QorIQ LSDK Introduction

QorIQ LSDK is NXP new generation of SDK for Layerscape productions, consists of a set of disaggregated components based on Linux distributions, meets market demand to more Linux distributions of more types, and satisfy the requirement from a wide variety of customers.

In LSDK we use Flexbuild to build all packages from LSDK, make root filesystem and generate the installer.

This document introduces the basic concept of LSDK, comparison between LSDK and Yocto SDK, how to use LSDK, plan and roadmap of LSDK.

1. Basic Concept of LSDK

Many software components of LSDK are available individually, this enables customers and 3rd parties to access them individually so they can integrate them into Linux distributions or systems by themselves.

1.1 LSDK Specific features

LSDK integrates LTS Linux Kernel. LTS(Long Term Support) is used to describe a kernel or Linux distribution that will be formally supported with prompt bug fixes, security updates, and limited feature additions for a defined time period.

LSDK is a complete Linux Kit from a specific provider, includes Kernel, tools, user space, etc.

LSDK provides upstreaming support, the process of adding support for NXP-specific hardware or features to a community (non-NXP) software repository. Users can download LSDK source code and related documents from https://lsdk.github.io/.

1.2 LSDK Components

The basic elements of LSDK is described as the following.

Two key components

Linux kernel – standardized to a stable configuration / revision level

Root file system – containing user space applications and dynamically loadable kernel modules for standard drivers

Commercial distributions usually rely on kernels and user space packages derived from an upstream community-driven distribution (feeder)

E.g. Debian, Slackware,Gentoo

Generally share build tools, package management system, etc. with progenitor

Frequent cross-pollination between feeder and derivative (i.e. not strictly a *fork*); e.g. derivative re-bases off new feeder releases, bug fixes, enhancements submitted upstream

Some commercial distros sponsor community distros; e.g. Red Hat → Fedora, CentOS

All derive from a release branch of the mainline kernel.org Linux kernel development tree

Often distinguished by how closely they track to kernel.org releases

Community distributions typically released more frequently and closer to kernel.org releases

Enterprise distros focus on stability with less frequent releases – based on long-term support “branches”.

1.3 LSDK Images Memory Map

Boards are shipped with the following LSDK images.

NOR image in Bank0/Bank4

Boot image and rootfs on SD card

NOR Image consists of Boot firmware (u-boot), RCW, PHY firmware, DPAA firmware (fman, MC), Minimal busybox rootfs.

Installer to install Ubuntu rootfs on SD consists of Standard distro rootfs and NXP specific user space such as restool, aiop_tool, fmc.

LSDK memory map is as the following.

Region 1	**Region 2**	**Region 3**	**Region 4**	**Region 5**
(4KB)	**(64MB)**	**(20MB)**	**(300MB)**	**(remaining space of disk)**
MBR/GPT	**Firmware** **RCW** **U-boot or UEFI** **Eth PHY firmware** **QE/uQE firmware** **FMan firmware** **MC firmware** **PPA firmware** **kernel image** **DTB** **Ramdisk RFS**	**Partition 1 (FAT32)** **EFI** **BOOTAA64.EFI** **grub.cfg**	**Partition 2** **(EXT4)** **Boot Partition** **Kernel image** **DTBs** **Flex_installer_<arch>.itb** **Distro boot scripts** **Secure headers** **Other**	**Partition 3** **(EXT4)** **rootfs** **Ubuntu** **or** **Ubuntu-Core** **or** **CentOS** **or** **Debian**

2. Comparison Between Layerscape SDK and QorIQ Yocto SDK

The benefit of LSDK can be summarized as the following.

Flexibility, customers need to be able to load whatever distro or run whatever open-source components.

Scalability, customers need to run the same software for low-end and high-end deployments.

Stability, customers need to their development on most recent LTS kernel versions.

Consistency, customers need to be able to move freely between different architecture, x86 or ARM.

The difference between Layerscape SDK and QorIQ Yocto SDK is described as the following table.

	QorIQ SDK	**Layerscape SDK**
Platforms supported	P, B, T – series and LS – series	LS, LX, LA – series
Features	LTS kernel, platform drivers, tools	**Choice of 2** LTS kernels, platform drivers, tools **Available as components too.**
User-space	Yocto	Ubuntu
Build-tool	Yocto	**Ubuntu, make, flexbuild**
Build Environment	Host	Host, **Target**
Boot/recovery options	Flash, network	Flash, network, **SD card, HDD**
Package Installation	Integrate into Yocto, build image, re-flash board.	**Apt-get over network**
Downloadable	Giant ISO with sources and binaries for all platforms	**Individual Binaries,** **Individual components source**

3. How to Usage LSDK

3.1 LSDK Flexbuild Utility

Flexbuild is integrated in LSDK, builds system with flexible system build and distro installation.

The LSDK build system includes three major components: package builder, rootfs maker and image installer.

The utility can run on x86 host of Ubuntu 16.04, arm targets and docker container.

3.2 Build LSDK using Flexbuild

Build LSDK using Flexbuild as the following.

General build command

```
$ tar xvzf flexbuild_<version>.tgz

$ cd flexbuild

$ source setup.env

$ flex-builder -i repo-fetch

$ flex-builder -i repo-tag    (check out  tags specified in file build_lsdk1706.cfg)
```

Build custom kernel and update the boot partition

```
$ flex-builder -c linux  -B menuconfig

$ flex-builder -i uimg

$ flex-builder -i mkbootpartition

$ cd build/qoriq-linux/kernel/arm64/lib && tar cvzf modules.tgz  modules
```

Build custom u-boot or application

```
$ flex-builder -c uboot -m <machine> -b <boottype>      #build uboot for <machine> to generate specified
nor/sd/qspi boot image

$ flex-builder -c <component> -a <arch>                  #build single application component for specified
<arch>
```

3.3 Deploy LSDK Images on the Target Board

Deploy LSDK images from Linux Host

```
$ wget http://www.nxp.com/lgfiles/sdk/lsdk1706/firmware_ls1088ardb_uboot_sdboot.img
```

Or `$ flex-builder -i mkrfs -a <arch> -B additional_packages_list_full`

```
$ flex-installer --bootpart=bootpartition_arm64.tgz --rootfs=build/images/ubuntu_xenial_arm64_rootfs.d --
firmware=firmware_ls1088ardb_uboot_sdboot.img --machine=ls1088ardb --device=/dev/sdX
```

Deploy LSDK images from Target board

Download LSDK composite firmware from NXP website

E.g. `$ wget http://www.nxp.com/lgfiles/sdk/lsdk1706/firmware_ls2088ardb_uboot_norboot.img`

Put LSDK composite firmware to a TFTP server, then download the firmware via TFTP to the target board under the U-Boot prompt

Reset the board and deploy boot partition and Ubuntu 16.04 user land to SD/USB/SATA.

Enable network connection to download LSDK images

Use flex-installer to create and format partitions

```
$ flex-installer -i install --bootpart=bootpartition_arm64.tgz --rootfs=ubuntu_xenial_arm64_rootfs.tgz  --
machine=ls2088ardb --device=usb
```

3.4 Add a Package using Flexbuild

This section introduces how to add a package not official supported by Ubuntu user land during build stage

Add extrinsic package name to extrinsic_packages_list in packages/apt-packages/additional_packages_list

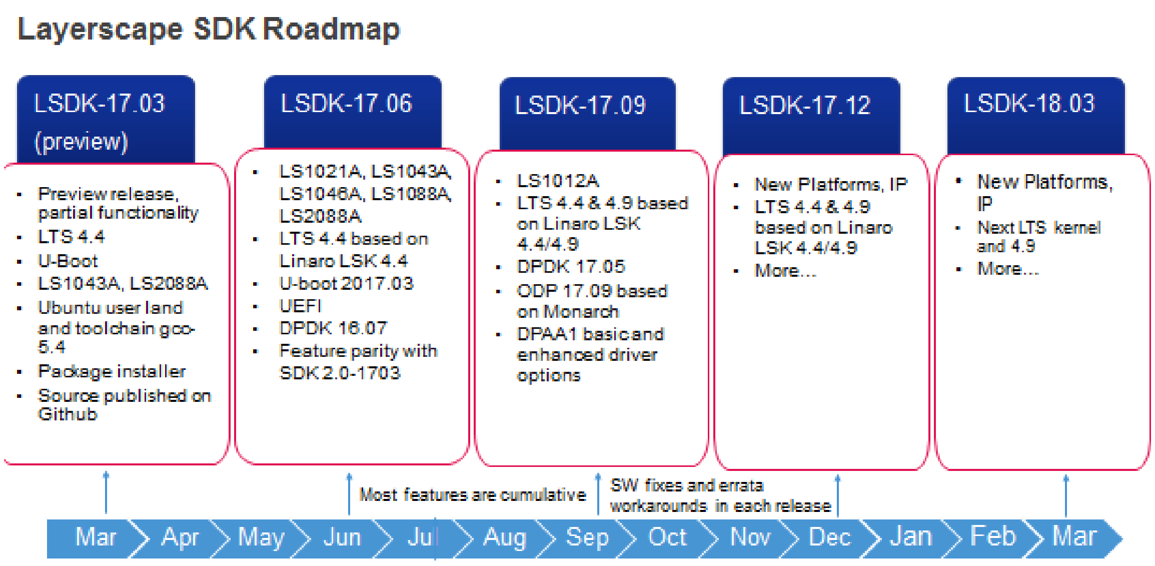
Put custom script of extrinsic package to packages/apt-packages/extrinsic-pkg (e.g. refer to nginx.sh)

Run flex-builder -i mkrfs -a <arch> to generate new Ubuntu rootfs

Install the new Ubuntu rootfs to target machine via flex-installer

4. Layerscape SDK Roadmap

The following is LSDK roadmap of the recent releases.



Layerscape SDK Roadmap

LSDK-17.03 (preview)
- Preview release, partial functionality
- LTS 4.4
- U-Boot
- LS1043A, LS2088A
- Ubuntu user land and toolchain gcc-5.4
- Package installer
- Source published on Github

LSDK-17.06
- LS1021A, LS1043A, LS1046A, LS1088A, LS2088A
- LTS 4.4 based on Linaro LSK 4.4
- U-boot 2017.03
- UEFI
- DPDK 16.07
- Feature parity with SDK 2.0-1703

LSDK-17.09
- LS1012A
- LTS 4.4 & 4.9 based on Linaro LSK 4.4/4.9
- DPDK 17.05
- ODP 17.09 based on Monarch
- DPAA1 basic and enhanced driver options

LSDK-17.12
- New Platforms, IP
- LTS 4.4 & 4.9 based on Linaro LSK 4.4/4.9
- More…

LSDK-18.03
- New Platforms, IP
- Next LTS kernel and 4.9
- More…

Most features are cumulative

SW fixes and errata workarounds in each release