

# NXP Automotive Linux® BSP Product Brief

## Contents

1.	Software Product Overview .....	1
2.	Software Content .....	2
2.1.	Arm Trusted Firmware .....	3
2.2.	U-Boot .....	3
2.3.	Linux Kernel .....	3
2.4.	Root File System.....	4
2.5.	Technologies.....	4
2.6.	Yocto .....	5
3.	Supported Targets .....	6
4.	Quality, Standards Compliance and Testing Approach.....	6
5.	Document Information .....	6

## 1. Software Product Overview

NXP Automotive Linux® BSP follows the general layout of a BSP, containing bootloaders, Linux kernel and root file system, which can include various libraries and middleware, and sample applications. The objective is to enable more hardware platforms, to build on top of the NXP Automotive Linux® BSP in order to add additional components like drivers, middleware or applications.

More details are provided in Chapter 2.

For the open-source components part of NXP Automotive Linux® BSP the upstream (open-source community mainline) versions are used, together with the necessary modifications required to enable execution on NXP hardware. The changes might include: specific configuration of components, driver modifications, device tree definitions for NXP specific hardware, new driver and feature implementations, additional sample applications, etc.

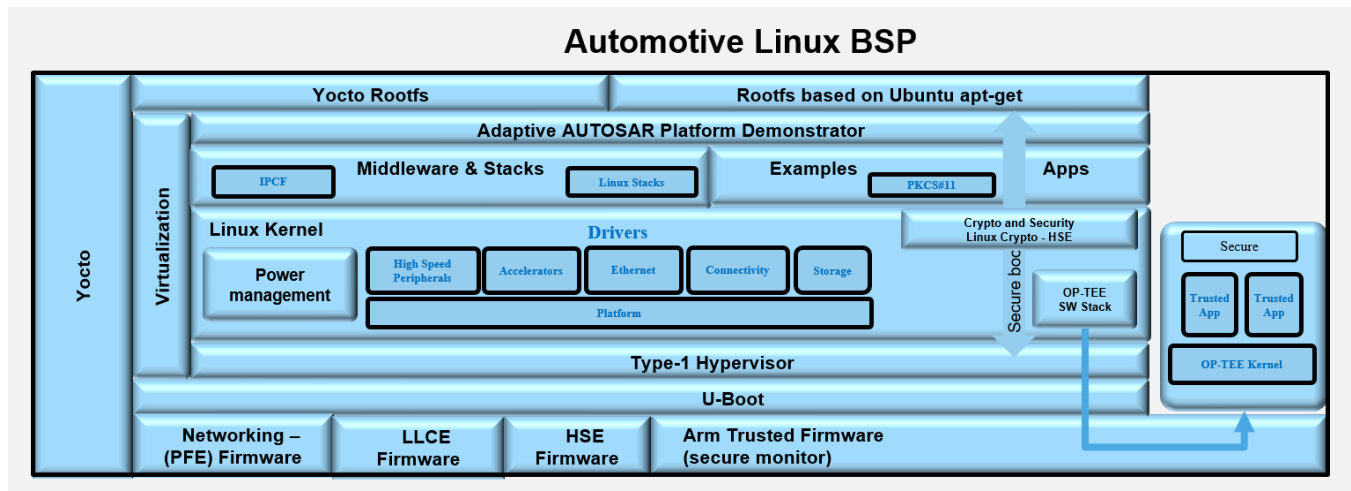


Figure 1. NXP Automotive Linux® BSP Architecture Diagram

## 2. Software Content

This section contains a description of the NXP Automotive Linux® BSP features covered by the following main components:

- Firmware: Non-open-source firmware binaries - prerequisites for S32 accelerators support
  - PFE Firmware
  - HSE Firmware
  - LLCE Firmware
- Bootloader
  - Arm® Trusted Firmware
  - U-Boot
- Linux Kernel
- Root file system
- Technologies
  - Virtualization
  - Power Management
  - Crypto, Security and OP-TEE
- Yocto Project based distribution
  - Yocto root file system and Ubuntu compatible root file system

- Middleware and Stacks
- Adaptive AUTOSAR Platform Demonstrator

## 2.1. Arm Trusted Firmware

The Arm Trusted Firmware (TF-A) is a software component of the Linux BSP with a double role, one role is to serve as a boot loader and the second one is to serve as Secure Monitor. As a boot loader, it runs before U-Boot on the Armv8 boot core and performs core and SoC initializations. As a Secure Monitor, the TF-A provides power management services to other software components via the standard Power State Coordination Interface (PSCI) and system management via System Control and Management Interface (SCMI) API/ABI.

## 2.2. U-Boot

U-Boot is the OS bootloader responsible for the initialization of specific hardware components up to a basic level, in order to allow the loading of the Linux kernel image, the device tree blob and an initial RAM file system (initramfs or initrd) from a boot media (such as an SD card, eMMC, QSPI flash, network, etc.). U-Boot also has a command-line interface which can be used during development.

## 2.3. Linux Kernel

The Linux kernel is responsible for memory and resource management, scheduling, driver loading and unloading, interrupt handling and APIs (syscalls, ioctls, pseudo or virtual file systems) provided for communication between the kernel or kernel drivers and user space applications.

The Linux kernel from the NXP Automotive Linux® BSP (ALB) is mainly comprised of an upstream version of the Linux with Real-Time (RT) patch and ALB specific additions in order to allow the kernel to run on NXP platforms. Linux with RT patch is an open-source extension to Linux kernel that converts Linux into a fully preemptible kernel.

The Linux kernel source also provides a reference implementation of the drivers, showcasing the way the driver for a specific peripheral should work in case an implementation on another operating system is desired.

Kernel drivers offer a hardware abstraction layer by implementing Linux kernel APIs and provide mechanisms for user space applications in order to interact with the hardware (usually indirectly, through libraries). The interaction paths are driver and hardware dependent and it can differ from driver to driver.

The kernel drivers are part of the same source as the Linux kernel, but in binary form (.ko modules) and can be provided separately from the kernel image.

Details on the communication, capabilities and limitations are provided in other documents, such as Release Notes, user manuals, or documentation included in the source code.

**NXP Automotive Linux® BSP Product Brief**  
**Rev 3.0, 02/2023**

## 2.4. Root File System

A Linux root file system (rootfs) is included by default in the NXP Automotive Linux® BSP. It can be either an NXP-curated image or an image generated using the Ubuntu package manager.

The rootfs allows Linux to boot into user space and may optionally provide default initializations of various subsystems (e.g. networking) or hardware components via their respective drivers. A number of standard applications are included in the rootfs, allowing basic interaction and development.

The root file system represents all of the BSP's internal files together with the hierarchy of directories used to organize them. It contains both system files (kernel image, binary drivers, system configuration files, etc.) as well as user files (custom applications, etc.) and tools. In addition, root file system may include libraries, middleware, sample applications, or Linux kernel modules. NXP Automotive Linux® BSP can operate with two flavors of root file systems: Yocto root file system as generated by default with Yocto build system, or root file system generated using Ubuntu package manager (apt-get).

## 2.5. Technologies

### 2.5.1. Virtualization

Virtualization technologies provide an abstraction layer that enables multiple operating systems – called Guest Operation Systems – to run in parallel on a single system. NXP Automotive Linux® BSP demonstrates virtualization technologies using Xen as a Type 1 Hypervisor. A Type 1 Hypervisor is a software component, running at Armv8 Exception Level 2, that creates and manages Virtual Machines, which can run multiple Guest Operating Systems. Xen is an Open-Source Type-1 Hypervisor hosted by The Linux Foundation.

### 2.5.2. Power Management

Power management features supported by NXP Automotive Linux® BSP are aligned with the upstream power management support in Linux kernel and Arm Trusted Firmware. Power state changes are typically requested by Linux and are executed by the Secure Monitor component of the Arm Trusted Firmware.

Suspend to RAM is a complex power management feature supported by NXP Automotive Linux® BSP. When Suspend to RAM procedure is initiated, the software context is saved and the SoC transitions to the Standby state, which is an SoC low power state. Suspend to RAM procedure puts DDR in self-refresh mode, all but the essential hardware blocks are powered off and the SoC is prepared for the Run to Standby transition. After a wakeup source is triggered, the SoC executes the Standby to Run transition, DDR and A53 clocks are enabled, DDR is taken out of the self-refresh mode and system software restores the saved context.

### 2.5.3. Crypto, Security and OP-TEE

The NXP Hardware Security Engine (HSE) is a security subsystem aimed for running relevant security functions for applications with stringent confidentiality and authenticity requirements. The HSE crypto drivers in NXP Automotive Linux® BSP provide support for offloading cryptographic operations to

HSE's dedicated coprocessors through the kernel crypto API. The HSE support for PKCS#11 provides a user-space module that enables communication with HSE from user-space applications.

NXP Automotive Linux® BSP demonstrates Arm TrustZone technology within Cortex-A based processors for trusted boot and trusted Operating System to create Trusted Execution Environment (TEE) using Open Portable Trusted Execution Environment (OP-TEE), an open source TEE implementing Arm TrustZone technology. The HSE crypto drivers in OP-TEE provide cryptographic offloading capabilities for the OP-TEE Core and Trusted Applications.

## 2.6. Yocto

Yocto Project is an open-source collaboration project that helps developers create custom Linux-based systems regardless of the hardware architecture. Yocto is not an embedded Linux distribution, but rather is able to create a custom one being able to build all the components (open-source and NXP-specific ones) and create an image with all the NXP Automotive Linux® BSP components.

The Yocto component is responsible for providing a way of building all BSP components and tools necessary to create the binary images that run on the target NXP hardware.

It gives developers and clients the flexibility to extend the provided images with other tools, binaries and utilities that should be part of the final image on the target hardware.

### 2.6.1. IPCF

The Inter-Platform Communication Framework (IPCF) is a set of NXP drivers and libraries bundled together to provide a common communication channel between separate processing units (individual cores belonging to the same or separate SoCs) over different hardware communication media (such as shared memory).

### 2.6.2. Adaptive AUTOSAR Platform Demonstrator

The AUTOSAR Adaptive Platform (AP) is the standardized platform for microprocessor-based ECUs supporting use cases like highly automated and autonomous driving as well as high-speed on-board and off-board communication.

### 3. Supported Targets

The software described in this document is intended to be used with NXP Semiconductors S32G2 and S32G3 devices.

### 4. Quality, Standards Compliance and Testing Approach

Linux BSP product is developed according to NXP Software Development Processes that are Automotive-SPICE (tailored for open source), IATF16949 and ISO9001 compliant.

### 5. Document Information

Table 1. Revision History

Revision number	Date	Substantive changes
1.0	10/2021	Initial release
3.0	02/2023	Updated for S32G3



**How to Reach Us:**

**Home Page:**  
[nxp.com](http://nxp.com)

**Web Support:**  
[nxp.com/support](http://nxp.com/support)

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. "Typical" parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including "typicals," must be validated for each customer application by customer's technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: [nxp.com/SalesTermsandConditions](http://nxp.com/SalesTermsandConditions).

NXP, the NXP logo, NXP SECURE CONNECTIONS FOR A SMARTER WORLD, COOLFLUX, EMBRACE, GREENCHIP, HITAG, I2C BUS, ICODE, JCOP, LIFE VIBES, MIFARE, MIFARE CLASSIC, MIFARE DESFire, MIFARE PLUS, MIFARE FLEX, MANTIS, MIFARE ULTRALIGHT, MIFARE4MOBILE, MIGLO, NTAG, ROADLINK, SMARTLX, SMARTMX, STARPLUG, TOPFET, TRENCHMOS, UCODE, Freescale, the Freescale logo, AltiVec, C 5, CodeTEST, CodeWarrior, ColdFire, ColdFire+, C Ware, the Energy Efficient Solutions logo, Kinetis, Layerscape, MagniV, mobileGT, PEG, PowerQUICC, Processor Expert, QorIQ, QorIQ Qonverge, Ready Play, SafeAssure, the SafeAssure logo, StarCore, Symphony, VortiQa, Vybrid, Airfast, BeeKit, BeeStack, CoreNet, Flexis, MXC, Platform in a Package, QUICC Engine, SMARTMOS, Tower, TurboLink, and UMEMS are trademarks of NXP B.V. All other product or service names are the property of their respective owners. ARM, AMBA, ARM Powered, Artisan, Cortex, Jazelle, Keil, SecurCore, Thumb, TrustZone, and  $\mu$ Vision are registered trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. ARM7, ARM9, ARM11, big.LITTLE, CoreLink, CoreSight, DesignStart, Mali, mbed, NEON, POP, Sensinode, Socrates, ULINK and Versatile are trademarks of ARM Limited (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. Oracle and Java are registered trademarks of Oracle and/or its affiliates. The Power Architecture and Power.org word marks and the Power and Power.org logos and related marks are trademarks and service marks licensed by Power.org.

NXP B.V. 2023. All rights reserved.

Document Number: 3.0  
Rev