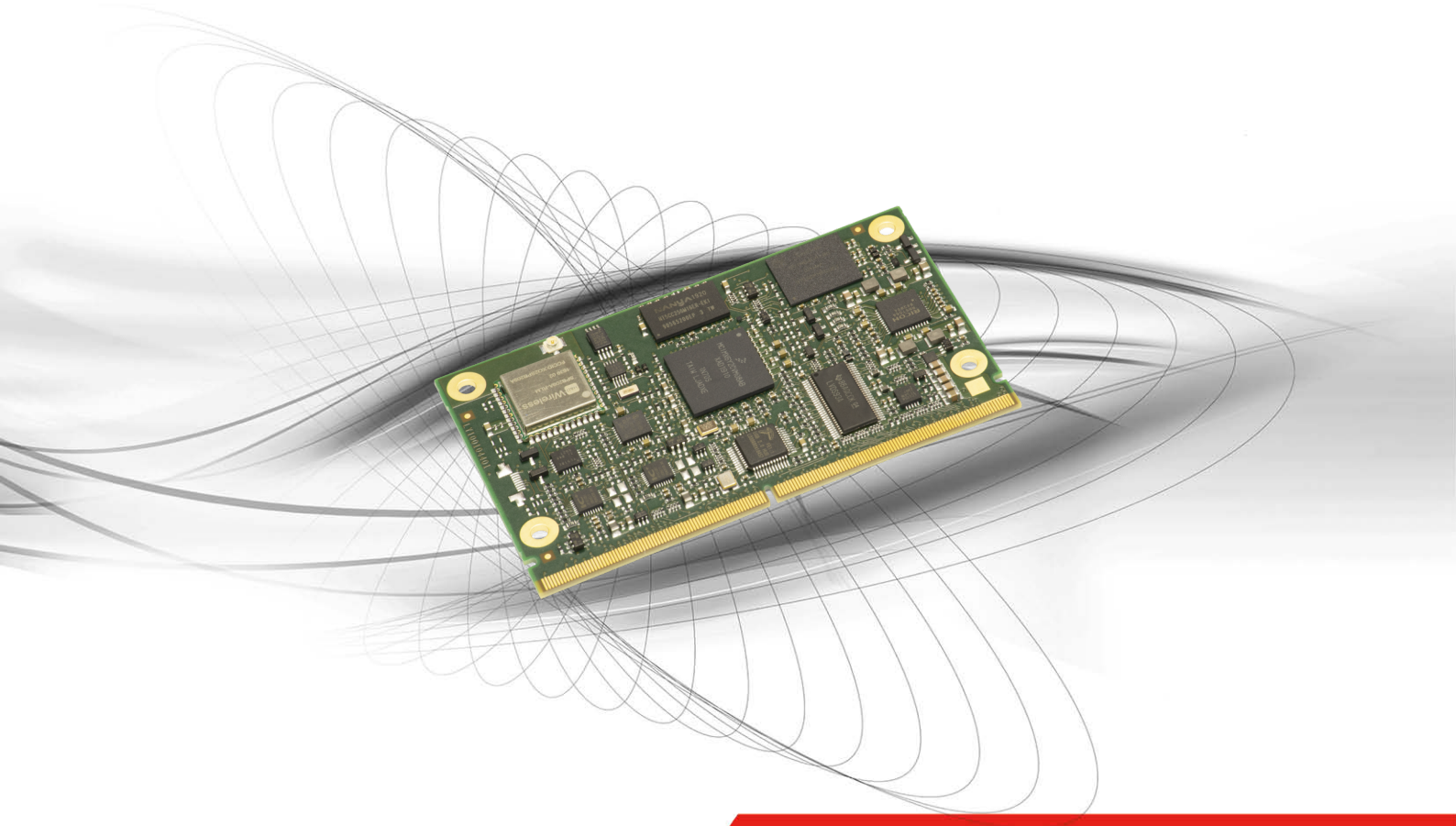
Intelligent Solutions

msc technologies
an Avnet Company



Engineering Leadership

MSC-LDK Manual
BSP msc_sm2_imx6ull V1.6.0

2021-04-27

Copyright Notice

Trademarks

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

Certification

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

Audience

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

Life-Cycle-Management

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

Disclaimer

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

Contents

List of Figures

List of Tables

Listings

1. History

1.1. Document

Revision	Changes
1.6.0	• Added chapter “Hardware monitoring sensors” • Added chapters “Enabling WiFi” and “Use the board as an access point” • Various fixes and cleanups
1.5.0	• Added chapter “I2C Bus Identification” • Added chapter “Reduced Cleanup Time” • Various fixes
1.4.0	• Removed chapter “Microsoft Azure”
1.3.0	• Added chapter “Power Analysis” • Added chapter “MSC-IO” • Added section “Setup Optional Docker Container”
1.2.0	• Added chapter “Microsoft Azure” • Added chapter “Real-Time” • Added chapter “Security” • Added chapter “Tips And Tricks” • Added section “ConnMan Configuration” • Updated section “Deploying Images To The Hardware” (new image type .wic) • Updated section “Enhancing The Images” • Updated section “Installation Of MSC-LDK”
1.1.0	• Added section “Traceable and Reproducable Images” • Added section “Hotfixed and Updating MSC-LDK” • Updated section “Installation of MSC-LDK”

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

1.2. MSC-LDK

Revision	Changes
1.6.0	• Updated to yocto 3.0.1 (zeus) • Added layers meta-chrome-instead-of-otter and boot2qt • Added support for Ryzen and .iMX8 BSPs • "make install" copies .wic.bmap files for use with bmaptool
1.5.0	• Updated to yocto 2.7 (warrior) • Added layer-marvell • Removed meta-iot-cloud and meta-java dependencies for msc-image-lxqt • Added layer meta-bsp-arm-verification • Various improvements to build MSC-LDK BSPs in a Docker container • Various improvements to build feature branches of the BSPs
1.4.0	• Updated to yocto 2.5 (sumo) • Added layer-marvell • Removed meta-iot-cloud and meta-java dependencies for msc-image-lxqt

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

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

1.3. BSP msc_sm2_imx6ull

Revision	Changes
0.2.0	• Added SMARC i.MX6ULL 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 3.0.1 project.

2.1. Scope

This document gives a hand in:

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

2.2. Out Of Scope

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

2.3. Features

Features of the MSC-LDK are:

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

2.4. Conventions

This section describes the conventions used in this manual.

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

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

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

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

Typographical conventions:

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

2.5. Product Support

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

3. Getting Started

3.1. Requirements

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

3.2. Registration On The MSC Git Server

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

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

3.2.1. Creating An SSH Key

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

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

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

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

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

3.2.2. Configuring For HTTP proxies

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

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

4. Using MSC-LDK

4.1. Yocto/MSC-LDK Terms

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

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

4.2. Setup Optional Docker Container

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

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

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

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

e.g. to:

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

To install the container:

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/docker-msc-ldk
user@devhost:$ cd docker-msc-ldk
user@devhost:$ git checkout v1.6.0
```

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

To use the container on an Ubuntu host:

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

To use the container on a CentOS host:

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

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

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

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

4.3. Installation Of MSC-LDK

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

```
user@devhost:$ git clone ssh://gitolite@msc-git02.msc-ge.com:9418/msc_ol99/msc-ldk
user@devhost:$ cd msc-ldk
user@devhost:$ git checkout v1.6.0
```

No files will be installed in other directories.

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

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

The question has to be answered with "no"

4.4. Directory Layout

Table 4.1. – MSC-LDK Directory Layout

Directory	Contents
`build/01044`	BSP build directory.
`doc`	This contains all the generated Yocto documentation. (only if build with: make doc)
`downloads/`	All downloaded sources are stored here. This directory can be shared with other MSC-LDK installations.
`scripts/`	Build helper scripts.
`sources/01044/*.git`	BSP specific layers. (will be created in chapter 4.3)
`sources/meta-freescale.git/`	Freescale specific layers, only on Freescale architectures.
`sources/meta-msc-ldk-*.git/`	MSC-LDK layers.
`sources/meta-openembedded-*.git/`	Additional useful tools that are not part of Yocto.
`sources/meta-qt5*.git/`	Qt5 layers used by msc-image-lxqt.
`sources/yocto.git/`	Yocto sources used by MSC-LDK.
`sstate-cache/`	All built files are stored here and reused on next build. This directory can be shared with other MSC-LDK installations.

4.4.1. MSC-LDK Layers

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

Figure 4.1. – MSC-LDK Layers

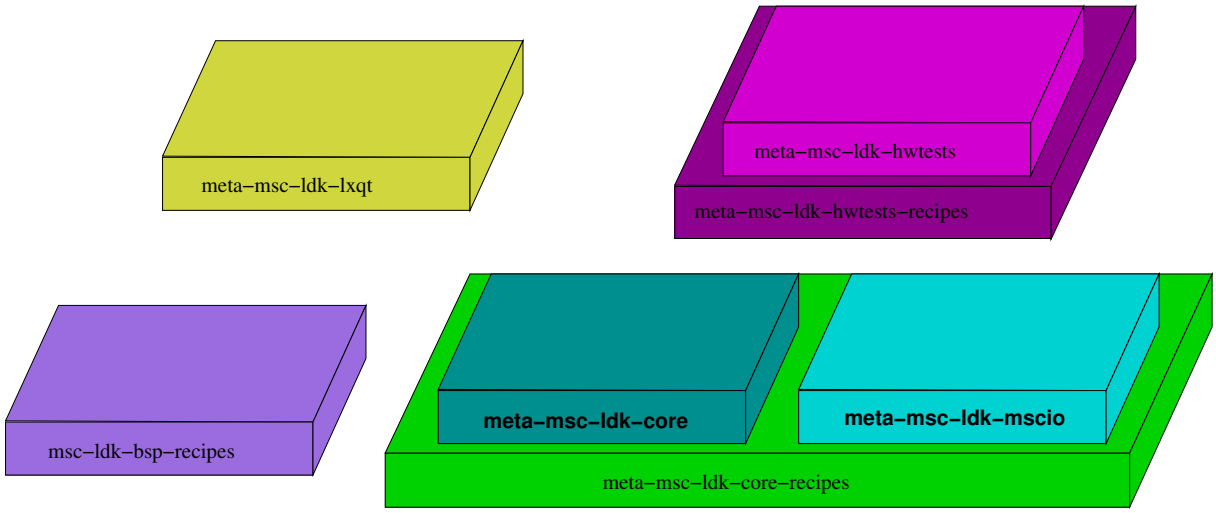


Table 4.2. – MSC-LDK Layer Description

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

4.5. Installation Of MSC-LDK BSPs

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

This BSP covers all variants of the msc-sm2s-imx6ull modules. The different modules are identified via boardinfo stored in the onboard eeprom. Different hardware versions are defined via devicetrees.

To generate an SD-Card image for any sm2s-imx6ull module, do:

```
user@devhost:msc-ldk$ ./setup.py --bsp=01044
```

This will clone the repository `01044/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/01044
user@devhost:msc-ldk/build/01044$
```

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

4.5.1. Repeated setup for target

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

```
ERROR: '/work/msc-ldk/build/01044*' 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=01044 --re-create-conf
```

And then continue as usual:

```
user@devhost:msc-ldk$ cd build/01044
user@devhost:msc-ldk/build/01044$
```

4.6. Reduced Cleanup Time

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

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

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

4.7. Building Images

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

```
user@devhost:msc-ldk$ ./setup.py --bsp=01044
user@devhost:msc-ldk$ cd build/01044
user@devhost:01044$ make
Loading cache: 100% |############################| ETA:  00:00:00
Loaded 4230 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           = "1.44.0"
BUILD_SYS            = "x86_64-linux"
NATIVELSBSTRING      = "linuxmint-19.1"
TARGET_SYS           = "arm-poky-linux-gnueabi"
MACHINE              = "msc-sm2-imx6ull"
DISTRO               = "poky"
DISTRO_VERSION       = "3.0.1"
TUNE_FEATURES        = "arm armv7a vfp thumb neon callconvention-hard"
TARGET_FPU           = "hard"
meta-poky
meta-yocto-bsp       = "zeus-msc:9a27336bf057ef68b3dd3a2c752abd6c1921b5f3"
meta                 = "develop:e09e2c9a8c8da436b08c463f9657f6168c476e6c"
meta                 = "zeus-msc:9a27336bf057ef68b3dd3a2c752abd6c1921b5f3"
meta-oe
meta-networking
meta-python          = "zeus-msc:d076e6e069ffa8a31da3d7337b4d395becd97980"
meta-qt5.git         = "zeus-msc:432ad2aa6c3a13253fefc909faba368851d21fb1"
meta
meta-integrity
meta-signing-key
meta-tpm
meta-tpm2            = "zeus-msc:b2fce5b8c527240ac8c3fa44cf46055ee66b7cd3"
meta-msc-ldk-core-recipes.git = "develop:ddec5c73fb303e209a5c21c3406386ae408d37d7"
meta-msc-ldk-core.git = "develop:f69e243b57b00e854c3c9a3f0be2b9611a095f2e"
meta-msc-ldk-mscio.git = "develop:dd46dc058064b622804619d9a614c078613f2dc8"
meta-freescale.git   = "zeus:4af1824f3daa13bd2bceafe8de7f371b35e0dfce"
meta-qt5-extra.git   = "zeus-msc:94aa90a4e051a62fdfc65dce44a3da8686e2d65b"
meta-msc-arm-extensions.git = "develop:9585c2d3a0ddc9b10c3c055b329e746060c75f2c"
meta-msc-ldk-marvell.git = "develop:f742a8a4899bed58d0ff18228bc1bee815e0e96c"
meta-gnome
meta-multimedia      = "zeus-msc:d076e6e069ffa8a31da3d7337b4d395becd97980"
```

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_sm2_imx6ull-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/01044
user@devhost:msc-ldk/build/01044$ make msc-image-base
```

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

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

4.11. Deploying Images To The Hardware

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

4.11.1. SD card

To create a bootable SD Card with a root filesysystem, copy the image `msc-image-basic-msc-sm2-imx6ull.wic` to the SD-Card (e.g. on `/dev/sdd`) with:

```
user@devhost:$ cd msc-ldk/build/01044/tmp/deploy/images/msc-sm2-imx6ull
user@devhost:images$ sudo dd if=msc-image-basic-msc-sm2-imx6ull.wic  of=/dev/sdd bs=4M
```

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

```
U-Boot SPL 2020.01-+g16a3026afe (Feb 02 2021 - 15:04:21 +0000)

Reading board info...

------------------------------
company .......... msc
form factor ...... sm2s
platform ......... imx6ull
processor ........ Y2
feature .......... 93N02E1I
serial ........... 10111021509
revision (MES) ... A0
boot count ....... 198
------------------------------
Trying to boot from MMC1


U-Boot 2020.01-+g16a3026afe (Feb 02 2021 - 15:04:21 +0000)

CPU:   Freescale i.MX6ULL rev1.1 at 396 MHz
CPU:   Reset cause is POR
Board: MSC SMARC2 i.MX6ULL
I2C:   ready
DRAM:  512 MiB
------------------------------
company .......... msc
form factor ...... sm2s
platform ......... imx6ull
processor ........ Y2
feature .......... 93N02E1I
serial ........... 10111021509
revision (MES) ... A0
boot count ....... 199
------------------------------
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
Loading Environment from MMC... OK
In:    serial
Out:   serial
```

```
Err:   serial
BOOT_SEL[0..2]: 1. Booting from mmc0...
Net:   FEC0
Hit any key to stop autoboot:  0
Boardinfo: OK, complete.
Attempting mmc0 boot...
switch to partitions #0, OK
mmc0 is current device
Loading environment (uEnv.txt) from MMC0 ...
Loading linux image (boot/zImage) from MMC0 ...
7865208 bytes read in 536 ms (14 MiB/s)
Booting from MMC0 ...
Loading FDT image (boot/msc-sm2s-imx6ull-Y2-93N02E1I-1280x800-lvds.dtb) from MMC0 ...
32376 bytes read in 352 ms (88.9 KiB/s)
## Flattened Device Tree blob at 81000000
   Booting using the fdt blob at 0x81000000
   Loading Device Tree to 9f569000, end 9f573e77 ... OK

Starting kernel ...

[    0.000000] Booting Linux on physical CPU 0x0
[    0.000000] Linux version 5.4.8-yocto-standard-00318-g60b6a6328c92-dirty (dher@msc-aac-
    mint06) (gcc version 9.2.1 20191025 (GNU Toolchain for the A-profile Architecture
    9.2-2019.12 (arm-9.10))) #293 SMP Thu Feb 25 10:04:02 CET 2021
[    0.000000] CPU: ARMv7 Processor [410fc075] revision 5 (ARMv7), cr=10c5387d
[    0.000000] CPU: div instructions available: patching division code
[    0.000000] CPU: PIPT / VIPT nonaliasing data cache, VIPT aliasing instruction cache
[    0.000000] OF: fdt: Machine model: MSC i.MX6ULL Smarc module 93N02E1I
[    0.000000] Memory policy: Data cache writealloc
[    0.000000] cma: Reserved 64 MiB at 0x9b400000
[    0.000000] percpu: Embedded 15 pages/cpu s31756 r8192 d21492 u61440
[    0.000000] Built 1 zonelists, mobility grouping on.  Total pages: 130048
[    0.000000] Kernel command line: console=ttymxc0,115200 root=/dev/mmcblk0p2 rootwait rw
    board_name= board_variant=- serial_number=10111021509 uboot_version=<U-Boot 2020.01-+
    g16a3026afe (Feb 02 2021 - 15:04:21 +0000)> consoleblank=0
[    0.000000] Dentry cache hash table entries: 65536 (order: 6, 262144 bytes, linear)
[    0.000000] Inode-cache hash table entries: 32768 (order: 5, 131072 bytes, linear)
[    0.000000] mem auto-init: stack:off, heap alloc:off, heap free:off
[    0.000000] Memory: 437664K/524288K available (10240K kernel code, 496K rwdata, 3676K
    rodata, 1024K init, 411K bss, 21088K reserved, 65536K cma-reserved, 0K highmem)
[    0.000000] SLUB: HWalign=64, Order=0-3, MinObjects=0, CPUs=1, Nodes=1
[    0.000000] rcu: Hierarchical RCU implementation.
[    0.000000] rcu:     RCU event tracing is enabled.
[    0.000000] rcu:     RCU restricting CPUs from NR_CPUS=4 to nr_cpu_ids=1.
[    0.000000] rcu: RCU calculated value of scheduler-enlistment delay is 10 jiffies.
[    0.000000] rcu: Adjusting geometry for rcu_fanout_leaf=16, nr_cpu_ids=1
[    0.000000] NR_IRQS: 16, nr_irqs: 16, preallocated irqs: 16
[    0.000000] Switching to timer-based delay loop, resolution 41ns
[    0.000019] sched_clock: 32 bits at 24MHz, resolution 41ns, wraps every 89478484971ns
[    0.000059] clocksource: mxc_timer1: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
    79635851949 ns
[    0.002811] Console: colour dummy device 80x30
[    0.002880] Calibrating delay loop (skipped), value calculated using timer frequency..
    48.00 BogoMIPS (lpj=240000)
[    0.002912] pid_max: default: 32768 minimum: 301
[    0.003221] Mount-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
[    0.003253] Mountpoint-cache hash table entries: 1024 (order: 0, 4096 bytes, linear)
[    0.004743] CPU: Testing write buffer coherency: ok
[    0.005393] CPU0: update cpu_capacity 1024
[    0.005425] CPU0: thread -1, cpu 0, socket 0, mpidr 80000000
[    0.006682] Setting up static identity map for 0x80100000 - 0x80100078
[    0.006962] rcu: Hierarchical SRCU implementation.
[    0.014444] smp: Bringing up secondary CPUs ...
[    0.014482] smp: Brought up 1 node, 1 CPU
[    0.014504] SMP: Total of 1 processors activated (48.00 BogoMIPS).
[    0.014518] CPU: All CPU(s) started in SVC mode.
[    0.015440] devtmpfs: initialized
[    0.031633] random: get_random_u32 called from bucket_table_alloc+0x58/0x14c with
    crng_init=0
[    0.032417] VFP support v0.3: implementor 41 architecture 2 part 30 variant 7 rev 5
[    0.033117] clocksource: jiffies: mask: 0xffffffff max_cycles: 0xffffffff, max_idle_ns:
    19112604462750000 ns
[    0.033172] futex hash table entries: 256 (order: 2, 16384 bytes, linear)
[    0.037829] pinctrl core: initialized pinctrl subsystem
[    0.040030] NET: Registered protocol family 16
[    0.043445] DMA: preallocated 256 KiB pool for atomic coherent allocations
[    0.045766] cpuidle: using governor menu
[    0.060865] vdd3p0: supplied by regulator-dummy
[    0.062093] cpu: supplied by regulator-dummy
[    0.063310] vddsoc: supplied by regulator-dummy
[    0.085476] No ATAGs?
[    0.085607] hw-breakpoint: found 5 (+1 reserved) breakpoint and 4 watchpoint registers.
[    0.085643] hw-breakpoint: maximum watchpoint size is 8 bytes.
[    0.089315] imx6ul-pinctrl 20e0000.iomuxc: initialized IMX pinctrl driver
[    0.090735] imx6ul-pinctrl 2290000.iomuxc-snvs: no groups defined in /soc/aips-bus@2200000
    /iomuxc-snvs@2290000
```

```
[    0.090877] imx6ul-pinctrl 2290000.iomuxc-snvs: initialized IMX pinctrl driver
[    0.153459] GPIO line 132 (sdio_mux) hogged as output/low
[    0.153519] GPIO line 131 (lvds_shtdn_n) hogged as output/high
[    0.161260] mxs-dma 1804000.dma-apbh: initialized
[    0.167890] vgaarb: loaded
[    0.169467] SCSI subsystem initialized
[    0.170644] usbcore: registered new interface driver usbfs
[    0.170782] usbcore: registered new interface driver hub
[    0.170986] usbcore: registered new device driver usb
[    0.173510] i2c-gpio i2c@1: using lines 129 (SDA) and 128 (SCL)
[    0.176336] i2c i2c-0: IMX I2C adapter registered
[    0.177904] i2c i2c-1: IMX I2C adapter registered
[    0.178342] mc: Linux media interface: v0.10
[    0.178454] videodev: Linux video capture interface: v2.00
[    0.178587] pps_core: LinuxPPS API ver. 1 registered
[    0.178607] pps_core: Software ver. 5.3.6 - Copyright 2005-2007 Rodolfo Giometti <
    giometti@linux.it>
[    0.178658] PTP clock support registered
[    0.179372] Advanced Linux Sound Architecture Driver Initialized.
[    0.181156] Bluetooth: Core ver 2.22
[    0.181279] NET: Registered protocol family 31
[    0.181298] Bluetooth: HCI device and connection manager initialized
[    0.181333] Bluetooth: HCI socket layer initialized
[    0.181357] Bluetooth: L2CAP socket layer initialized
[    0.181407] Bluetooth: SCO socket layer initialized
[    0.183410] clocksource: Switched to clocksource mxc_timer1
[    0.183750] VFS: Disk quotas dquot_6.6.0
[    0.183907] VFS: Dquot-cache hash table entries: 1024 (order 0, 4096 bytes)
[    0.204784] thermal_sys: Registered thermal governor 'step_wise'
[    0.205394] NET: Registered protocol family 2
[    0.206661] tcp_listen_portaddr_hash hash table entries: 512 (order: 0, 6144 bytes, linear
    )
[    0.206732] TCP established hash table entries: 4096 (order: 2, 16384 bytes, linear)
[    0.206841] TCP bind hash table entries: 4096 (order: 3, 32768 bytes, linear)
[    0.206997] TCP: Hash tables configured (established 4096 bind 4096)
[    0.207207] UDP hash table entries: 256 (order: 1, 8192 bytes, linear)
[    0.207273] UDP-Lite hash table entries: 256 (order: 1, 8192 bytes, linear)
[    0.207631] NET: Registered protocol family 1
[    0.208848] RPC: Registered named UNIX socket transport module.
[    0.208876] RPC: Registered udp transport module.
[    0.208892] RPC: Registered tcp transport module.
[    0.208909] RPC: Registered tcp NFSv4.1 backchannel transport module.
[    0.210107] PCI: CLS 0 bytes, default 64
[    0.212485] hw perfevents: enabled with armv7_cortex_a7 PMU driver, 5 counters available
[    0.216351] Initialise system trusted keyrings
[    0.216792] workingset: timestamp_bits=30 max_order=17 bucket_order=0
[    0.231517] NFS: Registering the id_resolver key type
[    0.231611] Key type id_resolver registered
[    0.231629] Key type id_legacy registered
[    0.231728] jffs2: version 2.2. (NAND) (C) 2001-2006 Red Hat, Inc.
[    0.232886] fuse: init (API version 7.31)
[    0.324440] Key type asymmetric registered
[    0.324470] Asymmetric key parser 'x509' registered
[    0.324520] io scheduler mq-deadline registered
[    0.324541] io scheduler kyber registered
[    0.330640] pwm-backlight backlight_display: backlight_display supply power not found,
    using dummy regulator
[    0.342888] 2020000.serial: ttymxc0 at MMIO 0x2020000 (irq = 23, base_baud = 5000000) is a
    IMX
[    1.028549] printk: console [ttymxc0] enabled
[    1.034898] 21e8000.serial: ttymxc1 at MMIO 0x21e8000 (irq = 64, base_baud = 5000000) is a
    IMX
[    1.045099] 21ec000.serial: ttymxc2 at MMIO 0x21ec000 (irq = 65, base_baud = 5000000) is a
    IMX
[    1.154630] random: fast init done
[    1.806912] tpm_i2c_infineon 0-0020: could not request locality
[    1.816438] panel-lvds-msc panel: prepare (POW_ON to video data) set to: 0 ms
[    1.823781] panel-lvds-msc panel: enable (video data to valid frame) set to: 0 ms
[    1.831309] panel-lvds-msc panel: disable (time to turn display off) set to: 0 ms
[    1.838886] panel-lvds-msc panel: unprepare (time to poweroff completely and powerofftime)
    set to: 0 ms
[    1.858570] [drm] Supports vblank timestamp caching Rev 2 (21.10.2013).
[    1.865344] [drm] No driver support for vblank timestamp query.
[    1.872847] [drm] Initialized mxsfb-drm 1.0.0 20160824 for 21c8000.lcdif on minor 0
[    1.898771] Console: switching to colour frame buffer device 100x30
[    1.915417] mxsfb 21c8000.lcdif: fb0: mxsfb-drmdrmfb frame buffer device
[    1.947007] brd: module loaded
[    1.968218] loop: module loaded
[    1.973190] at24 1-0057: 8192 byte 24c64 EEPROM, writable, 16 bytes/write
[    1.986600] boarddata 4-0050: new boarddata driver installed!
[    1.994421] DCDC1: Bringing 1450000uV into 1425000-1425000uV
[    2.012332] LDO3: Bringing 3300000uV into 3000000-3000000uV
[    2.029140] spi_imx 200c000.spi: set DMA-mode
[    2.048439] spi-nor spi1.0: w25q64fw (8192 Kbytes)
[    2.053310] 1 fixed-partitions partitions found on MTD device spi1.0
[    2.059751] Creating 1 MTD partitions on "spi1.0":
```

```
[    2.064644] 0x000000000000-0x000000800000 : "space2_0"
[    2.087410] spi-nor spi1.1: w25q64fw (8192 Kbytes)
[    2.092286] 1 fixed-partitions partitions found on MTD device spi1.1
[    2.098755] Creating 1 MTD partitions on "spi1.1":
[    2.103695] 0x000000000000-0x000000800000 : "space2_1"
[    2.112130] spi_imx 200c000.spi: probed
[    2.116825] spi_imx 2014000.spi: set DMA-mode
[    2.136620] spi-nor spi3.0: w25q64fw (8192 Kbytes)
[    2.141492] 1 fixed-partitions partitions found on MTD device spi3.0
[    2.147928] Creating 1 MTD partitions on "spi3.0":
[    2.152754] 0x000000000000-0x000000800000 : "space1_0"
[    2.175647] spi-nor spi3.1: w25q64fw (8192 Kbytes)
[    2.180520] 1 fixed-partitions partitions found on MTD device spi3.1
[    2.186958] Creating 1 MTD partitions on "spi3.1":
[    2.191784] 0x000000000000-0x000000800000 : "space1_1"
[    2.200297] spi_imx 2014000.spi: probed
[    2.206249] libphy: Fixed MDIO Bus: probed
[    2.211220] CAN device driver interface
[    2.222892] pps pps0: new PPS source ptp0
[    2.229295] libphy: fec_enet_mii_bus: probed
[    2.237058] fec 20b4000.ethernet eth0: registered PHC device 0
[    2.245820] pps pps1: new PPS source ptp1
[    2.280563] libphy: fec_enet_mii_bus: probed
[    2.286671] fec 2188000.ethernet eth1: registered PHC device 1
[    2.293167] e1000e: Intel(R) PRO/1000 Network Driver - 3.2.6-k
[    2.299247] e1000e: Copyright(c) 1999 - 2015 Intel Corporation.
[    2.305771] igb: Intel(R) Gigabit Ethernet Network Driver - version 5.6.0-k
[    2.312768] igb: Copyright (c) 2007-2014 Intel Corporation.
[    2.320127] ehci_hcd: USB 2.0 'Enhanced' Host Controller (EHCI) Driver
[    2.326916] ehci-pci: EHCI PCI platform driver
[    2.331563] ehci-mxc: Freescale On-Chip EHCI Host driver
[    2.337849] usbcore: registered new interface driver usb-storage
[    2.347926] imx_usb 2184000.usb: 2184000.usb supply vbus not found, using dummy regulator
[    2.360210] ci_hdrc ci_hdrc.0: EHCI Host Controller
[    2.365732] ci_hdrc ci_hdrc.0: new USB bus registered, assigned bus number 1
[    2.393558] ci_hdrc ci_hdrc.0: USB 2.0 started, EHCI 1.00
[    2.400899] hub 1-0:1.0: USB hub found
[    2.404960] hub 1-0:1.0: 1 port detected
[    2.416420] ci_hdrc ci_hdrc.1: EHCI Host Controller
[    2.421871] ci_hdrc ci_hdrc.1: new USB bus registered, assigned bus number 2
[    2.453521] ci_hdrc ci_hdrc.1: USB 2.0 started, EHCI 1.00
[    2.460865] hub 2-0:1.0: USB hub found
[    2.464953] hub 2-0:1.0: 1 port detected
[    2.473328] input: 20cc000.snvs:snvs-powerkey as /devices/soc0/soc/2000000.aips-bus/20
    cc000.snvs/20cc000.snvs:snvs-powerkey/input/input0
[    2.492738] rtc-rs5c372 0-0032: r2223tl found, 24hr, driver version 0.7.1
[    2.499698] rtc-rs5c372 0-0032: VDD power-on-reset condition detected
[    2.508479] rtc-rs5c372 0-0032: clock needs to be set
[    2.516949] rtc-rs5c372 0-0032: registered as rtc0
[    2.522119] i2c /dev entries driver
[    2.527386] IR NEC protocol handler initialized
[    2.532107] IR RC5(x/sz) protocol handler initialized
[    2.537265] IR RC6 protocol handler initialized
[    2.541817] IR JVC protocol handler initialized
[    2.546402] IR Sony protocol handler initialized
[    2.551037] IR SANYO protocol handler initialized
[    2.555795] IR Sharp protocol handler initialized
[    2.560516] IR MCE Keyboard/mouse protocol handler initialized
[    2.566400] IR XMP protocol handler initialized
[    2.578090] Bluetooth: HCI UART driver ver 2.3
[    2.582580] Bluetooth: HCI UART protocol H4 registered
[    2.588998] sdhci: Secure Digital Host Controller Interface driver
[    2.595316] sdhci: Copyright(c) Pierre Ossman
[    2.599697] sdhci-pltfm: SDHCI platform and OF driver helper
[    2.607396] sdhci-esdhc-imx 2190000.usdhc: Got CD GPIO
[    2.612632] sdhci-esdhc-imx 2190000.usdhc: Got WP GPIO
[    2.652997] mmc0: SDHCI controller on 2190000.usdhc [2190000.usdhc] using ADMA
[    2.698555] mmc1: SDHCI controller on 2194000.usdhc [2194000.usdhc] using ADMA
[    2.713558] mmc0: new high speed SDHC card at address 0002
[    2.728024] mmcblk0: mmc0:0002 N/A   3.72 GiB
[    2.738957]  mmcblk0: p1 p2
[    2.742803] usbcore: registered new interface driver usbhid
[    2.748582] usbhid: USB HID core driver
[    2.762550] sgtl5000 4-000a: sgtl5000 revision 0x11
[    2.769759] sgtl5000 4-000a: Using internal LDO instead of VDDD: check ER1 erratum
[    2.777549] usb 1-1: new high-speed USB device number 2 using ci_hdrc
[    2.808223] fsl-asoc-card sound: ASoC: failed to init link HiFi: -517
[    2.834635] usb 2-1: new high-speed USB device number 2 using ci_hdrc
[    2.843307] imx-sgtl5000 sound: sgtl5000 <-> 202c000.sai mapping ok
[    2.872881] NET: Registered protocol family 10
[    2.886824] Segment Routing with IPv6
[    2.890682] sit: IPv6, IPv4 and MPLS over IPv4 tunneling driver
[    2.896865] mmc1: new HS200 MMC card at address 0001
[    2.903594] mmcblk1: mmc1:0001 DG4008 7.28 GiB
[    2.908867] mmcblk1boot0: mmc1:0001 DG4008 partition 1 4.00 MiB
[    2.916564] NET: Registered protocol family 17
```

```
[    2.921074] can: controller area network core (rev 20170425 abi 9)
[    2.928066] mmcblk1boot1: mmc1:0001 DG4008 partition 2 4.00 MiB
[    2.934867] mmcblk1rpmb: mmc1:0001 DG4008 partition 3 4.00 MiB, chardev (243:0)
[    2.942580] NET: Registered protocol family 29
[    2.947286] can: raw protocol (rev 20170425)
[    2.953849] can: broadcast manager protocol (rev 20170425 t)
[    2.959874]  mmcblk1: p1 p2
[    2.963201] can: netlink gateway (rev 20190810) max_hops=1
[    2.976400] Key type dns_resolver registered
[    2.985442] Registering SWP/SWPB emulation handler
[    2.990600] Loading compiled-in X.509 certificates
[    2.997661] hub 1-1:1.0: USB hub found
[    3.003622] hub 1-1:1.0: 4 ports detected
[    3.032324] imx_thermal tempmon: Industrial CPU temperature grade - max:105C critical:100C
     passive:95C
[    3.046372] rtc-rs5c372 0-0032: hctosys: unable to read the hardware clock
[    3.053736] cfg80211: Loading compiled-in X.509 certificates for regulatory database
[    3.066572] cfg80211: Loaded X.509 cert 'sforshee: 00b28ddf47aef9cea7'
[    3.073530] ALSA device list:
[    3.076696] platform regulatory.0: Direct firmware load for regulatory.db failed with
    error -2
[    3.085536]   #0: msc-sm2-sgtl5000
[    3.089056] cfg80211: failed to load regulatory.db
[    3.095319] usb-storage 2-1:1.0: USB Mass Storage device detected
[    3.102477] scsi host0: usb-storage 2-1:1.0
[    3.185313] EXT4-fs (mmcblk0p2): recovery complete
[    3.192922] EXT4-fs (mmcblk0p2): mounted filesystem with ordered data mode. Opts: (null)
[    3.201201] VFS: Mounted root (ext4 filesystem) on device 179:2.
[    3.208464] devtmpfs: mounted
[    3.213247] Freeing unused kernel memory: 1024K
[    3.218160] Run /sbin/init as init process
INIT: version 2.88 booting
[    3.433510] usb 1-1.2: new high-speed USB device number 3 using ci_hdrc
[    3.692790] usb-storage 1-1.2:1.0: USB Mass Storage device detected
[    3.701347] scsi host1: usb-storage 1-1.2:1.0
Starting udev
[    3.903525] usb 1-1.3: new high-speed USB device number 4 using ci_hdrc
[    3.985598] udevd[188]: starting version 3.2.8
[    4.022528] random: udevd: uninitialized urandom read (16 bytes read)
[    4.030906] random: udevd: uninitialized urandom read (16 bytes read)
[    4.039179] random: udevd: uninitialized urandom read (16 bytes read)
[    4.102410] udevd[189]: starting eudev-3.2.8
[    4.164709] scsi 0:0:0:0: Direct-Access     Kingston DataTravelerMini PMAP PQ: 0 ANSI: 0
    CCS
[    4.182780] usb-storage 1-1.3:1.0: USB Mass Storage device detected
[    4.217948] scsi host2: usb-storage 1-1.3:1.0
[    4.308820] sd 0:0:0:0: [sda] 503808 512-byte logical blocks: (258 MB/246 MiB)
[    4.361043] sd 0:0:0:0: [sda] Write Protect is off
[    4.396542] sd 0:0:0:0: [sda] No Caching mode page found
[    4.401887] sd 0:0:0:0: [sda] Assuming drive cache: write through
[    4.443575] usb 1-1.4: new high-speed USB device number 5 using ci_hdrc
[    4.513539]  sda: sda1
[    4.545177] sd 0:0:0:0: [sda] Attached SCSI removable disk
[    4.726099] scsi 1:0:0:0: Direct-Access     Intenso  Rainbow Line     8.07 PQ: 0 ANSI: 4
[    4.744732] usb-storage 1-1.4:1.0: USB Mass Storage device detected
[    4.799292] sd 1:0:0:0: [sdb] 7680000 512-byte logical blocks: (3.93 GB/3.66 GiB)
[    4.829850] scsi host3: usb-storage 1-1.4:1.0
[    4.877604] sd 1:0:0:0: [sdb] Write Protect is off
[    4.950435] sd 1:0:0:0: [sdb] Write cache: disabled, read cache: enabled, doesn't support
    DPO or FUA
[    5.105759]  sdb: sdb1
[    5.159492] sd 1:0:0:0: [sdb] Attached SCSI removable disk
[    5.286153] scsi 2:0:0:0: Direct-Access     Intenso  Rainbow Line     8.07 PQ: 0 ANSI: 4
[    5.332742] sd 2:0:0:0: [sdc] 7680000 512-byte logical blocks: (3.93 GB/3.66 GiB)
[    5.360850] sd 2:0:0:0: [sdc] Write Protect is off
[    5.377991] sd 2:0:0:0: [sdc] Write cache: disabled, read cache: enabled, doesn't support
    DPO or FUA
[    5.454207]  sdc: sdc1
[    5.493741] sd 2:0:0:0: [sdc] Attached SCSI removable disk
[    5.927321] scsi 3:0:0:0: Direct-Access     Intenso  Rainbow Line     8.07 PQ: 0 ANSI: 4
[    5.972837] sd 3:0:0:0: [sdd] 7680000 512-byte logical blocks: (3.93 GB/3.66 GiB)
[    6.017230] sd 3:0:0:0: [sdd] Write Protect is off
[    6.063681] sd 3:0:0:0: [sdd] Write cache: disabled, read cache: enabled, doesn't support
    DPO or FUA
[    6.238922]  sdd: sdd1
[    6.290967] sd 3:0:0:0: [sdd] Attached SCSI removable disk
[    8.574814] FAT-fs (mmcblk0p1): Volume was not properly unmounted. Some data may be
    corrupt. Please run fsck.
[    8.671644] FAT-fs (mmcblk1p1): Volume was not properly unmounted. Some data may be
    corrupt. Please run fsck.
[    8.726068] EXT4-fs (mmcblk1p2): recovery complete
[    8.730903] EXT4-fs (mmcblk1p2): mounted filesystem with ordered data mode. Opts: (null)
[    9.578428] random: crng init done
[    9.581921] random: 1 urandom warning(s) missed due to ratelimiting
[   10.659307] FAT-fs (sdb1): Volume was not properly unmounted. Some data may be corrupt.
    Please run fsck.
```

```
[   10.669828] FAT-fs (sdc1): Volume was not properly unmounted. Some data may be corrupt.
    Please run fsck.
[   10.775896] FAT-fs (sdd1): Volume was not properly unmounted. Some data may be corrupt.
    Please run fsck.
[   11.071267] FAT-fs (sda1): Volume was not properly unmounted. Some data may be corrupt.
    Please run fsck.
[   11.321688] EXT4-fs (mmcblk0p2): re-mounted. Opts: (null)
hwclock: RTC_RD_TIME: Invalid argument
Wed Feb 10 13:54:05 UTC 2021
INIT: Entering runlevel: 5
Configuring network interfaces... done.
Starting system message bus: dbus.
Starting random number generator daemon.
OpenBSD Secure Shell server not in use (/etc/ssh/sshd_not_to_be_run)
Starting rpcbind daemon...done.
starting statd: done
Starting syslogd/klogd: done
Starting watchdog daemon...done
Starting internet superserver: inetd.
USB stack is already loaded. hwtests-msc-sm2-imx6ull.zip might not be loaded automatically.
    Consider setting command line option 'modprobe.blacklist=xhci_pci,ehci_pci,uhci_hcd'
Starting TCG TSS2 Access Broker and Resource Management daemon: device driver not loaded,
    skipping.
Starting Trusted Computing daemon:
device driver not loaded, skipping.

MSC-LDK LOL99_20200803_V1_6_0-14-gc41e2ad built on Tue Feb  9 16:06:34 UTC 2021 by dher@msc-
    aac-mint06
Poky (Yocto Project Reference Distro) 3.0.1 msc-sm2-imx6ull /dev/ttymxc0

msc-sm2-imx6ull login:
```

4.12. Login

Login is enabled via console or serial console (/dev/ttymxc0, 115200 baud/8 bits/no parity). The headless and Sato images also have telnet login enabled.

Table 4.3. – User Accounts

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

4.13. Enabled Services

Enabled services are:

- telnetd
- Serial console on /dev/ttymxc0 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/01044/conf/local.conf`:

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

Further information can be found here:
http://www.yoctoproject.org/docs/3.0.1/ref-manual/ref-manual.html#var-IMAGE_INSTALL

This can be automatized by calling `setup.py` with the argument `--local-conf` and an existing file whose content should be appended to `build/01044/conf/local.conf`. A non-existing file is silently ignored.

4.14.2. Adding Own Layers

The images can be further enhanced or configured by adding own layers. Extend the file `build/01044/conf/bblayers.conf` by these lines:

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

Then create the layer as described here:
http://www.yoctoproject.org/docs/3.0.1/dev-manual/dev-manual.html#creating-your-own-layer

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

```
user@devhost:msc-ldk$ ./setup.py --bsp=01044 --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/01044/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/01044/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:

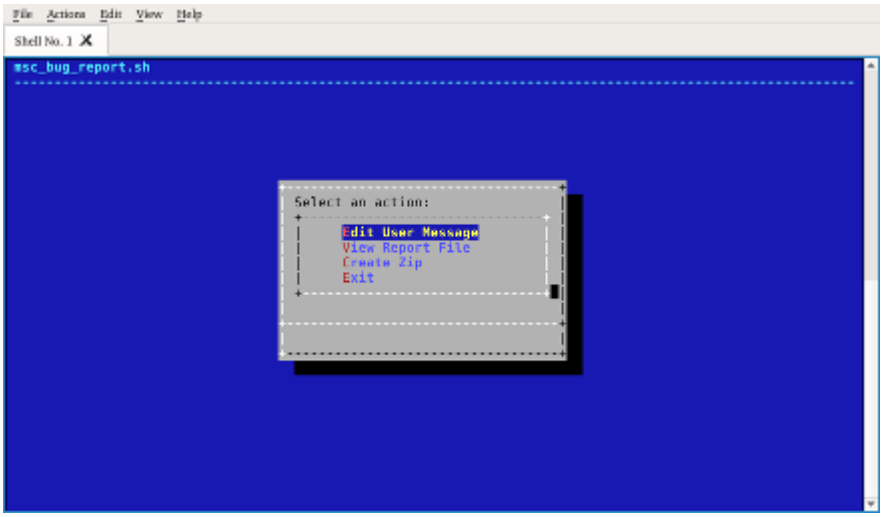
```
# rm -f /etc/localtime
# ln -s /usr/share/zoneinfo/America/New_York /etc/localtime
# echo "America/New_York" > /etc/timezone
```

4.15. 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.

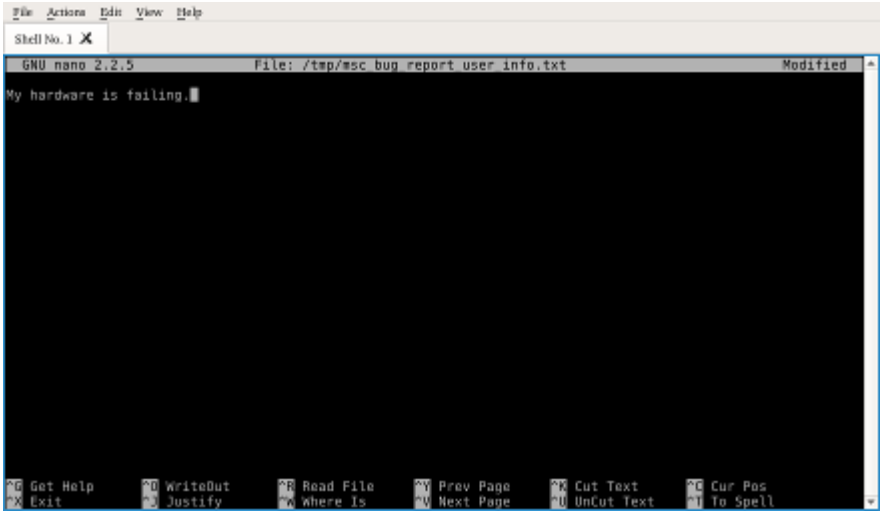
- Run `msc_bug_report.sh`.

Figure 4.2. – Bug Reporting - Start



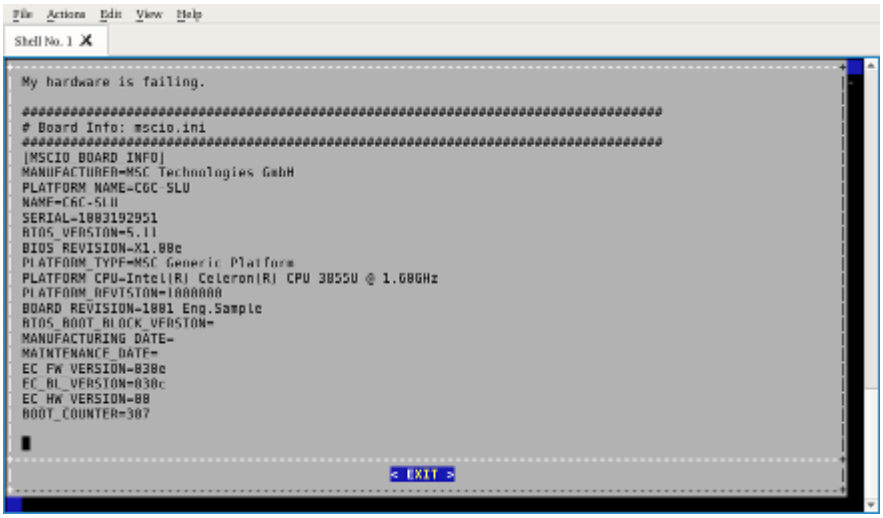
- Select “Edit User Message”.
- Enter bug report message and press Ctrl-O and Ctrl-X.

Figure 4.3. – Bug Reporting - Start



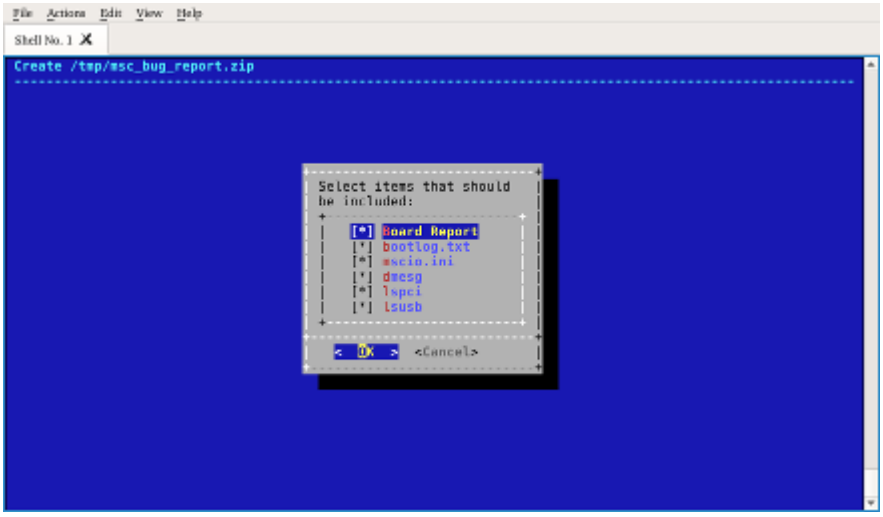
- Optionally you can then view the message with the board report (hardware information).

Figure 4.4. – Bug Reporting - View



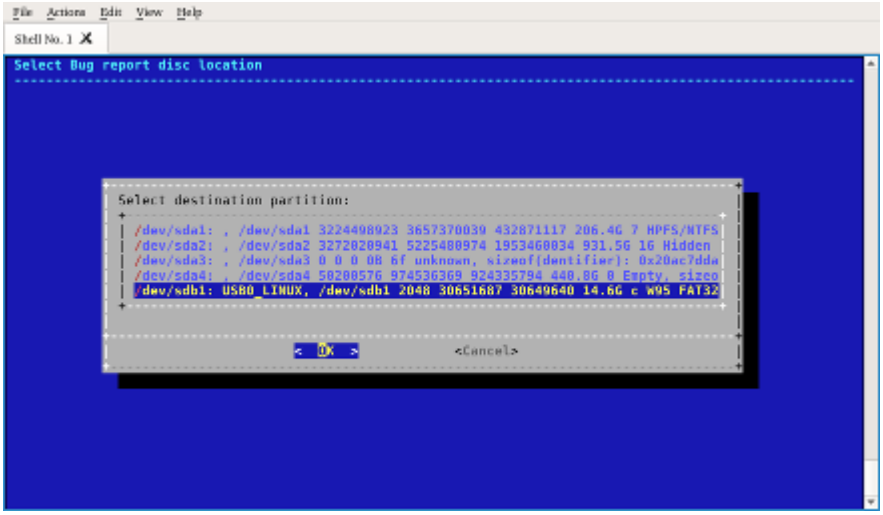
- Press “Create a zip file” and select the components you want to send (e.g. bootlog, mscio.ini, last kernel logs (dmesg) or the installed hardware).

Figure 4.5. – 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 4.6. – 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`)

4.16. Hotfixes And Updating MSC-LDK

Typically twice a year a full MSC-LDK release is created. A release may contain an updated Yocto or other updated layers as well as new supported boards. For each release an own branch is used (e.g. `v1.0.0`) which is tagged with the date encoded (e.g. `LC984_20150421_V0_4_0`, 21st April 2015), too. The release is checked out using the version syntax (`git checkout v1.6.0`) as described above.

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

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

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

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

5. Package List

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

5.1. mscio

This package provides mscio-setup and 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`.

5.2. system-test-controller

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

A. Glossary

BSP Board Support Package. 18, 22

MSC-LDK MSC Linux Development Kit. 14, 18

B. Bootloader Configuration

B.1. Baseboard Configuration

The Devicetrees are designed for the SM2-MB-EP1 baseboard.

Refer to the EP1 Hardware manual für setting the corresponding switches to select your desired boot device.

The baseboard switches TEST# and FORCE_RECOV# have to be set accordingly.

B.2. Boot source selection

BOOT_SELn determines from where U-boot loads kernel and devicetree.

Table B.1. – Bootsel Table

BOOT_SEL2	BOOT_SEL1	BOOT_SEL0	Bootsource
ON	ON	ON	(Carrier SATA) not supported
ON	ON	OFF	Carrier SD Card
ON	OFF	ON	(Carrier eMMC Flash) not supported
ON	OFF	OFF	(Carrier SPI) not supported
OFF	ON	ON	(Module SD) not supported
OFF	ON	OFF	Remote boot net
OFF	OFF	ON	Module eMMC
OFF	OFF	OFF	USB

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

B.2.1. Booting from SD card

To install, copy the appropiate .wic -file from:

`msc-ldk/build/01044/tmp/work/deploy/images/msc-sm2-imx6ull`

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>.wic of=/dev/<sd-card-device> bs=4M
```

To boot from SD card, this boot configuration must be set by enabling the TEST# switch. FORCE_RECOV# must not be set.

B.2.2. Booting from eMMC

At first the eMMC must be programmed. The image build by msc-ldk provides a set of helper scripts to prepare the eMMC:

init_emmc.sh - prepares the emmc for booting from. fmt_emmc.sh - creates partitions on emmc cp_uboot.sh - copies boot loader to emmc cp_rootfs.sh - copies kernel, devicetrees and root-filesystem to emmc

This can be done by booting via SD and then programming the eMMC via

B.3. 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 .wic -file from:
`msc-ldk/build/01044/tmp/work/deploy/images/msc-sm2-imx6ull`
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>.wic of=/dev/<sd-card-device> bs=4M
```

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

```
U-Boot SPL 2020.01-+g16a3026afe (Feb 02 2021 - 15:04:21 +0000)

Reading board info...

------------------------------
company .......... msc
form factor ...... sm2s
platform ......... imx6ull
processor ........ Y2
feature .......... 93N02E1I
serial ........... 10111021509
revision (MES) ... A0
boot count ....... 195
------------------------------
Trying to boot from MMC1


U-Boot 2020.01-+g16a3026afe (Feb 02 2021 - 15:04:21 +0000)

CPU:   Freescale i.MX6ULL rev1.1 at 396 MHz
CPU:   Reset cause is POR
Board: MSC SMARC2 i.MX6ULL
I2C:   ready
DRAM:  512 MiB
------------------------------
company .......... msc
form factor ...... sm2s
platform ......... imx6ull
processor ........ Y2
feature .......... 93N02E1I
serial ........... 10111021509
revision (MES) ... A0
boot count ....... 196
------------------------------
MMC:   FSL_SDHC: 0, FSL_SDHC: 1
Loading Environment from MMC... OK
In:    serial
Out:   serial
Err:   serial
BOOT_SEL[0..2]: 1. Booting from mmc0...
Net:   FEC0
Hit any key to stop autoboot:  0
```

```
Boardinfo: OK, complete.
Attempting mmc0 boot...
switch to partitions #0, OK
mmc0 is current device
Loading environment (uEnv.txt) from MMC0 ...
Loading linux image (boot/zImage) from MMC0 ...
7865208 bytes read in 536 ms (14 MiB/s)
Booting from MMC0 ...
Loading FDT image (boot/msc-sm2s-imx6ull-Y2-93N02E1I-800x480-lvds.dtb) from MMC0 ...
32376 bytes read in 352 ms (88.9 KiB/s)
## Flattened Device Tree blob at 81000000
   Booting using the fdt blob at 0x81000000
   Loading Device Tree to 9f569000, end 9f573e77 ... OK

Starting kernel ...
```

B.4. 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:

```
------------------------------
company .......... msc
form factor ...... sm2s
platform ......... imx6ull
processor ........ Y2
feature .......... 92N0230E
serial ........... 1011021607
revision (MES) ... A0
boot count ....... 42
------------------------------
```

B.5. 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:

```
<company>-<form factor>-<platform>-<feature>-\[emmc-\]<display|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 feature key as shown in boardinfo. Here is an example of setting a 1280x800 LVDS display for board with feature key 92N0230E:

```
=> setenv fdtfile sc-sm2s-imx6ull-Y2-93N0230E-1200x800-lvds.dtb
=> saveenv
Saving Environment to MMC...
Writing to MMC(0)... done
=>
```

B.6. 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

B.7. Change Display settings in U-Boot

There is currentlyno display support in U-Boot.

B.8. 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:  mmc emmc
```

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

```
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
```

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

C. Boarddata

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

C.1. 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-sm2-imx6ull:/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-sm2-imx6ull:/sys/class/boarddata# cat feature_key
93N02E1I
root@msc-sm2-imx6ull:/sys/class/boarddata# cat product_name
msc-sm2s-imx6ull-Y2-93N02E1I
root@msc-sm2-imx6ull:/sys/class/boarddata# cat revision
A0
root@msc-sm2-imx6ull:/sys/class/boarddata# cat serial_number
10111021509
root@msc-sm2-imx6ul:/sys/class/boarddata# cat variant_key
n/a
```

D. Devicetree Configuration

D.1. Concept

The main concept of devicetrees is to abstract the different hardware configurations while using the same kernel. There is a bunch of files in the folder `arch/arm/boot/dts`, you will recognize files with .dts and .dtsi suffix, and perhaps also 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.

D.2. Structure used for i.MX6ULL SMARC

The i.MX6ULL SMARC module uses the following devicetree files:

```
msc-sm2s-imx6ull-Y2-92N0230E-headless.dts
msc-sm2s-imx6ull-Y2-92N0230E-1280x800-lvds.dts
msc-sm2s-imx6ull-Y2-93N02E1I-1280x800-lvds.dts
msc-sm2s-imx6ull-Y2-93N02E1I-headless.dts
msc-sm2s-imx6ull-Z0-82N0N30C-headless.dts

msc-sm2s-imx6ull-Y2-92N0230E-emmc-headless.dts
msc-sm2s-imx6ull-Y2-92N0230E-emmc-1280x800-lvds.dts
msc-sm2s-imx6ull-Y2-93N02E1I-emmc-1280x800-lvds.dts
msc-sm2s-imx6ull-Y2-93N02E1I-emmc-headless.dts
msc-sm2s-imx6ull-Z0-82N0N30C-emmc-headless.dts
```

The upper block of the above listing shows the devicetrees for SD-Card. These devicetrees cannot support WLAN because the hardware allows only mutually exclusive usage. The lower block contain similar devicetrees, but for emmc boot. Here the WLAN functionality is usable.

above devicetrees include other ones:

dtsi-file	**Usage**
msc-sm2s-imx6ull-Z0-82N0N30C.dtsi	for ULZ Module 82N0N30C
msc-sm2s-imx6ull-Y2-92N0230E.dtsi	for ULL Module 92N0230E
msc-sm2s-imx6ull-Y2-93N02E1I.dtsi	for ULL Module 93N02E1I
msc-sm2s-imx6ull-Z0-82N0N30C-emmc.dtsi	for ULZ Module 82N0N30C
msc-sm2s-imx6ull-Y2-92N0230E-emmc.dtsi	for ULL Module 92N0230E
msc-sm2s-imx6ull-Y2-93N02E1I-emmc.dtsi	for ULL Module 93N02E1I
msc-sm2s-imx6ull-1280x800-lvds.dtsi	include for definition of 1280x800 LVDS display
msc-sm2s-imx6ullx.dtsi	common defs for all SM2s-iMX6ULL modules
imx6ulz.dtsi	freescale definitions for i.MX6ULZ
imx6ull.dtsi	freescale definitions for i.MX6ULL

D.3. Boardmarks used for i.MX6ULL SMARC

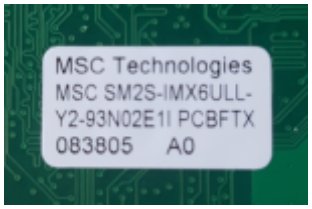


Figure D.1. – example BoardMark

The picture above shows the product label. The modul desription “MSC SM2S-IMX6ULL-Y2-93N02E1I” contains manufacturer, form factor, cpu, cpu version and an 8 digit feature key ABCDEFGH.

Table D.1. – CPU-versions

version	specification
Z0	i.MX6ULZ baseline
Y0	i.MX6ULL baseline
Y1	i.MX6ULL General Purpose 1 (Reduced Feature)
Y2	i.MX6ULL General Purpose 2 (Full Feature)

Each letter defines a feature explained in the following table:

digit	value	description
A	8	256 MB DRAM
	9	512 MB DRAM
B	0	1 GB eMMC
	1	2 GB eMMC
	2	4 GB eMMC
	3	8 GB eMMC
C	N	no SATA
D	0	no other onboard storages
E	N	no display support
	2	LVDS display support
F		Peripheral Options or-ed as on hex-digit
	xxx0	with USB-Hub 4x USB Host + 1x OTG
	xxx1	1x USB Host + 1x OTG
	xx1x	Dual CAN support
	x1xx	SPB209 WLAN modul
	1xxx	Dual 10/100Mbit LAN (only Y2)
G	0	no TPM support
	1	TPM support
H	C	std. temperature range 0 ... +60 °C
	E	ext. temperature range -20 ... +85 °C
	I	industrial temp. range -40 ... +85 °C

So 93N02E1I defines: 512MB DRAM, 8GB eMMC,no Sata, no onboard storage, LVDS, 4x USB-host + 1x OTG, Dual CAN, SPB209, Dual LAN

D.4. Creating own variants

The above description is only for educational purposes, since using msc-ldk you don't operate in kernel source tree, all changes are usually to be applied as patches or recipes.
To create your own device tree, you can use the prepared example `example_devicetree.dts` provided in the layer meta-msc-arm-extensions. This layer is already included in the setup phase of BSP **NOTE**: The layer is checked out from msc-git02.msc-ge.com and therefore it is configured as remote repository. You cannot push any changes here. If you intend to do, you have to connect it to your own repository with:

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

Coming back to the example devicetree, which is located in
`.../meta-msc-arm-extensions.git/recipes-kernel/device-tree/files`,
it's designed as base to be copied/expanded by your needs. Copying it, you have to add the name (ending with .dtb) in
`.../meta-msc-arm-extensions.git/recipes-kernel/device-tree/linux-yocto-custom_%.bbappend`
in the variable named **KERNEL_DEVICETREE**

There are HDMI an LVDS display interfaces as example listed, but commented out to be inactive. You may activate the LVDS variant for this module. Here is the base version of this file:

```
1  /*
2   * Copyright (C) 2017 MSC Technologies, Design Center Aachen
3   *
4   * This program is free software; you can redistribute it and/or modify
5   * it under the terms of the GNU General Public License as published by
6   * the Free Software Foundation; either version 2 of the License, or
7   * (at your option) any later version.
8
9   * This program is distributed in the hope that it will be useful,
10  * but WITHOUT ANY WARRANTY; without even the implied warranty of
11  * MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the
12  * GNU General Public License for more details.
13
14  * You should have received a copy of the GNU General Public License along
15  * with this program; if not, write to the Free Software Foundation, Inc.,
16  * 51 Franklin Street, Fifth Floor, Boston, MA 02110-1301 USA.
17  */
18 /dts-v1/;
19
20 /*****************************************************************************
21  *                                                                           *
22  * Example Devicetree                                                        *
23  *                                                                           *
24  * please select the dtsi file according to your board.                      *
25  *                                                                           *
26  *****************************************************************************/
27 /{};
28 /* enable by changing #if 0 to #if 1 */
29 #if 0
30 #include "msc-<form-factor>-<platform>-<processor>-<feature>.dtsi"
31 #endif
32
33 /* example LVDS Display 800x480 */
34 /* enable by changing #if 0 to #if 1 */
35
36 #if 0
37 /*  modul := [ msc-sm2-imx6ull ] */
38 #include "msc-<modul>-custom-lvds.dtsi"
39 #endif
40
41 /* add your setting below */
```

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

```
msc-sm2s-imx6ull-Z0-82N0N30C.dtsi
msc-sm2s-imx6ull-Y2-92N0230E.dtsi
msc-sm2s-imx6ull-Y2-93N02E1I.dtsi
```

depending on the used module.

NOTE: Don't forget to change the #if 0's into #if 1's or remove them in your copy.
The #if 0's are nessecary, because the example_devicetree is always compiled within a kernel build,

otherways there are sysyntax errors and the build breaks.

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

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

As an example for LVDS display configuration, there is the file `msc-sm2-custom-lvds.dtsi`:

```
1  /*
2   * Copyright (C) 2020 MSC technologies GmbH
3   *
4   */
5
6  &backlight_display {
7          status = "okay";
8  };
9
10 &pwm1 {
11         status = "okay";
12 };
13
14 / {
15         panel {
16                 compatible = "panel-lvds-msc";
17                 width-mm = <261>;
18                 height-mm = <163>;
19                 label = "AA121TH11";
20                 data-mapping = "rgb-24";
21                 backlight = <&backlight_display>;
22                 enable-gpios = <&gpio4 13 GPIO_ACTIVE_HIGH>;
23                 status = "okay";
24
25                 panel-timing {
26                         /* mitsubishi,aa121th11-de1 */
27                         clock-frequency = <71000000>;
28                         hactive = <1280>;
29                         vactive = <800>;
30                         hfront-porch = <60>;
31                         hback-porch = <60>;
32                         vback-porch = <10>;
33                         vfront-porch = <10>;
34                         hsync-len = <40>;
35                         vsync-len = <3>;
36                 };
37
38                 port {
39                         panel_in: endpoint {
40                                 remote-endpoint = <&lcdif_out>;
41                         };
42                 };
43         };
44 };
45
46 &lcdif {
47         assigned-clocks = <&clks IMX6UL_CLK_LCDIF_PRE_SEL>;
48         assigned-clock-parents = <&clks IMX6UL_CLK_PLL2_PFD1>;
49         status = "okay";
50
51         port {
52                 lcdif_out: endpoint {
53                         remote-endpoint = <&panel_in>;
54                 };
55         };
56 };
57
58 &gpio5 {
59         lvds_shtdn_n {
60                         gpio-hog;
61                         gpios = <3 0>;
62                         output-high;
63                         line-name = "lvds_shtdn_n";
64                 };
65 };
```

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

prepare_ms : delay between display power enable to video data **enable_ms** : delay between video data to valid frame (delay for backlight) **disable_ms** : delay for turning display off **unprepare_ms** : delay for poweroff completely and powerofftime

The values of the above setting are printed in dmesg:

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

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

In the section **panel-timing** you can modify the timings for the needs of your used display. Take a look at the **data-mapping**, here are three possible settings: **jeida-18**, **jeida-24** and **vesa-24**. Keep this settings in sync with the entries **fsl,data-width** and **fsl,data-mapping** in line 47 and 48. Enter here **spwg** for vesa-modes and **jeida** for jeida-modes. You may also update **width-mm** (line 15) and **height-mm** (line 16) to get the right dpi scaling. All other settings may be unchanged.

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

This device tree is compiled with the kernel. To rebuild, you can compile the kernel by typing:

```
msc-ldk/build/01044$ ./build.sh bitbake linux-yocto-custom -c deploy
```

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

```
./tmp/deploy/images/msc-sm2-imx6ull/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 wic -image:

```
msc-ldk/build/01044$ ./build.sh bitbake msc-image-base
```

E. Supported Periphery

E.1. Linux

Table E.1. – Supported periphery in Linux

Periphery	Status	Comment
Audio	implemented	see Audio
Backlight	implemented	see Backlight
Buttons	implemented	Buttons
CAN	implemented	see CAN
eMMC	implemented	Device is `/dev/mmcblk1`
EPROM	implemented	see EEPROM
Ethernet	implemented	LAN
GPIOs	implemented	see GPIOs
Hardware sensors	implemented	see Hardware Monitoring
I2C	implemented	see I2C
LVDS	implemented	see LVDS
RTC	implemented	RTC
SD/MMC	implemented	see SD/MMC
SPI	implemented	SPI
TPM	t.b.d.	
UART	implemented	see UART
USB-2.0	implemented	USB2.0
USB OTG	implemented	see USB Gadget/OTG
Watchdogs	implemented	see Watchdog
WLAN	implemented	WLAN / WiFi

E.1.1. Audio

Audio is supported by ALSA. Sound controls can be modified with `alsamixer`, which is very complex, there are above xx controls for the used Audio Codec SGTL5000. 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`.

Playback on Headphone

To play sound on the headphone jack, do:

```
# restore defaults
alsactl restore
amixer sset 'PCM' 100%
# test both channels alternating
speaker-test -p 5 -c 2 -f 800 -t sine
# or play a file (the alsa sounds are all mono)
aplay /usr/share/sounds/alsa/Front_Center.wav
```

E.1.2. 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
```

Record from Mic

To record from the microphone, do:

```
# restore defaults
alsactl restore
amixer sset 'MIC' 100%
# record 10 sec
arecord -c 2 -f S16_LE -r 22050 -d 10 record.wav
# Play back with:
aplay record.wav
```

Persistent Mixer Settings

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

Depending on your changes, the example setting may not work anymore

E.1.3. Buttons

On the baseboard SM2-MB-EP1 the button `POWER#` is available as input event.

It can be accessed with:

```
evtest /dev/input/by-path/platform-20cc000.snvs:snvs-powerkey-event

# pressing the POWER button
Event: time 1613297992.206311, type 1 (EV_KEY), code 116 (KEY_POWER), value 1
Event: time 1613297992.206311, ---------- EV_SYN ------------
Event: time 1613297992.416310, type 1 (EV_KEY), code 116 (KEY_POWER), value 0
Event: time 1613297992.416310, ---------- EV_SYN ------------
```

E.1.4. CAN

Two CAN interfaces are available, `can0` and `can1`.

If they are connected, messages can be sent from one interface to the other via:

```
canconfig can0 bitrate 1000000
canconfig can1 bitrate 1000000

ifconfig can0 up
ifconfig can1 up

candump can0 &
cansend can1 3#00
```

which results in the output:

```
interface = can1, family = 29, type = 3, proto = 1
<0x001> [1] 03
```

E.1.5. EEPROM

The EEPROM device `/sys/bus/i2c/devices/4-0050` contains board data and is reserved.

E.1.6. LAN

There are two 100Mbit interfaces available:

Table E.2. – LAN Ports

name	SM2 Connector	MSC-SM2-MB-EP1
eth0	GBE0	X702A (with USB3)
eth1	GBE1	X1201A (with USB2)

E.1.7. GPIOs

The GPIOs are available via the standard linux sysfs API or with names via MSC-IO.

A) 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 E.3. – GPIOs

GPIO count	gpio base	gpio name
32	0	GPIO_1.x
32	32	GPIO_2.x

Table E.3. – GPIOs (continued)

GPIO count	gpio base	gpio name
32	64	GPIO_3.x
32	96	GPIO_4.x
32	128	GPIO_5.x

B) MSC-IO supports using the same software on different CPU modules where the GPIO numbers might change as the names stay same.

As some GPIOs have a special function on some baseboards.

Table E.4. – GPIOs on baseboard

Baseboard	GPIOs
sm2-mb-ep1	GPIO0, GPIO1, GPIO2, GPIO3, GPIO4, GPIO5, GPIO6, GPIO7, GPIO8, GPIO9, GPIO10, GPIO11

Generic information how to access named GPIOs via MSC-IO can be found in User GPIOs.

E.1.8. I2C

The module features various I2C busses. To simplify usage, they can be used by bus names instead of numbers.

These busses are available:

```
i2c-1   i2c             21a4000.i2c                             I2C adapter
i2c-4   i2c             i2c@1                                   I2C adapter
i2c-0   i2c             21a0000.i2c                             I2C adapter
i2c-1   *link*          pm                                      ID mapped
i2c-4   *link*          gp                                      ID mapped
i2c-0   *link*          dev                                     ID mapped
```

`pm`, `gp` and `dev` are mappings for the SMARC Specification bus names

More details on named I2C busses can be found in I2C Bus Identification.

E.1.9. LVDS

LVDS is the only display interface supported by the sm2s-imx6ull module.

It can be accessed in Linux via `/dev/fb0`.

E.1.10. RTC

RTC is avaiable as /dev/rtc0 which is linked to /dev/rtc.

usually it is used by hwclock:

```
# hwlock
Fre Feb 26 16:29:20 2021  0.000000 seconds
```

For continous operation a backup battery on the baseboard is needed.

E.1.11. SD/MMC

The SD/MMC device is `/dev/mmcblk0`.

It can only used in alternative to WLAN / WiFi
SD/MMC usage is forced by forcing TEST# low.

E.1.12. SPI

The SPI ports are available as predefined spi-nor devices `/dev/mtd<n>` device nodes. They are assigned as:

Table E.5. – SPI Ports

bus.cs	SM2 Connector	MSC-SM2-MB-EP1
spi1.0	SPI_CS0	X1302.8 (mtd0)
spi1.1	SPI_CS1	X1302.3 (mtd1)
spi3.0	ESPI_CS0	X1301.3 (mtd2)
spi3.1	ESPI_CS0	X1301.5 (mtd3)

E.1.13. UART

The UART ports are available via `/dev/ttymxc<n>` device nodes. They are assigned as:

Table E.6. – UART Ports

/dev/<n>	SM2 Connector	MSC-SM2-MB-EP1
ttymxc0	SER0	X2301B (UART0)
ttymxc1	SER1	X2302 (UART1)
ttymxc2	SER2	n.a (UART2)
ttymxc3	SER3	n.a. (UART3)

E.1.14. USB2.0

Depending on the module variant there are 2 or 5 USB2 interfaces.
One port is an OTG Port, the other is host only and optionally connected to an USB-Hub.
you can see the role of the OTG port:

```
# cat /sys/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/role
gadget
```

or, if an device is connected:

```
# cat /sys/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/role
host
```

the other port is always host:

```
# cat /sys/devices/soc0/soc/2100000.aips-bus/2184000.usb/ci_hdrc.0/role
```

E.1.15. USB Gadget/OTG

If no USB OTG cable is used, the USB OTG port is configured for gadget mode on power-on. Therefore USB devices inserted will not be detected.

To use it as USB host port, do:

```
echo host >/sys/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/role
```

To configure it as USB device/gadget port, do:

```
echo gadget >/sys/devices/soc0/soc/2100000.aips-bus/2184200.usb/ci_hdrc.1/role
```

E.1.16. WLAN / WiFi

NOTE: WLAN is only available if the system is started from eMMC.

WLAN connectivity is provided by the `mlan0` interface. To use it, attach an antenna to the module's X0801 connector.

WLAN with msc-image-base

To ensure hardware functionality, enable the driver and scan for all networks.

```
ifconfig mlan0 up
iw mlan0 scan
```

The output then is something like:

```
BSS 1c:6a:7a:9d:5c:70(on mlan0)
        TSF: 9259437323 usec (0d, 02:34:19)
        freq: 2437
        beacon interval: 102 TUs
        capability: ESS ShortPreamble ShortSlotTime RadioMeasure (0x1421)
        signal: -77.00 dBm
        last seen: 0 ms ago
        SSID: Visitor
        Supported rates: 9.0 12.0* 18.0 24.0 36.0 48.0 54.0
        DS Parameter set: channel 6
[...]
```

More sophisticated functionality is provided by the tools `wpa_supplicant` and `wpa_cli`, whose documentation is not covered here.

E.2. U-Boot

Table E.7. – Supported periphery in U-Boot

Periphery	Status	Comment
eMMC	partially (no booting)	see eMMC
Ethernet	implemented	see Ethernet
I2C	implemented	see I2C
SD/MMC	implemented	see SD/MMC
USB	implemented	see USB

E.2.1. eMMC

To use the eMMC:

```
=> mmc dev 1
switch to partitions #0, OK
mmc1(part0) is current device
=> mmc info
Device: FSL_SDHC
Manufacturer ID: 45
OEM: 100
Name: DG400
Bus Speed: 52000000
Mode: MMC High Speed (52MHz)
Rd Block Len: 512
MMC version 5.1
[..]
```

To list the partition table of the EMMC chip:

```
=> mmc part

Partition Map for MMC device 1  --   Partition Type: DOS

Part    Start Sector    Num Sectors     UUID            Type
  1     8192            32768           eb81df2d-01     0c
  2     40960           15232640        eb81df2d-02     83
```

contents of the partitions

```
=> fatls mmc 1:1

0 file(s), 0 dir(s)

=> ext4ls mmc 1:2
<DIR>       4096 .
<DIR>       4096 ..
<DIR>      16384 lost+found
<DIR>       4096 boot
<DIR>       4096 mnt
<DIR>       4096 home
<DIR>       4096 sbin
<DIR>       4096 proc
<DIR>       4096 run
<DIR>       4096 etc
<SYM>          8 tmp
<DIR>       4096 usr
<DIR>       4096 bin
<DIR>       4096 var
<DIR>       4096 sys
<DIR>       4096 media
<DIR>       4096 lib
<DIR>       4096 dev
```

E.2.2. Ethernet

The default ethernet port is eth0.

To retrieve an IP address via DHCP:

```
=> dhcp
BOOTP broadcast 1
BOOTP broadcast 2
BOOTP broadcast 3
DHCP client bound to address 192.168.2.12 (1027 ms)
```

To set an IP address manually:

```
=> setenv ipaddr 192.168.150.3
```

To ping a remote server 192.168.2.1:

```
=> ping 192.168.2.1
Using FEC0 device
host 192.168.2.1 is alive
```

eth1 cannot be used in u-boot.

E.2.3. I2C

There are three I2C busses supported:

Table E.8. – I2C Busses

Name	Device/Bus number	Comment
dev	0	module internal bus
pm	1	SMARC bus powermanagement
gp	2	SMARC general purpose bus

In U-boot the busses are named different, but the bus number is the same

```
=> i2c bus
Bus 0:  mxc0
Bus 1:  mxc1
Bus 2:  soft00
```

To scan the bus, select the bus number with `i2c dev #`, e.g. for the `dev` bus:

```
=> i2c dev 1
Setting bus to 1
=> i2c probe
Valid chip addresses: 57
```

E.2.4. SD/MMC

To use the SD/MMC, run:

```
=> mmc dev 0
switch to partitions #0, OK
mmc0 is current device
=> mmc info
Device: FSL_SDHC
Manufacturer ID: 3
OEM: 5344
Name: SU08G
Bus Speed: 50000000
Mode: SD High Speed (50MHz)
Rd Block Len: 512
SD version 3.0
[..]
```

To list the content of the SD card, do:

```
=> fatls mmc 0:1
    32240   msc-sm2s-imx6ull-Y2-92N0230E-1280x800-lvds.dtb
    32123   msc-sm2s-imx6ull-Y2-92N0230E-emmc-1280x800-lvds.dtb
    31327   msc-sm2s-imx6ull-Y2-92N0230E-emmc-headless.dtb
    31432   msc-sm2s-imx6ull-Y2-92N0230E-headless.dtb
    32376   msc-sm2s-imx6ull-Y2-93N02E1I-1280x800-lvds.dtb
    32219   msc-sm2s-imx6ull-Y2-93N02E1I-emmc-1280x800-lvds.dtb
    31423   msc-sm2s-imx6ull-Y2-93N02E1I-emmc-headless.dtb
    31568   msc-sm2s-imx6ull-Y2-93N02E1I-headless.dtb
    29796   msc-sm2s-imx6ull-Z0-82N0N30C-emmc-headless.dtb
    26304   msc-sm2s-imx6ull-Z0-82N0N30C-headless.dtb
 75618975   rootfs.tgz
  7838640   zImage

13 file(s), 0 dir(s)
```

If the SD card has been replaced since the last reset, it has to be registered manually in the system. Run:

```
=> mmc rescan
```

E.2.5. USB

To determine all connected USB devices, run `usb start` and then `usb tree`. Whenever a device is changed, run `usb rescan`.

```
=> usb start
USB0:  USB EHCI 1.00
scanning bus 0 for devices... EHCI timed out on TD - token=0x80008d80

      USB device not accepting new address (error=22)
4 USB Device(s) found
USB1:  USB EHCI 1.00
scanning bus 1 for devices... 2 USB Device(s) found
       scanning usb for storage devices... 3 Storage Device(s) found

=> usb tree
USB device tree:
  1  Hub (480 Mb/s, 0mA)
  |  u-boot EHCI Host Controller
  |
  +-2  Hub (480 Mb/s, 100mA)
    |   USB 2.0 Hub [MTT]
    |
    +-3  Mass Storage (480 Mb/s, 200mA)
    |    Alcor Tech Intenso Rainbow Line 16090200023400
    |
    +-4  Mass Storage (480 Mb/s, 200mA)
         Alcor Tech Intenso Rainbow Line 14082900029355

  5  Hub (480 Mb/s, 0mA)
  |  u-boot EHCI Host Controller
  |
  +-6  Mass Storage (480 Mb/s, 200mA)
       Kingston DataTravelerMini 5B691C000991
```

As seen above, there is a timeout problem with some USB-Sticks. This is due a specification violation of these sticks.

F. MSC-IO

F.1. What Is MSC-IO?

MSC-IO provides access to CPU module information and none plug-and-play hardware devices.

F.2. Components

F.2.1. mscio-drivers

For fully supporting the board, some drivers are necessary:

Table F.1. – mscio-drivers

Driver	Feature
`i2c_ids_of`	Maps the I2C numbers to their names. This simplifies accessing them.
`user-gpios`	Maps the numbers of the user GPIOs on CPU module to their names. This simplifies accessing them.

Backlight

The backlight of the LVDS display can be controlled via the backlight devices under `/sys/class/backlight/`.

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

To change the brightness:

```
# Display brightness
cat /sys/class/backlight/pwm_backlight/actual_brightness
25

# Dim it
echo 12 >/sys/class/backlight/pwm_backlight/brightness
```

or:

```
mscio-cmd get backlights
pwm_backlight,

mscio-cmd geti backlight_brightness pwm_backlight
25

mscio-cmd seti backlight_brightness pwm_backlight 12
```

Hardware Monitoring

The CPU module features various sensors. They are accessible through the standard linux hwmon API interface with a few extensions under `/sys/class/hwmon/`. Sensor values can be read from the files named `*_input`, human readable names are available in the matching `*_label` files, e.g. the core voltage from `in0_label` and `in0_input`.

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

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

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

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

To list all sensors, run:

```
mscio-cmd get sensors
temp1@imx_thermal_zone,
```

To read one sensor, run:

```
mscio-cmd get sensor_value temp1@imx_thermal_zone
42856mC
```

Watchdog

The module features one or more watchdog. In case multiple watchdogs are present accessing the correct one might be tricky as their numbers can change.

Typically to start the watchdog:

```
# Trigger every second, reboot after no trigger within 10s
watchdog -t 1 -T 10 -F /dev/watchdog0
# press ctrl-z and wait 10s for a reboot.
```

or with names:

```
mscio-watchdog start -reset-timeout=10s -trigger-time=1s -watchdog=DA9063_Watchdog
```

The device node can be easily retrieved with:

```
mscio-cmd get watchdogs
DA9063_Watchdog,imx2+_watchdog,

mscio-cmd geti watchdog_device DA9063_Watchdog/dev/watchdog0
```

User GPIOs

If the devicetree contains a user GPIO mapping it can be located at `/sys/bus/platform/devices/user_gpios/gpios`. For each GPIO a file with its linux number exists, e.g. `GPI0` or `GPO3`. GPIOs named `GPO` should be used only for output, GPIOs named `GPI` only for inputs.

To read a GPIO0 with the linux API, do:

```
GPIO=$(cat /sys/bus/platform/devices/user_gpios/gpios/GPIO0)
cd /sys/class/gpio
echo ${GPIO} >export
cd gpio${GPIO}
echo in >direction
cat value
```

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

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

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

```
GPIO
  Name: GPIO0
  Number: 28
  CanInput: 1
  CanOutput: 1
  CanInterrupt: 0
  IsInput: 1
  Value: 0
```

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

To list all GPIOs, run:

```
mscio-cmd get gpios
GPIO0,GPIO1,GPIO2,GPIO3,GPIO4,GPIO5,GPIO6,GPIO7,
```

To read the current state, do:

```
mscio-cmd seti gpio_config GPIO0 in
mscio-cmd geti gpio_value GPIO0
0
```

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

```
mscio-cmd wait-for-gpio-interrupt --mode=rising --timeout=2s GPIO0
GPIO level after interrupt: 1
```

Verify that an interrupt has been raised with:

```
cat /proc/interrupts|grep gpiolib
107:          0          0          0          0  gpio-mxc  28 Edge      gpiolib
```

To set the current state to high, do:

```
mscio-cmd seti gpio_config GPIO0 high
# or
mscio-cmd seti gpio_config GPIO0 1
```

F.2.2. I2C Bus Identification

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

The current assignment can be analyzed with this command:

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

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

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

Both approaches are error prone. Therefore a third option exists. On modern systems the firmware provides an device-tree entry `msc,i2c-ids`. The driver `i2c_ids` creates mapping files that contain the I2C bus number which can be found in the directory. `/sys/bus/platform/devices/i2c_ids/i2c_ids`.

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

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

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

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

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

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

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

Board Information

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

To print everything known, do:

```
mscio-example
```

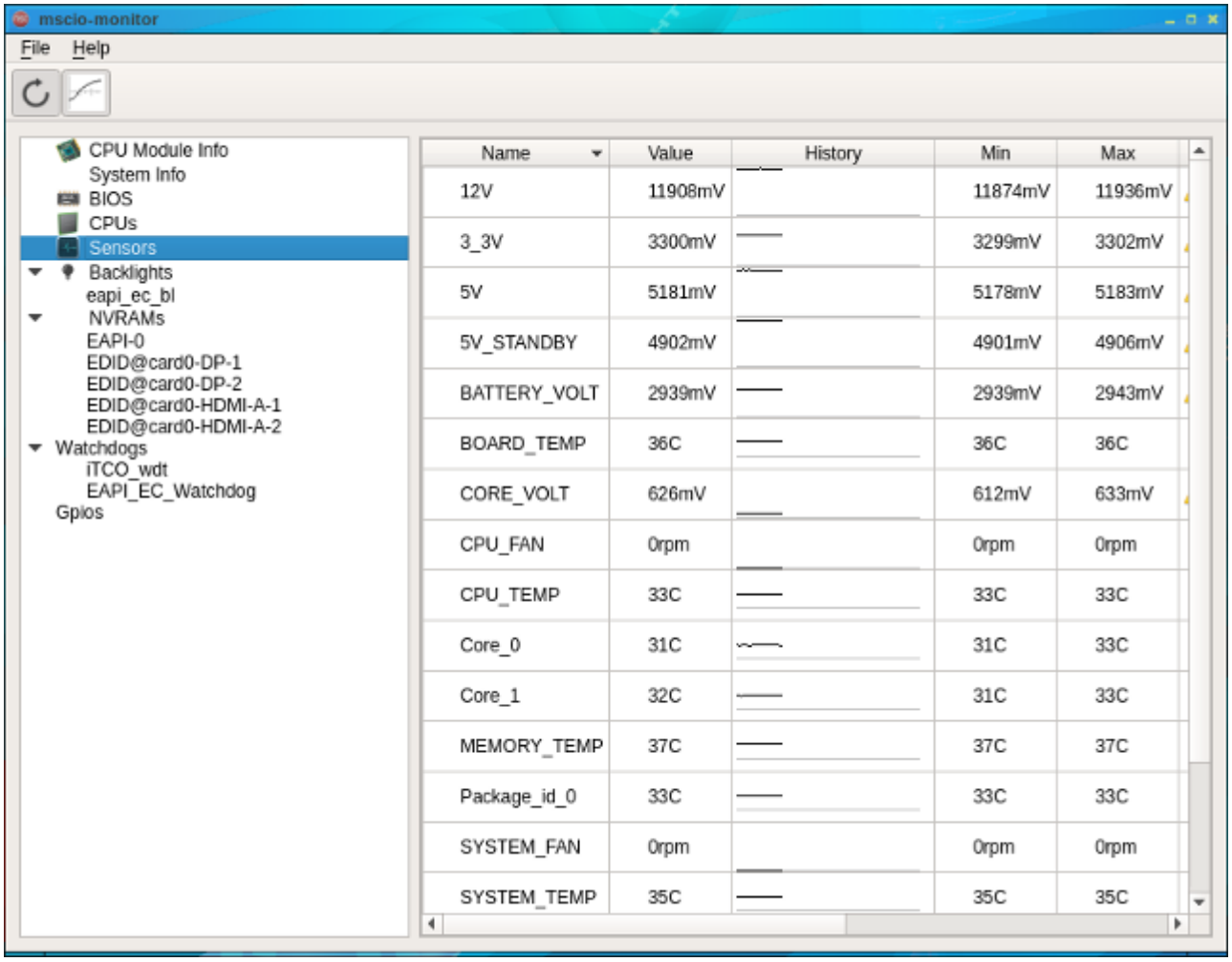
To just retrieve the serial number, do:

```
mscio-cmd get board_serial
1004036878
```

F.2.3. mscio-monitor

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

Figure F.1. – mscio-monitor



F.2.4. User Library

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

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

using namespace Msc::MscIo;
using namespace std;

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

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

    mscio.DeInit();

    return 0;
    }
```

File F.1 – example.cpp

It can be compiled with:

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

Output is:

```
Sensor
  Name: temp1@imx_thermal_zone
  Value: 50C
```

G. Troubleshooting

Table G.1. – Errors

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

H. Tips And Tricks

H.1. Using A Standard Browser

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

H.1.1. Using The google-chrome Browser

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

```
user@devhost:msc-ldk$ ./setup.py --bsp=01044 -layers-lxqt -layers-browser-chrome-instead-of-otter
```

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

H.2. Timekeeping

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

H.2.1. htpdate

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

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

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

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

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

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

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

To update the RTC, do:

```
hwclock --systohtc
```

I. Security

I.1. Introduction

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

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

b) Increase the security of a running system.

I.2. Layer meta-security-isafw

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

Adding the following line in `local.conf` enables a post processing step after the image creation:

```
INHERIT += "isafw"
```

In that post processing step are several checks performed and the result is written to `build/01044/tmp/log/isafw-report*/`:

CFA : Analyze executables on the image

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

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

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

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

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

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

KCA : Kernel settings

LA : Undesired recipe licenses

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

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

I.3. Setting Up A Project

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

```
user@devhost:msc-ldk$ ./setup.py --bsp=01044 --layers-lxqt --layers-security
user@devhost:msc-ldk$ cd build/01044-security
user@devhost:msc-ldk/build/01044-security$ make msc-image-lxqt
```

J. Power Analysis

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

The most import CPU core states are:

Table J.1. – CPU core states

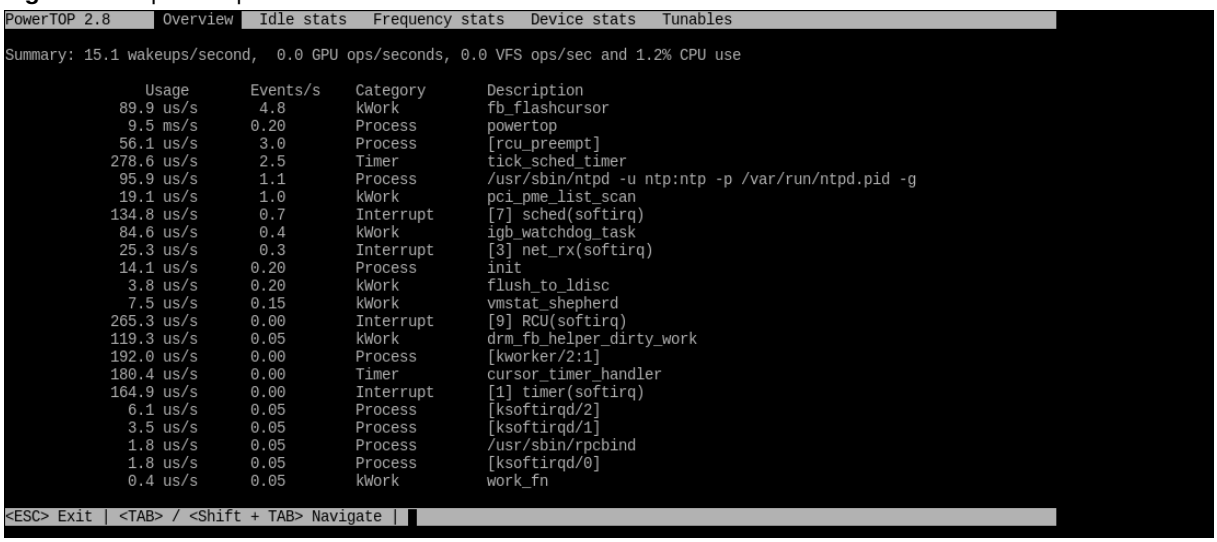
Mode	Description
C0	CPU fully turned on
C1	The main CPU clocks used for executing instructions are stopped, all others are still on
C3	All internal CPU clocks are stopped.
C6	All internal CPU clocks are stopped and internal voltage is reduced.

Depending on the CPU more modes might be available.

J.1. powertop

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

Figure J.1. – powertop overview



K. Links

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

L. License Overview

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

Licenses		
(GPLv2+ |LGPLv3)	AFL-2.1	Artistic-1.0
BSD	BSD-2-Clause	BSD-3-Clause
BSD-4-Clause	BSL-1.0	CPL-1.0
Firmware-imx-sdma_firmware	FreeType	GFDL-1.2
GPL-1.0+	GPL-2+	GPL-2.0
GPL-2.0+	GPL-3.0-with-GCC-exception	GPLv1
GPLv2	GPLv2+	GPLv2.0+
GPLv3	GPLv3+	ICU
ISC	LGPL-2.1	LGPL-2.1+
LGPLv2	LGPLv2+	LGPLv2.1
LGPLv2.1+	LGPLv3	LGPLv3+
Libpng	MIT	MIT-X
MIT-style	MPL-1.1	MPL-2.0
MPLv1.1	NTP	PD
PSFv2	Proprietary	Python-2.0
Zlib	bzip2	openssl
zsh		

Table L.2. – License overview listing

Recipe	Version	Licenses
acl *libacl*	2.2.52	LGPLv2.1+
alsa-lib *alsa-conf*	1.1.9 *alsa-lib*	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*	1.1.9 *alsa-utils-aconnect* *alsa-utils-alsaloop* *alsa-utils-alsatplg* *alsa-utils-amixer* *alsa-utils-aseqdump* *alsa-utils-iecset* *alsa-utils-speakertest*	GPLv2+
attr *libattr*	2.4.47	LGPLv2.1+
avahi *avahi-daemon* *libavahi-common*	0.7 *avahi-locale-en-gb* *libavahi-core*	GPLv2+, LGPLv2.1+
base-files *base-files*	3.0.14	GPLv2
base-passwd *base-passwd*	3.5.29	GPLv2+
bash *bash*	5.0	GPLv3+
bc *bc*	1.07.1	GPLv3+
binutils *binutils*	2.32.0 *libbfd*	GPLv3
bluez5 *bluez5* *bluez5-obex*	5.50 *bluez5-noinst-tools*	GPLv2+, LGPLv2.1+
bonnie++ *bonnie++*	1.03e	GPLv2
boost *boost-filesystem* *boost-regex*	1.71.0 *boost-iostreams*	BSL-1.0, MIT, Python-2.0
bootlog *bootlog*	1.0	MIT
busybox *busybox* *busybox-syslog*	1.31.0 *busybox-hwclock* *busybox-udhcpc*	GPLv2, bzip2
bzip2 *libbz2*	1.0.8	bzip2
ca-certificates *ca-certificates*	20190110	GPL-2.0+, MPL-2.0
cairo *cairo*	1.16.0 *cairo-gobject*	MPL-1.1, LGPLv2.1
cifs-utils *cifs-utils*	6.4	GPLv3, LGPLv3
cmdline-keyboard *cmdline-keyboard*	1.0	MIT

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

Recipe	Version	Licenses
connman *connman*	1.37 *connman-client*	GPLv2
connman-info *connman-info*	1.0	GPLv2
coreutils *coreutils*	8.31	GPLv3+
cpufrequtils *cpufrequtils*	008	GPLv2
curl *libcurl*	7.66.0	MIT
dbus *dbus*	1.12.16 *dbus-lib*	AFL-2.1, GPLv2+
devmem2 *devmem2*	1.0	GPLv2+
dialog *dialog*	1.3-20190728	LGPL-2.1
dosfstools *dosfstools*	4.1	GPLv3
e2fsprogs *e2fsprogs-badblocks* *e2fsprogs-mke2fs* *libcomerr* *libext2fs*	1.45.3 *e2fsprogs-e2fsck* *e2fsprogs-tune2fs* *libe2p*	GPLv2
elfutils *elfutils* *libelf*	0.177 *libdw*	GPLv3+
ell *ell*	0.22	LGPLv2.1
ethtool *ethtool*	5.2	GPLv2+
eudev *eudev* *libudev*	3.2.8 *eudev-hwdb*	GPLv2.0+, LGPL-2.1+
evtest *evtest*	1.31	GPLv2
expat *expat*	2.2.8	MIT
firmware-imx *firmware-imx-vpu-imx6d*	8.1.1 *firmware-imx-vpu-imx6q*	Proprietary
flac *libflac*	1.3.3	GFDL-1.2, GPLv2+, LGPLv2.1+, BSD
fontconfig *fontconfig*	2.13.1	MIT-style, MIT, PD
freetype *freetype*	2.10.1	FreeType, GPLv2+
gcc-runtime *libstdc++*	9.2.0	GPL-3.0-with-GCC-exception
gdbm *gdbm*	1.18.1 *gdbm-compat*	GPLv3
glib-2.0	2.60.7	LGPLv2.1+, BSD, PD

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

Recipe	Version	Licenses
glib-2.0	*glib-2.0-locale-en-gb*	
glibc	2.30	GPLv2, LGPLv2.1
glibc	*glibc-dbg*	
glibc-locale	2.30	GPLv2, LGPLv2.1
glibc-locale-en-gb	*locale-base-de-de*	
locale-base-en-gb	*locale-base-en-us*	
gmp	6.1.2	GPLv2+, LGPLv3+
gmp		
gnutls	3.6.8	LGPLv2.1+
gnutls		
gobject-introspection	1.60.2	LGPLv2+, GPLv2+
gobject-introspection		
grep	3.3	GPLv3
grep		
hdparm	9.58	BSD
hdparm		
helper-tools	1.0	GPLv2
helper-tools		
hostapd	2.9	BSD-3-Clause
hostapd		
htpdate	1.1.3	GPLv2
htpdate		
i2c-ids	1.0	MIT
i2c-ids		
i2c-tools	4.1	GPLv2+
i2c-tools		
i2c-write-read	git	GPLv2
i2c-write-read		
icu	64.2	ICU
libicudata	*libicui18n*	
libicuuc		
init-ifupdown	1.0	GPLv2
init-ifupdown		
initscripts	1.0	GPLv2
initscripts	*initscripts-functions*	
iperf3	3.7	BSD
iperf3		
iptables	1.8.3	GPLv2+
iptables	*iptables-module-ip6t-ah*	
iptables-module-ip6t-dnat	*iptables-module-ip6t-dnpt*	
iptables-module-ip6t-dst	*iptables-module-ip6t-eui64*	
iptables-module-ip6t-frag	*iptables-module-ip6t-hbh*	
iptables-module-ip6t-hl	*iptables-module-ip6t-icmp6*	
iptables-module-ip6t-ipv6header	*iptables-module-ip6t-log*	
iptables-module-ip6t-masquerade	*iptables-module-ip6t-mh*	
iptables-module-ip6t-netmap	*iptables-module-ip6t-redirect*	
iptables-module-ip6t-reject	*iptables-module-ip6t-rt*	
iptables-module-ip6t-snat	*iptables-module-ip6t-snpt*	
iptables-module-ip6t-srh	*iptables-module-ipt-ah*	
iptables-module-ipt-clusterip	*iptables-module-ipt-dnat*	
iptables-module-ipt-ecn	*iptables-module-ipt-icmp*	
iptables-module-ipt-log	*iptables-module-ipt-masquerade*	
iptables-module-ipt-netmap	*iptables-module-ipt-realm*	
iptables-module-ipt-redirect	*iptables-module-ipt-reject*	
iptables-module-ipt-snat	*iptables-module-ipt-ttl*	
iptables-module-ipt-ulog	*iptables-module-xt-addrtype*	

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

Recipe	Version	Licenses
iptables-module-xt-audit	*iptables-module-xt-bpf*	
iptables-module-xt-cgroup	*iptables-module-xt-checksum*	
iptables-module-xt-classify	*iptables-module-xt-cluster*	
iptables-module-xt-comment	*iptables-module-xt-connbytes*	
iptables-module-xt-connlimit	*iptables-module-xt-connmark*	
iptables-module-xt-connsecmark	*iptables-module-xt-conntrack*	
iptables-module-xt-cpu	*iptables-module-xt-ct*	
iptables-module-xt-dccp	*iptables-module-xt-devgroup*	
iptables-module-xt-dscp	*iptables-module-xt-ecn*	
iptables-module-xt-esp	*iptables-module-xt-hashlimit*	
iptables-module-xt-helper	*iptables-module-xt-hmark*	
iptables-module-xt-idletimer	*iptables-module-xt-ipcomp*	
iptables-module-xt-iprange	*iptables-module-xt-ipvs*	
iptables-module-xt-led	*iptables-module-xt-length*	
iptables-module-xt-limit	*iptables-module-xt-mac*	
iptables-module-xt-mark	*iptables-module-xt-multiport*	
iptables-module-xt-nfacct	*iptables-module-xt-nflog*	
iptables-module-xt-nfqueue	*iptables-module-xt-osf*	
iptables-module-xt-owner	*iptables-module-xt-physdev*	
iptables-module-xt-pkttype	*iptables-module-xt-policy*	
iptables-module-xt-quota	*iptables-module-xt-rateest*	
iptables-module-xt-recent	*iptables-module-xt-rpfilter*	
iptables-module-xt-sctp	*iptables-module-xt-secmark*	
iptables-module-xt-set	*iptables-module-xt-socket*	
iptables-module-xt-standard	*iptables-module-xt-statistic*	
iptables-module-xt-string	*iptables-module-xt-synproxy*	
iptables-module-xt-tcp	*iptables-module-xt-tcpmss*	
iptables-module-xt-tcpoptstrip	*iptables-module-xt-tee*	
iptables-module-xt-time	*iptables-module-xt-tos*	
iptables-module-xt-tproxy	*iptables-module-xt-trace*	
iptables-module-xt-u32	*iptables-module-xt-udp*	
iptables-modules		
iw *iw*	5.3	BSD-2-Clause
kbd *kbd-keymaps*	2.0.4	GPLv2+
kexec-tools *kexec*	2.0.19	GPLv2
kmod *kmod*	26 *libkmod*	GPL-2.0+, LGPL-2.1+
libcap *libcap*	2.27	BSD, GPLv2
libdaemon *libdaemon*	0.14	LGPLv2.1+
libffi *libffi*	3.3 rc0	MIT
libgcc *libgcc*	9.2.0	GPL-3.0-with-GCC-exception
libgcrypt *libgcrypt*	1.8.4	LGPLv2.1+
libgpg-error *libgpg-error*	1.36	GPLv2+, LGPLv2.1+
libical *libical*	3.0.6	LGPLv2.1, MPL-2.0
libidn2 *libidn2*	2.2.0	(GPLv2+ | LGPLv3)
libjitterentropy	2.1.2	GPLv2+, BSD

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

Recipe	Version	Licenses
libjitterentropy		
libmscboost	git	LGPLv2.1
libmscboost		
libmscboostpython	git	LGPLv2.1
libmscboostpython		
libnl	3.5.0	LGPLv2.1
libnl	*libnl-genl*	
libnsl2	1.2.0	LGPL-2.1
libnsl2		
libnss-mdns	0.10	LGPLv2.1+
libnss-mdns		
libogg	1.3.4	BSD
libogg		
libpcap	1.9.1	BSD
libpcap		
libpcre	8.43	BSD
libpcre		
libpng	1.6.37	Libpng
libpng		
libsamplerate0	0.1.9	BSD-2-Clause
libsamplerate0		
libsndfile1	1.0.28	LGPLv2.1
libsndfile1		
libtirpc	1.1.4	BSD
libtirpc		
libunistring	0.9.10	LGPLv3+, GPLv2
libunistring		
libusb1	1.0.22	LGPLv2.1+
libusb1		
libvorbis	1.3.6	BSD
libvorbis		
libxcrypt	4.4.8	LGPLv2.1
libxcrypt		
libxml2	2.9.9	MIT
libxml2	*libxml2-utils*	
linux-firmware	20190815	Firmware-imx-sdma_firmware
linux-firmware-imx-sdma-imx6q	*linux-firmware-imx-sdma-imx7d*	
linux-firmware-imx-sdma-license		
linux-sdio-driver-209a	20200203	GPLv2
kernel-module-bt8xxx-5.4.8-yocto-standard	*kernel-module-mlan-5.4.8-yocto-standard*	
kernel-module-sd8xxx-5.4.8-yocto-standard	*linux-sdio-driver-209a*	
linux-tools-testusb	1.0	GPL-2.0
linux-tools-testusb		
linux-yocto-custom	5.4.8	GPLv2
kernel-base	*kernel-devicetree*	
kernel-image	*kernel-image-zimage*	
kernel-module-asix-5.4.8-yocto-standard	*kernel-module-at24-5.4.8-yocto-standard*	
kernel-module-at25-5.4.8-yocto-standard	*kernel-module-ax88179-178a-5.4.8-yocto-standard*	
kernel-module-binfmt-misc-5.4.8-yocto-standard	*kernel-module-brcmfmac-5.4.8-yocto-standard*	
kernel-module-brcmutil-5.4.8-yocto-standard	*kernel-module-cdc-acm-5.4.8-yocto-standard*	
kernel-module-cdc-eem-5.4.8-yocto-standard	*kernel-module-cdc-ether-5.4.8-yocto-standard*	
kernel-module-cdc-ncm-5.4.8-yocto-standard	*kernel-module-cdc-subset-5.4.8-yocto-standard*	
kernel-module-coda-vpu-5.4.8-yocto-standard	*kernel-module-configfs-5.4.8-yocto-standard*	
kernel-module-crc-ccitt-5.4.8-yocto-standard	*kernel-module-crc-itu-t-5.4.8-yocto-standard*	

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

Recipe	Version	Licenses
kernel-module-crc7-5.4.8-yocto-standard	*kernel-module-dummy-hcd-5.4.8-yocto-standard*	
kernel-module-dw-hdmi-ahb-audio-5.4.8-yocto-standard	*kernel-module-echainiv-5.4.8-yocto-standard*	
kernel-module-ehset-5.4.8-yocto-standard	*kernel-module-evbug-5.4.8-yocto-standard*	
kernel-module-ftdi-sio-5.4.8-yocto-standard	*kernel-module-g-acm-ms-5.4.8-yocto-standard*	
kernel-module-g-audio-5.4.8-yocto-standard	*kernel-module-g-cdc-5.4.8-yocto-standard*	
kernel-module-g-ether-5.4.8-yocto-standard	*kernel-module-g-ffs-5.4.8-yocto-standard*	
kernel-module-g-hid-5.4.8-yocto-standard	*kernel-module-g-mass-storage-5.4.8-yocto-standard*	
kernel-module-g-multi-5.4.8-yocto-standard	*kernel-module-g-ncm-5.4.8-yocto-standard*	
kernel-module-g-serial-5.4.8-yocto-standard	*kernel-module-g-zero-5.4.8-yocto-standard*	
kernel-module-gadgetfs-5.4.8-yocto-standard	*kernel-module-gspca-main-5.4.8-yocto-standard*	
kernel-module-i2c-algo-pca-5.4.8-yocto-standard	*kernel-module-i2c-algo-pcf-5.4.8-yocto-standard*	
kernel-module-imx-vdoa-5.4.8-yocto-standard	*kernel-module-isofs-5.4.8-yocto-standard*	
kernel-module-libcomposite-5.4.8-yocto-standard	*kernel-module-libcrc32c-5.4.8-yocto-standard*	
kernel-module-msdos-5.4.8-yocto-standard	*kernel-module-net1080-5.4.8-yocto-standard*	
kernel-module-net2280-5.4.8-yocto-standard	*kernel-module-nls-iso8859-15-5.4.8-yocto-standard*	
kernel-module-option-5.4.8-yocto-standard	*kernel-module-ov5640-5.4.8-yocto-standard*	
kernel-module-pegasus-5.4.8-yocto-standard	*kernel-module-psmouse-5.4.8-yocto-standard*	
kernel-module-r8152-5.4.8-yocto-standard	*kernel-module-rtl8150-5.4.8-yocto-standard*	
kernel-module-serport-5.4.8-yocto-standard	*kernel-module-snd-hwdep-5.4.8-yocto-standard*	
kernel-module-snd-rawmidi-5.4.8-yocto-standard	*kernel-module-snd-usb-audio-5.4.8-yocto-standard*	
kernel-module-snd-usbmidi-lib-5.4.8-yocto-standard	*kernel-module-spidev-5.4.8-yocto-standard*	
kernel-module-u-audio-5.4.8-yocto-standard	*kernel-module-u-ether-5.4.8-yocto-standard*	
kernel-module-u-serial-5.4.8-yocto-standard	*kernel-module-udf-5.4.8-yocto-standard*	
kernel-module-usb-f-acm-5.4.8-yocto-standard	*kernel-module-usb-f-ecm-5.4.8-yocto-standard*	
kernel-module-usb-f-ecm-subset-5.4.8-yocto-standard	*kernel-module-usb-f-eem-5.4.8-yocto-standard*	
kernel-module-usb-f-fs-5.4.8-yocto-standard	*kernel-module-usb-f-hid-5.4.8-yocto-standard*	
kernel-module-usb-f-mass-storage-5.4.8-yocto-standard	*kernel-module-usb-f-midi-5.4.8-yocto-standard*	
kernel-module-usb-f-ncm-5.4.8-yocto-standard	*kernel-module-usb-f-obex-5.4.8-yocto-standard*	
kernel-module-usb-f-printer-5.4.8-yocto-standard	*kernel-module-usb-f-rndis-5.4.8-yocto-standard*	
kernel-module-usb-f-serial-5.4.8-yocto-standard	*kernel-module-usb-f-ss-lb-5.4.8-yocto-standard*	
kernel-module-usb-f-uac1-5.4.8-yocto-standard	*kernel-module-usb-f-uac2-5.4.8-yocto-standard*	
kernel-module-usb-f-uvc-5.4.8-yocto-standard	*kernel-module-usb-wwan-5.4.8-yocto-standard*	
kernel-module-usbnet-5.4.8-yocto-standard	*kernel-module-usbserial-5.4.8-yocto-standard*	
kernel-module-usbtest-5.4.8-yocto-standard	*kernel-module-uvcvideo-5.4.8-yocto-standard*	
kernel-module-videobuf2-vmalloc-5.4.8-yocto-standard	*kernel-module-wl12xx-5.4.8-yocto-standard*	
kernel-module-wlcore-5.4.8-yocto-standard	*kernel-module-wlcore-sdio-5.4.8-yocto-standard*	
kernel-module-zaurus-5.4.8-yocto-standard	*kernel-modules*	
lrzsz *lrzsz*	0.12.20	GPLv2+
ltrace *ltrace*	7.91	GPLv2
mem-edit *mem-edit*	git	GPLv2
memtester *memtester*	4.3.0	GPLv2
minicom *minicom*	2.7.1	GPLv2+
mmc-utils *mmc-utils*	0.1	GPLv2
mobile-broadband-provider-info *mobile-broadband-provider-info*	20190618	PD
modutils-initscripts *modutils-initscripts*	1.0	PD
msc-bug-report *msc-bug-report*	1.0	GPLv2
msc-completion *msc-completion*	1.0	MIT
msc-cpufreq *msc-cpufreq*	1.0	GPLv2

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

Recipe	Version	Licenses
msc-init-script-early *msc-init-script-early*	1.0	MIT
msc-init-script-late *msc-init-script-late*	1.0	MIT
msc-ldk-benchmark *msc-ldk-benchmark*	git	GPLv2
msc-ldk-verification *msc-ldk-verification*	git	GPLv2
msc-ldk-verification-apps *msc-ldk-verification-apps*	git	GPLv2
msc-ldk-verification-script *msc-ldk-verification-script*	1.0	MIT
msc-linux-scripts *msc-linux-scripts*	git	GPLv2
mscio-cmd *mscio-cmd*	git	LGPLv2.1
mscio-drivers *kernel-module-devreg-5.4.8-yocto-standard* *kernel-module-eapi-ec-bl-5.4.8-yocto-standard* *kernel-module-eapi-ec-running-time-5.4.8-yocto-standard* *kernel-module-eapi-nvram-5.4.8-yocto-standard* *kernel-module-user-gpios-5.4.8-yocto-standard*	git *kernel-module-eapi-ec-5.4.8-yocto-standard* *kernel-module-eapi-ec-hwm-5.4.8-yocto-standard* *kernel-module-eapi-ec-wdt-5.4.8-yocto-standard* *kernel-module-i2c-ids-of-5.4.8-yocto-standard* *mscio-drivers*	GPLv2
mscio-lib *mscio-lib*	git	LGPLv2.1
mtd-utils *mtd-utils*	2.1.1	GPLv2+
nano *nano*	2.2.5	GPLv2
ncurses *ncurses-libformw* *ncurses-libncurses* *ncurses-libpanelw* *ncurses-terminfo*	6.1 *ncurses-libmenuw* *ncurses-libncursesw* *ncurses-libtinfo* *ncurses-terminfo-base*	MIT
neard *neard*	0.16	GPLv2
netbase *netbase*	5.6	GPLv2
netpipe *netpipe*	3.7.2	GPLv1
nettle *nettle*	3.5.1	LGPLv3+, GPLv2+
nfs-utils *nfs-utils-client*	2.4.1 *nfs-utils-mount*	MIT, GPLv2+, BSD
ntp *ntp* *ntpdate*	4.2.8p13 *ntp-tickadj*	NTP
ofono *ofono*	1.30	GPLv2
openssh *openssh* *openssh-scp* *openssh-sshd*	8.0p1 *openssh-keygen* *openssh-ssh*	BSD, ISC, MIT
openssl *libcrypto*	1.1.1d *libssl*	openssl

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

Recipe	Version	Licenses
openssl	*openssl-bin*	
openssl-conf		
opkg-utils	0.4.1	GPLv2+
update-alternatives-opkg		
packagegroup-base	1.0	MIT
packagegroup-base	*packagegroup-base-3g*	
packagegroup-base-alsa	*packagegroup-base-bluetooth*	
packagegroup-base-extended	*packagegroup-base-ipv6*	
packagegroup-base-nfc	*packagegroup-base-nfs*	
packagegroup-base-pci	*packagegroup-base-usbgadget*	
packagegroup-base-usbhost	*packagegroup-base-wifi*	
packagegroup-base-zeroconf	*packagegroup-distro-base*	
packagegroup-machine-base		
packagegroup-core-boot	1.0	MIT
packagegroup-core-boot		
packagegroup-core-ssh-openssh	1.0	MIT
packagegroup-core-ssh-openssh		
packagegroup-msc-ldk-core	1.0	MIT
packagegroup-msc-ldk-core		
packagegroup-mscio	1.0	MIT
packagegroup-mscio		
packagegroup-tpm	1.0	MIT
packagegroup-tpm		
parted	3.2	GPLv3+
parted		
pci2uio	git	GPLv2
kernel-module-pci2uio-5.4.8-yocto-standard	*pci2uio*	
pciutils	3.6.2	GPLv2+
libpci	*pciutils*	
pciutils-ids		
perl	5.30.0	Artistic-1.0, GPL-1.0+
perl	*perl-module-config-heavy*	
pixman	0.38.4	MIT, MIT-style, PD
pixman		
pm-utils	1.4.1	GPLv2
pm-utils		
powertop	2.10	GPLv2
powertop	*powertop-locale-en-gb*	
powertop-locale-en-us		
python	2.7.17	PSFv2
libpython2	*python-argparse*	
python-codecs	*python-compression*	
python-contextlib	*python-core*	
python-crypt	*python-ctypes*	
python-datetime	*python-email*	
python-fcntl	*python-io*	
python-lang	*python-logging*	
python-math	*python-mime*	
python-netclient	*python-numbers*	
python-pickle	*python-shell*	
python-stringold	*python-subprocess*	
python-textutils	*python-threading*	
python-unixadmin	*python-zlib*	
python-pyserial	3.4	BSD
python-pyserial		

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

Recipe	Version	Licenses
python3	3.7.5	PSFv2
libpython3	*python3-2to3*	
python3-asyncio	*python3-audio*	
python3-codecs	*python3-compile*	
python3-compression	*python3-core*	
python3-crypt	*python3-ctypes*	
python3-curses	*python3-datetime*	
python3-db	*python3-debugger*	
python3-difflib	*python3-distutils*	
python3-distutils-windows	*python3-doctest*	
python3-email	*python3-fcntl*	
python3-html	*python3-idle*	
python3-image	*python3-io*	
python3-json	*python3-logging*	
python3-mailbox	*python3-math*	
python3-mime	*python3-misc*	
python3-mmap	*python3-modules*	
python3-multiprocessing	*python3-netclient*	
python3-netserver	*python3-numbers*	
python3-pathlib	*python3-pickle*	
python3-pkgutil	*python3-plistlib*	
python3-pprint	*python3-profile*	
python3-pydoc	*python3-resource*	
python3-shell	*python3-smtpd*	
python3-sqlite3	*python3-stringold*	
python3-syslog	*python3-terminal*	
python3-threading	*python3-tkinter*	
python3-typing	*python3-unittest*	
python3-unixadmin	*python3-venv*	
python3-xml	*python3-xmlrpc*	
python3-dbus	1.2.12	MIT
python3-dbus		
python3-pycairo	1.18.1	LGPLv2.1, MPLv1.1
python3-pycairo		
python3-pygobject	3.34.0	LGPLv2.1
python3-pygobject		
readline	8.0	GPLv3+
readline		
rfkill	0.5	BSD
rfkill		
rng-tools	6.7	GPLv2
rng-tools		
rpcbind	1.2.5	BSD
rpcbind		
run-postinsts	1.0	MIT
run-postinsts		
shadow	4.6	BSD, Artistic-1.0
shadow	*shadow-base*	
shadow-securetty	4.6	MIT
shadow-securetty		
shared-mime-info	1.10	GPLv2
shared-mime-info		
smartmontools	6.3	GPLv2
smartmontools-ctl		
spi-register	git	GPLv2
spi-register		
sqlite3	3.29.0	PD
libsqlite3		

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

Recipe	Version	Licenses
strace *strace*	5.3	LGPL-2.1+, GPL-2+
sudo *sudo*	1.8.27	ISC, BSD, Zlib
sysfsutils *libsysfs*	2.1.0	LGPLv2.1
system-test-controller-config *system-test-controller-config*	1.0	MIT
system-test-controller-v2 *system-test-controller-v2*	git	GPLv2
sysvinit *sysvinit*	2.88dsf *sysvinit-pidof*	GPLv2+
sysvinit-inittab *sysvinit-inittab*	2.88dsf	GPLv2
tcp-wrappers *libwrap*	7.6	BSD
tcpdump *tcpdump*	4.9.3	BSD
tiny-shell *tiny-shell*	git	GPLv2
tpm-tools *tpm-tools*	1.3.9.1	CPL-1.0
tpm2-abrmd *tpm2-abrmd*	2.0.3	BSD-2-Clause
tpm2-tools *tpm2-tools*	3.1.1	BSD
tpm2-tss *libtss2* *libtss2-tcti-device* *tpm2-tss*	2.0.0 *libtss2-mu* *libtss2-tcti-mssim*	BSD-2-Clause
trousers *libtspi*	0.3.14 *trousers*	BSD
tzdata *tzdata* *tzdata-americas* *tzdata-arctic* *tzdata-atlantic* *tzdata-core* *tzdata-misc* *tzdata-posix*	2019c *tzdata-africa* *tzdata-antarctica* *tzdata-asia* *tzdata-australia* *tzdata-europe* *tzdata-pacific* *tzdata-right*	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
update-flash *update-flash*	git	GPLv2
update-rc.d *update-rc.d*	0.8	GPLv2+
usbutils *usbutils*	012	GPLv2+

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

Recipe	Version	Licenses
useradd-msc *useradd-msc*	1.0	MIT
util-linux	2.34	GPLv2+, LGPLv2.1+, BSD-3-Clause, BSD-4-Clause
util-linux-blkid *util-linux-fstrim* *util-linux-libmount* *util-linux-libuuid* *util-linux-lscpu* *util-linux-sulogin* *util-linux-swapon* *util-linux-umount*	*util-linux-dmesg* *util-linux-libblkid* *util-linux-libsmartcols* *util-linux-lsblk* *util-linux-mount* *util-linux-swapoff* *util-linux-swaponoff*	
valgrind *valgrind*	3.15.0	GPLv2, GPLv2+, BSD
watchdog *watchdog*	5.15	GPL-2.0+
watchdog-config *watchdog-config*	1.0	MIT-X
wireless-regdb *wireless-regdb-static*	2019.06.03	ISC
wpa-supplicant *wpa-supplicant* *wpa-supplicant-passphrase*	2.9 *wpa-supplicant-cli*	BSD
xz *liblzma*	5.2.4	PD
zip *zip*	3.0	BSD-3-Clause
zlib *zlib*	1.2.11	Zlib
zsh *zsh*	5.4.2	zsh
zsh-config *zsh-config*	1.0	MIT