FSL Community BSP Release Notes Documentation

Release 2.1 (Draft document)

FSL Community BSP Team

April 15, 2016

CONTENTS

This document is the release notes for the FSL Community BSP 2.1 (Draft document), which is the result of a community effort to improve Freescale's SoC support for OpenEmbedded and Yocto Project.

> **Warning:** This document is still in **draft** form and *shouldn't be considered finished*. In case you wish to contribute with suggestions, fixes or comments, then please get in touch through the meta-freescale mailing list.

If you want to make part of FSL Community BSP access http://freescale.github.io and find links to this document, how to contribute, and how to download both the source code and several pre-built images.

CHAPTER

ONE

DEFINING THE FSL COMMUNITY BSP

The FSL Community BSP is a community-driven project to provide and maintain Board Support Package (BSP) metadata layers for use in OpenEmbedded and Yocto Project with Freescale's SoCs.

The FSL Community BSP follows Yocto Project's *release schedule* and *branch naming* (since release 1.3, denzil).

See the Yocto Project Release for details on the Yocto Project.

1.1 Motivation

The FSL Community BSP started with the goal of easing the use of OpenEmbedded and Yocto Project with Freescale's SoCs and providing an example of how to assemble an easy-to-use platform as the basis for future products.

The FSL Community BSP provides:

- common environment configuration;
- multiple download layers with the use of repo;
- common location for discussing Freescale SoCs, kernels, bootloaders, user space packages, (BSP in general), bugs, how-tos, and so on

1.2 What the FSL Community BSP is not

The FSL Community BSP does not have a paid support team. The members of this community have full-time jobs and work on the project in their spare time. Most of them are working with Freescale SoCs in their full-time job, so it means some of them can provide paid support if requested.

The provided source code is not intended to be a product in itself. It is a reference platform for people to build products with. Because of this, plan to have a development and test cycle for your product if you decide to base it on the FSL Community BSP.

The project is community-driven work, and it is NOT an official Freescale support channel.

1.3 What you can expect

- You can expect help when you post a question, but please be patient. Wait for at least two days for a response. Most of the time, people do reply when they know an answer or have advice to offer. If you don't receive a reply, then it may be due to no one in the community having an adequate response.
- The stable branch is supported for six months after the release date (following the Yocto Project's release schedule);
- The upstreaming takes place as quickly as possible and any needed adjustment is going to be made accordingly.

1.4 What the community expects from you

The community does expect that you contribute back by:

- replying when you know the answer to a question in the mailing list;
- reviewing the patches sent to mailing list;
- testing new patches that affect you directly or indirectly;
- reporting bugs you may find;
- upstreaming bug fixes;
- upstreaming features that may be good for the community.

CHAPTER

TWO

UPSTREAMING

The FSL Community BSP provides test images and demos in addition to the base BSP for Freescale reference boards and third-party boards. In addition to the BSP, a Linux-based operating system typically requires several other packages, such as ssh client/server, window managers, applications, and so on. These packages are not part of the BSP. In other words, the FSL Community BSP is used with applications, tools and metadata from other projects, such as OpenEmbedded and Poky.

The FSL Community BSP always offers a stable version and a development version. You may face errors that are not caused by FSL Community BSP's layers but instead by OpenEmbedded's or Poky's metadata. In this case, the error must be fixed in its layer.

The following image shows the upstream levels:

2.1 Main branch names

- master-next: this branch is used to keep the patches to be built by the autobuilder for the very first test build. Do not expect to have a clear merging schedule, or to have a stable project when working with the master-next branch;
- master: this is the branch where development takes place. Any new feature or bug fix must be merged here first. This is the development of the next stable branch;
- krogoth: the latest stable branch. This branch only accepts bug fixes, and is supported for 6 months after the release date.

There are other branches available, and they are the previous stable branches. They are kept online for users' convenience, and you should not expect backports or bug fixes.

2.2 Upstream cycle

In addition to the normal Yocto Project upstream process, there is also a BSP upstream cycle.

The BSP upstream cycle starts just after a Freescale Official Release is published in git.freescale.com. The patches to adapt the recipes from **meta-fsl-bsp-release** are sent out for review to the **meta-freescale** mailing list and are merged in the **meta-fsl-arm** and **meta-fsl-demos** layers or upstreamed to Yocto Project accordingly.

A more detailed step-by-step process is shown below:

1. New Freescale Official Release is published;
2. The patches are sent to **meta-freescale**;
3. After the review process, the patches are merged in the proper layer's *master-next* branch;
4. Source code is built by the autobuilder;
5. After one week in *master-next*, it is merged in *master*;
6. Freescale internally bases the next Freescale Official Release from the community source code;
7. Back to step 1.

The result is that Freescale uses the FSL Community BSP source code with its bug fixes, improvements, and any new features to create the *next* Freescale Official Release.

Freescale uses the latest stable branch from Yocto Project to base the *next* Freescale Official Release. When this release is published, it is rebased and reworked to be merged in the current development branch.

CHAPTER

THREE

THE DIFFERENCES BETWEEN FSL COMMUNITY BSP AND FREESCALE OFFICIAL RELEASE

The goal for each project is different. See below for the main points of divergence.

3.1 Freescale Official Release

The Freescale Official Release is intended to provide a static base for Freescale to test and validate the BSP modules with Freescale evaluation boards, and it is developed internally by Freescale. The set of supported boards vary from release to release and is listed in the Freescale Official Release notes for the specific version. The release points to a static revision of every included layer. Therefore, the release does not receive updates and bug fixes.

3.2 FSL Community BSP

The FSL Community BSP is a reference system that can be used as a base for products and is an open project that accepts contributions from the community. It supports a wide range of boards which range from Freescale evaluation boards (**meta-fsl-arm** layer) to third-party boards (**meta-fsl-arm-extra**). The release is a "*moving target*", so there are updates on top of the released source code, such as the addition of new features and bug fixes.

Table 3.1: Comparative between Freescale Official Release and FSL Community BSP

	Freescale Official Release	FSL Community BSP
Intended use	Reference system for BSP modules test and validation on Freescale Reference Boards	Reference system for use as base for any project for all supported boards
Code	Static. Only include any bug fixes on the upcoming release	Updates. Receives bug fixes and has security issues fixed often
Contribution	Indirect contribution via FSL Community BSP. After revision, contribution may be merged in upcoming release	Open, everyone is welcome to contribute to the project
Board Support	Limited, as it supports just the Freescale evaluation boards listed in the Release Notes	Extended, as it supports both Freescale evaluation boards and 3rd party boards. See *Supported Board List*
Yocto Project Compatible	No	Yes
Support	i.MX Community	meta-freescale
Repository	git.freescale.com	github.com/Freescale

CHAPTER

FOUR

FSL COMMUNITY BSP SCOPE

The scope of the FSL Community BSP includes the meta layers:

- meta-fsl-arm: provides the base support and Freescale ARM reference boards;
- meta-fsl-arm-extra: provides support for 3rd party and partner boards;
- meta-fsl-demos: provides images recipes, demo recipes, and packagegroups used to easy the development with Yocto Project.
- Documentation: provides the source code for FSL Community BSP Release Notes (RN), User Guide (UG) and Frequently Asked Questions (FAQ)

4.1 License

The FSL Community BSP is a project with the same licensing of most Yocto Project layers. It means the recipe file is under a certain license, and the source code used by that recipe is under another certain license (being it equal or not).

Most of FSL Community BSP's metadata is under MIT license, however the extensive and accurate list of package's license provided by the Yocto Project's metadata can be generated with few commands, for detailed information on how license is handled by Yocto Project see the Reference Manual.

4.1.1 End User License Agreement (EULA)

Freescale releases basically two kind of packages, the open sourced packages use regular open source licenses (GPLv2 for example).

The close sourced packages are released under the Freescale License (known as EULA). Each package has a copy of EULA inside itself and a copy of the EULA text is also included inside **meta-fsl-arm** root dir (`sources/meta-fsl-arm/EULA`).

The FSL Community BSP handles the EULA acceptance by prompting user to read and accept EULA text at the very first environment setup. It is user's duty to read and understand it before accepting it. After it is accepted the first time, it is assumed accepted in any other build.

4.2 Kernel Release Notes

The FSL Community BSP includes support for several kernel providers. Each machine may have a different Linux Kernel provider.

The FSL Community BSP is not responsible for the content of those kernels. Although we *as community* should feel empowered to submit bug fixes and new features for those projects.

See the respective Linux Kernel provider for your machine in section *Linux Kernel*.

4.3 Different Product SoC Families

Currently, the FSL Community BSP includes the following Product SoC Families:

- **i.MX Application Processors (imx)**: Regarding the i.MX Freescale Page: *i.MX applications processors are multicore ARM®-based solutions for multimedia and display applications with scalability, high performance, and low power capabilities.*
- **Vybrid Controller Solutions based on ARM® Cores (vybrid)**: Regarding the Vybrid Freescale Page: *Vybrid controller solutions are built on an asymmetrical-multiprocessing architecture using ARM® cores as the anchor for the platform, and are ideal for many industrial applications.*
- **Layerscape Architecture (ls)**: Regarding the Layerscape Freescale Page: *delivers unprecedented efficiency and scale for the smarter, more capable networks of tomorrow.*

Freescale groups a set of SoCs which target different markets in product families. Those are grouped according to their SoC features and internal hardware capabilities.

The Yocto Project's tools have the required capabilities to differentiate the architectures and BSP components for the different SoC families. In this perspective, the FSL Community BSP can support a wide range of architectures and product lines which go across several markets.

For the FSL Community BSP, the different SoCs, from all product lines manufactured by Freescale, can be seen as different machines, thus easing the use of same architecture across different markets.

4.4 Supported Board List

Please, see the next table for the complete supported board list.

Table 4.1: Supported machines in FSL Community BSP

Machine	Name	SoC	Layer
apalis-imx6	Toradex Apalis iMX6Q/D	i.MX6	meta-fsl-arm-extra
cfa10036	Crystalfontz CFA-10036	i.MX28	meta-fsl-arm-extra
cfa10037	Crystalfontz CFA-10037	i.MX28	meta-fsl-arm-extra
cfa10049	Crystalfontz CFA-10049	i.MX28	meta-fsl-arm-extra

Continued on next page

Table 4.1 – continued from previous page

Machine	Name	SoC	Layer
cfa10055	Crystalfontz CFA-10055	i.MX28	meta-fsl-arm-extra
cfa10056	Crystalfontz CFA-10056	i.MX28	meta-fsl-arm-extra
cfa10057	Crystalfontz CFA-10057	i.MX28	meta-fsl-arm-extra
cfa10058	Crystalfontz CFA-10058	i.MX28	meta-fsl-arm-extra
cgtqmx6	Congatec QMX6 Evaluation board	i.MX6 Q/DL	meta-fsl-arm-extra
cm-fx6	CompuLab CM-FX6	i.MX6 Q/DL	meta-fsl-arm-extra
colibri-imx6	Toradex Colibri iMX6DL/S	i.MX6 DL/S	meta-fsl-arm-extra
colibri-vf	Toradex Colibri VF50/VF61	VF500/VF610	meta-fsl-arm-extra
cubox-i	SolidRun CuBox-i and HummingBoard	i.MX6 Q/DL	meta-fsl-arm-extra
imx233-olinuxino-maxi	OLIMEX iMX233-OLinuXino-Maxi	i.MX23	meta-fsl-arm-extra
imx233-olinuxino-micro	OLIMEX iMX233-OLinuXino-Micro	i.MX23	meta-fsl-arm-extra
imx233-olinuxino-mini	OLIMEX iMX233-OLinuXino-Mini	i.MX23	meta-fsl-arm-extra
imx233-olinuxino-nano	OLIMEX iMX233-OLinuXino-Nano	i.MX23	meta-fsl-arm-extra
imx23evk	Freescale i.MX23 Evaluation Kit	i.MX23	meta-fsl-arm
imx28evk	Freescale i.MX28 Evaluation Kit	i.MX28	meta-fsl-arm
imx51evk	Freescale i.MX51 Evaluation Kit	i.MX51	meta-fsl-arm
imx53ard	Freescale i.MX53 SABRE Automotive Board	i.MX53	meta-fsl-arm
imx53qsb	Freescale i.MX53 Quick Start Board	i.MX53	meta-fsl-arm
imx6dl-riotboard	RIoTboard	i.MX6S	meta-fsl-arm-extra
imx6dlsabreauto	Freescale i.MX6DL SABRE Automotive	i.MX6DL	meta-fsl-arm
imx6dlsabresd	Freescale i.MX6DL SABRE Smart Device	i.MX6DL	meta-fsl-arm
imx6qdl-variscite-som	Variscite i.MX6Q/DL VAR-SOM-MX6	i.MX6Q/DL	meta-fsl-arm-extra
imx6qpsabreauto	Freescale i.MX6Q Plus SABRE Automotive	i.MX6QP	meta-fsl-arm
imx6qpsabresd	Freescale i.MX6Q Plus SABRE Smart Device	i.MX6QP	meta-fsl-arm
imx6qsabreauto	Freescale i.MX6Q SABRE Automotive	i.MX6Q	meta-fsl-arm

Continued on next page

Table 4.1 – continued from previous page

Machine	Name	SoC	Layer
imx6qsabrelite	Boundary Devices i.MX6Q SABRE Lite	i.MX6Q	meta-fsl-arm-extra
imx6qsabresd	Freescale i.MX6Q SABRE Smart Device	i.MX6Q	meta-fsl-arm
imx6sl-warp	WaRP	i.MX6SL	meta-fsl-arm-extra
imx6slevk	Freescale i.MX6SL Evaluation Kit	i.MX6SL	meta-fsl-arm
imx6solosabreauto	Freescale i.MX6Solo SABRE Automotive	i.MX6S	meta-fsl-arm
imx6solosabresd	Freescale i.MX6Solo SABRE Smart Device	i.MX6S	meta-fsl-arm
imx6sxsabreauto	Freescale i.MX6SoloX Sabre Automotive	i.MX6SX	meta-fsl-arm
imx6sxsabresd	Freescale i.MX6SoloX SabreSD	i.MX6SX	meta-fsl-arm
imx6ulevk	Freescale i.MX6UL Evaluation Kit	i.MX6UL	meta-fsl-arm
imx7d-warp7	WaRP7	i.MX7S	meta-fsl-arm-extra
imx7dsabresd	Freescale i.MX7D SABRE Smart Device	i.MX7D	meta-fsl-arm
ls1021atwr	Freescale LS1021ATWR board	ls102xa	meta-fsl-arm
m28evk	DENX M28 SoM Evaluation Kit	i.MX28	meta-fsl-arm-extra
m53evk	DENX M53 SoM Evaluation Kit	i.MX53	meta-fsl-arm-extra
nitrogen6sx	Boundary Devices Nitrogen6SX	i.MX6SX	meta-fsl-arm-extra
nitrogen6x	Boundary Devices Nitrogen6X	i.MX6 Q/DL	meta-fsl-arm-extra
nitrogen6x-lite	Boundary Devices Nitrogen6X Lite	i.MX6S	meta-fsl-arm-extra
nitrogen7	Boundary Devices Nitrogen7	i.MX7D	meta-fsl-arm-extra
pcm052	Phytec phyCORE Vybrid Development Kit	vf60	meta-fsl-arm-extra
twr-vf65gs10	Freescale Vybrid TWR-VF65GS10	VF610	meta-fsl-arm
tx6q-10x0	Ka-Ro electronics i.MX6Q TX6Q Computer-On-Module	i.MX6Q	meta-fsl-arm-extra
tx6q-11x0	Ka-Ro electronics i.MX6Q TX6Q Computer-On-Module	i.MX6Q	meta-fsl-arm-extra
tx6s-8034	Ka-Ro electronics i.MX6S TX6S Computer-On-Module	i.MX6S	meta-fsl-arm-extra

Continued on next page

Table 4.1 – continued from previous page

Machine	Name	SoC	Layer
tx6s-8035	Ka-Ro electronics i.MX6S TX6S Computer-On-Module	i.MX6S	meta-fsl-arm-extra
tx6u-8033	Ka-Ro electronics i.MX6DL TX6DL Computer-On-Module	i.MX6DL	meta-fsl-arm-extra
tx6u-80x0	Ka-Ro electronics i.MX6DL TX6DL Computer-On-Module	i.MX6DL	meta-fsl-arm-extra
tx6u-81x0	Ka-Ro electronics i.MX6DL TX6DL Computer-On-Module	i.MX6DL	meta-fsl-arm-extra
ventana	i.MX6Q/DL Ventana Platform	i.MX6Q/DL	meta-fsl-arm-extra
wandboard	Wandboard i.MX6 Wandboard Quad/Dual/Solo	i.MX6Q/DL	meta-fsl-arm-extra

4.4.1 Machine Maintainers

Since FSL Community BSP Release 1.6 (Daisy), the maintainer field in machine configuration files of **meta-fsl-arm** and **meta-fsl-arm-extra** is mandatory for any new board to be added.

So now on, every new board must have someone assigned as maintainer. This ensures, in long term, all boards with a maintainer assigned. Current orphan boards are not going to be removed unless it causes maintenance problem and the fix is not straightforward.

The maintainer duties:

- The one with casting vote when a deadlock is faced.
- Responsible to keep that machine working (that means, booting and with some stability) Keep kernel, u-boot updated/tested/working.
- Keep release notes updated
- Keep test cycle updated
- Keep the most usual images building and booting

When a build error is detected, the maintainer will “fix” it. For those maintainers with kernel control (meta-fsl-arm-extra), it is expected that they properly fix the kernel issue (when it’s a kernel issue). However, anything out of community control should be worked around anyway.

Machines with maintainers

Table 4.2: Machines with maintainers

Machine	Name
apalis-imx6	Toradex Apalis iMX6Q/D
cfa10036	Crystalfontz CFA-10036
cfa10037	Crystalfontz CFA-10037
cfa10049	Crystalfontz CFA-10049
cfa10055	Crystalfontz CFA-10055
cfa10056	Crystalfontz CFA-10056
cfa10057	Crystalfontz CFA-10057
cfa10058	Crystalfontz CFA-10058
cgtqmx6	Congatec QMX6 Evaluation board
cm-fx6	CompuLab CM-FX6
colibri-imx6	Toradex Colibri iMX6DL/S
colibri-vf	Toradex Colibri VF50/VF61
cubox-i	SolidRun CuBox-i and HummingBoard
imx23evk	Freescale i.MX23 Evaluation Kit
imx28evk	Freescale i.MX28 Evaluation Kit
imx51evk	Freescale i.MX51 Evaluation Kit
imx53ard	Freescale i.MX53 SABRE Automotive Board
imx53qsb	Freescale i.MX53 Quick Start Board
imx6dl-riotboard	RIoTboard
imx6dlsabreauto	Freescale i.MX6DL SABRE Automotive
imx6dlsabresd	Freescale i.MX6DL SABRE Smart Device
imx6qdl-variscite-som	Variscite i.MX6Q/DL VAR-SOM-MX6
imx6qpsabreauto	Freescale i.MX6Q Plus SABRE Automotive
imx6qpsabresd	Freescale i.MX6Q Plus SABRE Smart Device
imx6qsabreauto	Freescale i.MX6Q SABRE Automotive
imx6qsabrelite	Boundary Devices i.MX6Q SABRE Lite
imx6qsabresd	Freescale i.MX6Q SABRE Smart Device
imx6sl-warp	WaRP
imx6slevk	Freescale i.MX6SL Evaluation Kit
imx6solosabresd	Freescale i.MX6Solo SABRE Smart Device
imx6sxsabreauto	Freescale i.MX6SoloX Sabre Automotive
imx6sxsabresd	Freescale i.MX6SoloX SabreSD
imx6ulevk	Freescale i.MX6UL Evaluation Kit
imx7d-warp7	WaRP7
imx7dsabresd	Freescale i.MX7D SABRE Smart Device
ls1021atwr	Freescale LS1021ATWR board
nitrogen6sx	Boundary Devices Nitrogen6SX
nitrogen6x	Boundary Devices Nitrogen6X
nitrogen6x-lite	Boundary Devices Nitrogen6X Lite
nitrogen7	Boundary Devices Nitrogen7
pcm052	Phytec phyCORE Vybrid Development Kit
twr-vf65gs10	Freescale Vybrid TWR-VF65GS10

Continued on next page

Table 4.2 – continued from previous page

Machine	Name
tx6q-10x0	Ka-Ro electronics i.MX6Q TX6Q Computer-On-Module
tx6s-8034	Ka-Ro electronics i.MX6S TX6S Computer-On-Module
tx6u-80x0	Ka-Ro electronics i.MX6DL TX6DL Computer-On-Module
ventana	i.MX6Q/DL Ventana Platform
wandboard	Wandboard i.MX6 Wandboard Quad/Dual/Solo

Machines without a maintainer

Table 4.3: Machines without a maintainer

Machine	Name
imx233-olinuxino-maxi	OLIMEX iMX233-OLinuXino-Maxi
imx233-olinuxino-micro	OLIMEX iMX233-OLinuXino-Micro
imx233-olinuxino-mini	OLIMEX iMX233-OLinuXino-Mini
imx233-olinuxino-nano	OLIMEX iMX233-OLinuXino-Nano
imx6solosabreauto	Freescale i.MX6Solo SABRE Automotive
m28evk	DENX M28 SoM Evaluation Kit
m53evk	DENX M53 SoM Evaluation Kit

CHAPTER

FIVE

SOFTWARE ARCHITECTURE

5.1 SoC Hierarchy

The following tree shows the SoC hierarchy:

5.2 Linux Kernel

FSL Community BSP supports the following sources for Linux Kernel:

- **linux-boundary**: Linux kernel for Boundary Devices boards.
- **linux-cfa**: Linux kernel for Crystalfontz boards.
- **linux-compulab**: Linux kernel for CompuLab cm-fx6 boards.
- **linux-congatec**: linux-congatec version 3.14-r0.
- **linux-denx**: DENX mainline based Linux kernel.
- **linux-fslc**: Linux kernel based on mainline kernel used by FSL Community BSP in order to provide support for some backported features and fixes, or because it was applied in linux-next and takes some time to become part of a stable version, or because it is not applicable for upstreaming.
- **linux-fslc-imx**: Linux kernel based on Freescale 3.14.52-1.1.0 GA release, used by FSL Community BSP in order to provide support for i.MX6 based platforms and include official Linux kernel stable updates, backported features and fixes coming from the vendors, kernel community or FSL Community itself.
- **linux-gateworks-imx**: linux-gateworks-imx version 3.14-r0.
- **linux-imx**: Linux Kernel provided and supported by Freescale with focus on i.MX Family Reference Boards. It includes support for many IPs such as GPU, VPU and IPU.
- **linux-karo**: Linux Kernel for Ka-Ro electronics TX Computer-On-Modules.
- **linux-ls1**: Linux Kernel provided and supported by Freescale with focus on Layerscape1 Family Boards.
- **linux-timesys**: Linux Kernel with added drivers and board support for Vybrid-based platforms.

- **linux-toradex**: Linux kernel for Toradex Colibri VFxx Computer on Modules.
- **linux-variscite**: linux-variscite version 3.14.28-r0.
- **linux-wandboard**: Linux kernel for Wandboard.

As stated in *Kernel Release Notes*, FSL Community BSP is not responsible for the Linux Kernel content in any kernel provider. If you are looking for the feature list, supported devices, official way to get a support channel or how to report bug, please, see above where to get help, for each kernel provider.

- **linux-imx**: provider, Freescale has a release notes document for each version released. This document has a list of known issues, new features, list of kernel arguments, and the linux-imx kernel scope for each Freescale Reference Board. This document is present into the Document Bundle provided by Freescale.

5.2.1 Default Linux Providers

The following table shows the default version of Linux Kernel provided by FSL Community BSP for each supported machine.

Table 5.1: Default Linux kernel version for each supported machine

Board	Kernel Provider	Kernel Version
apalis-imx6	linux-toradex	3.14.28-v2.5b2
cfa10036	linux-cfa	4.1.13
cfa10037	linux-cfa	4.1.13
cfa10049	linux-cfa	4.1.13
cfa10055	linux-cfa	4.1.13
cfa10056	linux-cfa	4.1.13
cfa10057	linux-cfa	4.1.13
cfa10058	linux-cfa	4.1.13
cgtqmx6	linux-congatec	3.14-1.0.x-mx6-qmx6
cm-fx6	linux-compulab	3.14.28-cm-fx6
colibri-imx6	linux-toradex	3.14.28-v2.5b2
colibri-vf	linux-toradex	4.1-v2.5b3
cubox-i	linux-fslc	4.4+git
imx233-olinuxino-maxi	linux-fslc	4.4+git
imx233-olinuxino-micro	linux-fslc	4.4+git
imx233-olinuxino-mini	linux-fslc	4.4+git
imx233-olinuxino-nano	linux-fslc	4.4+git
imx23evk	linux-fslc	4.4+git
imx28evk	linux-fslc	4.4+git
imx51evk	linux-fslc	4.4+git
imx53ard	linux-fslc	4.4+git
imx53qsb	linux-fslc	4.4+git
imx6dl-riotboard	linux-fslc	4.4+git

Continued on next page

Table 5.1 – continued from previous page

Board	Kernel Provider	Kernel Version
imx6dlsabreauto	linux-fslc-imx	4.1-1.0.x+git
imx6dlsabresd	linux-fslc-imx	4.1-1.0.x+git
imx6qdl-variscite-som	linux-variscite	3.14.28-1.1.0
imx6qpsabreauto	linux-imx	4.1.15-1.0.0
imx6qpsabresd	linux-fslc-imx	4.1-1.0.x+git
imx6qsabreauto	linux-fslc-imx	4.1-1.0.x+git
imx6qsabrelite	linux-boundary	3.14.52-1.1.0_ga+yocto
imx6qsabresd	linux-fslc-imx	4.1-1.0.x+git
imx6sl-warp	linux-fslc	4.4+git
imx6slevk	linux-fslc-imx	4.1-1.0.x+git
imx6solosabreauto	linux-fslc-imx	4.1-1.0.x+git
imx6solosabresd	linux-fslc-imx	4.1-1.0.x+git
imx6sxsabreauto	linux-fslc-imx	4.1-1.0.x+git
imx6sxsabresd	linux-fslc-imx	4.1-1.0.x+git
imx6ulevk	linux-imx	4.1.15-1.0.0
imx7d-warp7	linux-fslc-imx	4.1-1.0.x+git
imx7dsabresd	linux-imx	4.1.15-1.0.0
ls1021atwr	linux-ls1	3.12+ls1
m28evk	linux-fslc	4.4+git
m53evk	linux-denx	3.9-master
nitrogen6sx	linux-boundary	3.14.52-1.1.0_ga+yocto
nitrogen6x	linux-boundary	3.14.52-1.1.0_ga+yocto
nitrogen6x-lite	linux-boundary	3.14.52-1.1.0_ga+yocto
nitrogen7	linux-boundary	3.14.52-1.1.0_ga+yocto
pcm052	linux-timesys	3.13
twr-vf65gs10	linux-fslc	4.4+git
tx6q-10x0	linux-karo	3.16-2015-09-18
tx6q-11x0	linux-karo	3.16-2015-09-18
tx6s-8034	linux-karo	3.16-2015-09-18
tx6s-8035	linux-karo	3.16-2015-09-18
tx6u-8033	linux-karo	3.16-2015-09-18
tx6u-80x0	linux-karo	3.16-2015-09-18
tx6u-81x0	linux-karo	3.16-2015-09-18
ventana	linux-gateworks-imx	3.14-1.0.x_ga+yocto
wandboard	linux-wandboard	3.14.28_1.0.0_ga-wandboard

5.3 Bootloaders

FSL Community BSP supports barebox and u-boot as bootloaders.

- **barebox**: Barebox - a bootloader that inherits the best of U-Boot and the Linux kernel
- **u-boot-boundary**: u-boot for Boundary Devices boards.

- **u-boot-fslc**: U-Boot based on mainline U-Boot used by FSL Community BSP in order to provide support for some backported features and fixes, or because it was submitted for revision and it takes some time to become part of a stable version, or because it is not applicable for upstreaming.
- **u-boot-imx**: U-Boot provided by Freescale with focus on i.MX reference boards.
- **u-boot-karo**: u-boot for Ka-Ro electronics TX Computer-On-Modules.
- **u-boot-ls1**: U-Boot provided by Freescale with focus on QorIQ Layerscape1 boards
- **u-boot-toradex**: U-Boot bootloader with support for Toradex Computer on Modules.
- **u-boot-variscite**: U-Boot for Variscite i.MX6Q/DL VAR-SOM-MX6.

The following table shows the default bootloaders (and their versions) for the supported boards.

Table 5.2: Default bootloader version for each supported machine

Board	Bootloader	Bootloader version
apalis-imx6	u-boot-toradex	v2015.04-v2.5b3+git
cfa10036	barebox	2015.10.0
cfa10037	barebox	2015.10.0
cfa10049	barebox	2015.10.0
cfa10055	barebox	2015.10.0
cfa10056	barebox	2015.10.0
cfa10057	barebox	2015.10.0
cfa10058	barebox	2015.10.0
cgtqmx6	u-boot-fslc	v2016.03+git
cm-fx6	u-boot-fslc	v2016.03+git
colibri-imx6	u-boot-toradex	v2015.04-v2.5b3+git
colibri-vf	u-boot-toradex	v2015.04-v2.5b3+git
cubox-i	u-boot-fslc	v2016.03+git
imx233-olinuxino-maxi	u-boot-fslc	v2016.03+git
imx233-olinuxino-micro	u-boot-fslc	v2016.03+git
imx233-olinuxino-mini	u-boot-fslc	v2016.03+git
imx233-olinuxino-nano	u-boot-fslc	v2016.03+git
imx23evk	u-boot-fslc	v2016.03+git
imx28evk	u-boot-fslc	v2016.03+git
imx51evk	u-boot-fslc	v2016.03+git
imx53ard	u-boot-fslc	v2016.03+git
imx53qsb	u-boot-fslc	v2016.03+git
imx6dl-riotboard	u-boot-fslc	v2016.03+git
imx6dlsabreauto	u-boot-fslc	v2016.03+git
imx6dlsabresd	u-boot-fslc	v2016.03+git
imx6qdl-variscite-som	u-boot-variscite	2013.10
imx6qpsabreauto	u-boot-imx	2015.04-imx_v2015.04_4.1.15_1.0.0_ga
imx6qpsabresd	u-boot-fslc	v2016.03+git

Continued on next page

Table 5.2 – continued from previous page

Board	Bootloader	Bootloader version
imx6qsabreauto	u-boot-fslc	v2016.03+git
imx6qsabrelite	u-boot-boundary	v2016.03+git
imx6qsabresd	u-boot-fslc	v2016.03+git
imx6sl-warp	u-boot-fslc	v2016.03+git
imx6slevk	u-boot-fslc	v2016.03+git
imx6solosabreauto	u-boot-imx	2015.04-imx_v2015.04_4.1.15_1.0.0_ga
imx6solosabresd	u-boot-imx	2015.04-imx_v2015.04_4.1.15_1.0.0_ga
imx6sxsabreauto	u-boot-imx	2015.04-imx_v2015.04_4.1.15_1.0.0_ga
imx6sxsabresd	u-boot-fslc	v2016.03+git
imx6ulevk	u-boot-fslc	v2016.03+git
imx7d-warp7	u-boot-fslc	v2016.03+git
imx7dsabresd	u-boot-fslc	v2016.03+git
ls1021atwr	u-boot-ls1	2015.01+ls1
m28evk	u-boot-fslc	v2016.03+git
m53evk	u-boot-fslc	v2016.03+git
nitrogen6sx	u-boot-boundary	v2016.03+git
nitrogen6x	u-boot-boundary	v2016.03+git
nitrogen6x-lite	u-boot-boundary	v2016.03+git
nitrogen7	u-boot-boundary	v2016.03+git
pcm052	u-boot-fslc	v2016.03+git
twr-vf65gs10	u-boot-fslc	v2016.03+git
tx6q-10x0	u-boot-karo	v2015.10-rc2+git
tx6q-11x0	u-boot-karo	v2015.10-rc2+git
tx6s-8034	u-boot-karo	v2015.10-rc2+git
tx6s-8035	u-boot-karo	v2015.10-rc2+git
tx6u-8033	u-boot-karo	v2015.10-rc2+git
tx6u-80x0	u-boot-karo	v2015.10-rc2+git
tx6u-81x0	u-boot-karo	v2015.10-rc2+git
ventana	u-boot-gateworks-imx	v2015.04+git
wandboard	u-boot-fslc	v2016.03+git

5.4 User Space Packages

There is a huge number of user space packages provided by the Yocto Project. The following table shows some version for few highlighted packages.

Table 5.3: Main user space package versions

Package	Board/SoC Family	Version
gstreamer1.0	All	1.6.3
udev	All	3.1.5

5.4.1 Freescale User Space Packages

This section shows the version package for each board. Those packages provide hardware acceleration for GPU or VPU, hardware optimization or some hardware test tools.

- **Hardware acceleration** is achieved using a different core for processing some specific task. In this case, GPU or VPU.
- **Hardware optimization** is achieved with some changes in source code in order to get a better performance for a specific task on a specific hardware. For example, audio decode made by software, but with optimizations for ARM.
- **Hardware-specific** is applicable when the package was designed to be executed on a specific hardware, and it does not make sense on other hardware. For example, imx-test is a test package for imx boards. It can be cross-compiled for any other core, although it will only behave as expect if executed on imx boards.

The package version and variety varies on *SoC Hierarchy*. For example, machines with i.MX28 SoC does not have VPU, the recipe imx-vpu is not needed. There are differences, as well, in GPU support recipes.

Version by *SoC Hierarchy*

The following table shows the version of each package depending on the *SoC Hierarchy*.

Table 5.4: User space package version by SoC hierarchy

Package name	ls102xa	mx28	mx5	mx6q / mx6dl	mx6sl	vf60
apptrk	git	git	git	git	git	git
cst	git	git	git	git	git	git
devregs	1.0+AUTOINC+34ed402b92	1.0+AUTOINC+34ed402b92	1.0+AUTOINC+34ed402b92	1.0+AUTOINC+34ed402b92	1.0+AUTOINC+34ed402b92	1.0+AUTOINC+34ed402b92
directfb	1.7.7	1.7.7	1.7.7	1.7.7	1.7.7	1.7.7
directfb-examples	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0	1.7.0
elftosb	10.12.01	10.12.01	10.12.01	10.12.01	10.12.01	10.12.01
firmware-imx	–	–	5.4	5.4	5.4	–
fsl-alsa-plugins	–	–	–	1.0.25	1.0.25	–
gpu-viv-bin-mx6q	–	–	–	–	–	–
gpu-viv-g2d	–	–	–	–	–	–
gst1.0-fsl-plugin	–	–	–	4.0.8	4.0.8	–
gstreamer1.0-plugins-imx	–	–	–	0.12.1	0.12.1	–
imx-kobs	–	–	–	–	–	–
imx-lib	–	–	–	5.4	5.4	–
imx-test	–	00.00.00	00.00.00	5.4	5.4	00.00.00
imx-uuc	0.5.1	0.5.1	0.5.1	0.5.1	0.5.1	0.5.1
imx-vpu	–	–	–	5.4.33	5.4.33	–
libf-slcodec	–	–	–	4.0.8	4.0.8	–
libf-slparser	–	–	–	4.0.8	4.0.8	–
libfs-lvpuwrap	–	–	–	1.0.62	–	–
libmcc	–	–	–	–	–	1.05.1
mqxboot	–	–	–	–	–	2.0.1
mxsldr	0.0.0+git	0.0.0+git	0.0.0+git	0.0.0+git	0.0.0+git	0.0.0+git
qe-ucode	git	–	–	–	–	–
qemu-fsl	2.2.0+AUTOINC+00ac004143	2.2.0+AUTOINC+00ac004143	2.2.0+AUTOINC+00ac004143	2.2.0+AUTOINC+00ac004143	2.2.0+AUTOINC+00ac004143	2.2.0+AUTOINC+00ac004143
rcw	git	–	–	–	–	–
xf86-video-imxfb	–	–	–	–	–	–
xf86-video-imxfb-vivante	–	–	–	5.0.11.p8.3	5.0.11.p8.3	–

Hardware relation by *SoC Hierarchy*

The following table shows how packages interact with hardware depending on the *SoC Hierarchy*

Table 5.5: Hardware dependent packages

Package Name	mx28	mx5	mx6	vf60
imx-test	HW-specific	HW-specific	HW-specific	–
gst-fsl-plugin	HW-specific	HW-specific	HW-specific	–
libfslcodec	HW optimization	HW acceleration	HW acceleration	–
libfslparser	HW optimization	HW optimization	HW optimization	–
imx-vpu	–	HW acceleration	HW acceleration	–
imx-lib	–	HW acceleration	HW acceleration	–
firmware-imx	–	HW-specific	HW-specific	–
mxsldr	HW-specific	–	–	–
gpu-viv-g2d	–	–	HW acceleration	–
xf86-video-imxfb-vivante	–	–	HW acceleration	–
gpu-viv-bin-mx6q	–	–	HW acceleration	–
directfb	–	–	HW acceleration	–
directfb-examples	–	–	HW acceleration	–
xf86-video-imxfb	–	HW acceleration	–	–
amd-gpu-bin-mx51	–	HW acceleration	–	–
libz160	–	HW acceleration	–	–
amd-gpu-x11-bin-mx51	–	HW acceleration	–	–
libfslvpuwrap	–	–	HW acceleration	–
fsl-alsa-plugins	–	–	HW-specific	–
gstreamer1.0-plugins-imx	–	–	HW acceleration	–
imx-uuc	HW-specific	HW-specific	HW-specific	–
libmcc	–	–	–	
mqxboot	–	–	–	HW-specific

5.5 PackageGroups and Images

The FSL Community BSP provides a list of PACKAGEGROUPS and images intended to ease the initial development of custom applications.

The main goal is not to provide a production solution, on the contrary, it should be seen as an example of package set for a specific IP development, and an example of initial generic development and test images.

5.5.1 PACKAGEGROUPS

The following list shows the current PACKAGEGROUPs available in Krogoth when using FSL Community BSP.

You can understand what a PACKAGEGROUPS is and learn how to use it in Yocto Project Development Manual

- **packagegroup-fsl-gstreamer1.0**: Package group used by FSL Community to provide audio, video, networking and debug GStreamer plugins with the required hardware acceleration (if supported by the SoC).
- **packagegroup-fsl-gstreamer1.0-full**: Package group used by FSL Community to provide all GStreamer plugins from the base, good, and bad packages, as well as the ugly and libav ones if commercial packages are whitelisted, and plugins for the required hardware acceleration (if supported by the SoC).
- **packagegroup-fsl-mfgtool**: Freescale Manufacturing Tool requirements.
- **packagegroup-fsl-tools-benchmark**: Package group used by FSL Community to provide a set of benchmark applications.
- **packagegroup-fsl-tools-gpu**: Package group used by FSL Community to add the packages which provide GPU support.
- **packagegroup-fsl-tools-gpu-external**: Package group used by FSL Community to provide graphic packages used to test the several hardware accelerated graphics APIs including packages not provided by Freescale.
- **packagegroup-fsl-tools-testapps**: Packagegroup used by FSL Community to provide a set of packages and utilities for hardware test.

5.5.2 Images

The following images are provided by FSL Community BSP only. See the list of Yocto Project's reference images in Yocto Project Reference Manual

- **fsl-image-machine-test**: A console-only image that includes gstreamer packages, Freescale's multimedia packages (VPU and GPU) when available, and test and benchmark applications.
- **fsl-image-mfgtool-initramfs**: Small image to be used with Manufacturing Tool (mfg-tool) in a production environment.
- **fsl-image-multimedia**: A console-only image that includes gstreamer packages and Freescale's multimedia packages (VPU and GPU) when available for the specific machine.
- **fsl-image-multimedia-full**: A console-only image that includes gstreamer packages and Freescale's multimedia packages (VPU and GPU) when available for the specific machine.

CHAPTER

SIX

TEST RESULTS

Freescale has a complete test cycle for the BSP released. It includes tests for Linux Kernel for the GPU package and for the VPU package (and all other package needed by the BSP, such as imx-lib).

The results and known issues, from Linux Kernel, GPU and VPU packages can be found in the Freescale Release Notes (Download tab of freescale.com/imx).

For boards from meta-fsl-arm-extra, the test cycle is performed by each mantainer.

CHAPTER

SEVEN

ACKNOWLEDGEMENTS

The FSL BSP Community is a community effort of keeping and mantaining a Freescale boards/chips layer for the Yocto Project.

7.1 Krogoth Source Code

The statistics can be seen at the FSL Community BSP website. It has not been included here as it changes every time bug fixes are included during the maintenance cycle of the release and it would be outdated most of time.

CHAPTER

EIGHT

KNOWN ISSUES

The list of known issues for the FSL Community BSP can be seen at the following URL:

https://bugzilla.yoctoproject.org/buglist.cgi?quicksearch=meta-fsl-arm

It has not been included here as it changes every time bug fixes are included during the maintenance cycle of the release and it would be outdated most of time.