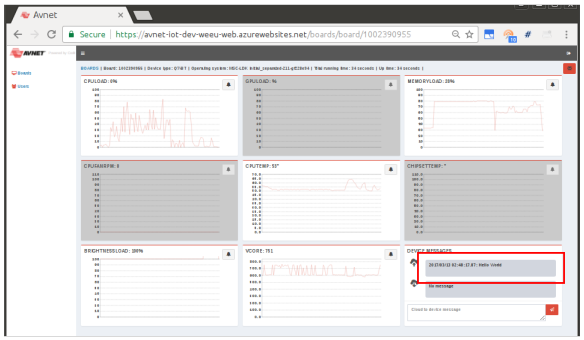
Press the button to open the send message dialog.

Figure L.7. – DemoAAOT - Send Message Dialog



Then enter the message which is sent to the AAOT immediately.

Figure L.8. – Dashboard - Message received



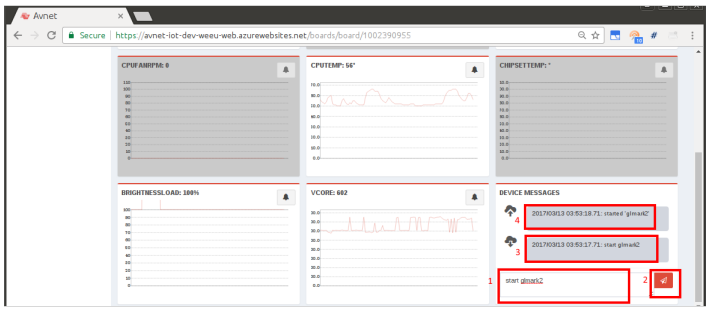
L.5.5. How To Receive Messages

Messages can be entered on the AAOT and will be put in a queue. When the device connects, e.g. the DemoAAOT application is started, it will retrieve the messages from the queue. Messages older than a configurable time (typically one day) will be dropped from the queue.

To send a message from the AAOT to the device scroll to the bottom right chart:

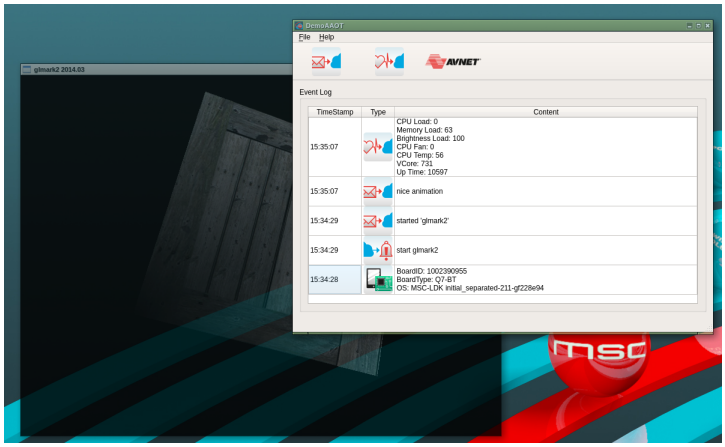
1. Enter the text. Take care not to add invisible newlines.
2. Press send.
3. The sent message is then displayed in the device and the dashboard.
4. Some text messages like “help” or “start glmark2” are interpreted on the device and trigger a response.

Figure L.9. – Dashboard - Message received



On the DemoAAOT side the received message will then be displayed in the event log of the application.

Figure L.10. – DemoAAOT - Notification Received



L.5.6. How To Open AAOT In Browser

Press the button AVNET to open the AAOT homepage in a browser.

L.5.7. Configuration

On first launch the default configuration will be written to `~/.config/MSC/DemoAAOT.conf`

```
[General]
auto_health_timeout_seconds=60
browser_cmd=/usr/bin/google-chrome %1
dashboard_url=http://avnet-iot-dev-weeu-web.azurewebsites.net/
hostname=avnet-iot-dev-tele-all.azure-devices.net
device_key="DrxdQ1KlFCUG19lqkRCZuoE5IqDLnqaIc83t3BcasssaX="
shared_key="7JNGqGlHJ4b8WQPUTPsq2YbhbKq2qAvRfFAPLTWXas224="
simulate_connection=false
simulate_health=false
```

File L.2 – ~/.config/MSC/DemoAAOT.conf

The file can then be edited via `l3afpad ~/.config/MSC/DemoAAOT.conf`

Table L.2. – DemoAAOT's Customizable Settings

Setting	Description
`auto_health_timeout_seconds`	The timeout until the next health telegram is being sent.

L.6. Source Code

Retrieve the source with `git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/demoaaot.g` `DemoAAOT.git`.

The most important files are:

Table L.3. – DemoAAOT's Most Import Files

File	Description
`CMakeLists.txt`	Demonstrates how to include and link to the `azure-iot-sdk` from Microsoft.
`src/AzureConnection.cpp`	Manages the connection to AAOT and sending telegrams.
`src/EapiOs.cpp`	Retrieves sensor values from the hardware.
`src/ModelDeclaration.h`	Defines the telegrams exchanged with AAOT.

L.7. How To Build Your Own Azure Applications

The yocto recipe for your application must contain the line:

```
DEPENDS += " azure-iot-sdks "
```

Then the `azure-iot-sdk` headers can be included from `/usr/include/azureiot`.

Detailed descriptions for `azure-iot-sdk` can be found here:

1. Azure IoT device SDK for C
2. Azure IoT Hub developer guide
3. Azure IoT device SDK for C - more about serializer

M. Links

- MSC Technologies http://www.msc-technologies.eu
- Yocto project https://www.yoctoproject.org
- Yocto project documentation
  http://www.yoctoproject.org/docs/2.4/mega-manual/mega-manual.html
- LXQt http://lxqt.org
- Otter Browser https://otter-browser.org
- Microsoft Azure https://azure.microsoft.com