



Arrow Lake S Intel® Platform for Linux* Kernel 6.11 Kernel BKM

Validation Report

WW46, November 2024



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis. You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning Intel products described herein. You agree to grant Intel a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest Intel product specifications and roadmaps.

All product plans and roadmaps are subject to change without notice.

The products described may contain design defects or errors known as errata, which may cause the product to deviate from published specifications. Current characterized errata are available on request.

Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Performance varies depending on system configuration. No computer system can be absolutely secure. Check with your system manufacturer or retailer or learn more at [intel.com](https://www.intel.com).

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and non-infringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

© Intel Corporation. Intel, the Intel logo, and other Intel marks are trademarks of Intel Corporation or its subsidiaries.

*Other names and brands may be claimed as the property of others.

Copyright© 2023-2024 Intel Corporation. All rights reserved.

Contents

1	Overview	5
1.1	References	5
1.2	Security Considerations.....	5
2	Pre-requisites.....	6
2.1	Install Ubuntu* 24.04 on the Host System	6
2.2	Tools	6
3	Building the Kernel.....	7
3.1	Setup	7
3.2	Build the Kernel	8

Revision History

Revision	Description	Revision Date
1.0	• Initial Release	September 2023
2.0	• Updated Kernel Version to 6.5.7	February 2024
3.0	• Updated Kernel Version to 6.8.0	June 2024
4.0	• Updated Kernel Version to 6.9.0	July 2024
5.0	• Updated Kernel Version to 6.10.1	August 2024
6.0	• Updated Kernel Version to 6.11	November 2024

§§

1 Overview

This document provides the steps for building the **Intel® Platform for Linux Kernel 6.11** for the Intel ARL-S platform. The OSV/OEM can follow alternative practices best suited for their environment. The Intel® Platform for Linux Kernel 6.11 is based on upstream 6.11 kernel (commit id: 98f7e32f20d28ec452afb208f9cffc08448a2652) and set of patches needed for the ARL-S Platform. Please refer to the references for more information.

1.1 References

- Intel Platform for Linux Kernel 6.11 Patches for ARL-S WW46.xlsx
- Intel Platform for Linux Kernel 6.11 Kernel Config BKM for ARL-S WW46.pdf

1.2 Security Considerations

This document provides a sample integration BKM for configuring the **Intel® Platform for Linux Kernel 6.11**. These instructions shall be used as reference only, and it's the responsibility of the <integrator> to ensure that all components and their dependencies are using the latest available secure versions of the implementation. These instructions are for reference only, hence it's the responsibility of the <integrator> to take security considerations into account prior building a production capable system."

§§

2 *Pre-requisites*

2.1 **Install Ubuntu* 24.04 on the Host System**

Make sure that the host system has Ubuntu* 24.04 installed. The below steps are to be executed on the host system.

2.2 **Tools**

Install the required tools using the below command.

```
sudo apt-get install -y apt-transport-https git rsync build-essential kernel-package fakeroot
```

```
libncurses5-dev libssl-dev ccache bison flex wget openssl libssl-dev dkms libelf-dev libudev-dev  
libpci-dev libiberty-dev liblz4-tool autoconf kernel-wedge asciidoc binutils-dev ca-certificates
```

§§

3 Building the Kernel

Follow the steps in this section to build the kernel.

3.1 Setup

The instructions create a folder called “**intel**” in the home directory. Please follow the below steps to sync the kernel source and setup the environment.

- `mkdir -p intel`
- `cd intel`
- `mkdir -p scripts build src linux-firmware`
 - `mkdir -p build/kernel-debs`
- Clone the LTS kernel 6.11 source code from link: <https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git> to the “src” folder
- `git clone https://git.kernel.org/pub/scm/linux/kernel/git/stable/linux.git src`
- `cd src`
- `git checkout 98f7e32f20d28ec452afb208f9cffc08448a2652`
- Please cherry-pick the patches mentioned in the document “**Intel_Platform_for_Linux_Kernel_6.11_Patches_for_ARL-S_WW46.xlsx**”. Some patches may need to be backported.
 - Note: ww46
 - If merge conflicts occur, it is required to manually resolve.
- Please refer to the reference kernel config entries from “**Intel_Platform_for_Linux_Kernel_6.11_Kernel_Config_BKM_for_ARL-S_WW46.pdf**” to create the .config file for kernel configuration.
- Open the .config and update the following config entries:
 - `CONFIG_EXTRA_FIRMWARE_DIR=“../linux-firmware”`
 - `cd ../`
- Sync the linux-firmware git
 - `git clone https://git.kernel.org/pub/scm/linux/kernel/git/firmware/linux-firmware.git linux-firmware`
 - Make sure the files synced from the git repo are present under the “linux-firmware” directory.
 - `cd linux-firmware`
 - `git checkout f8462923ed8fc874f770b8c6dfad49d39b381f14`
- `cd ../src`
- `make olddefconfig`

Make sure the below are added to the kernel parameters.

- `i915.enable_guc=3`
- `snd-intel-dspcfg.dsp_driver=3`

- `snd_sof_intel_hda_common.sof_use_tplg_nhlt=1`

3.2 Build the Kernel

```
cd ../scripts
```

Create a file “build_ubuntu_kernel.sh” and copy the below contents to this file

```
#!/usr/bin/env bash

cd ../src

if [ -f .config ]; then

    make -j `getconf _NPROCESSORS_ONLN` bindeb-pkg LOCALVERSION=-for-ubuntu

    mkdir -p ../build/kernel-debs

    mv ../*.deb ../build/kernel-debs

else

    echo "Error: Please make sure the .config file is present under ../src folder."

fi

chmod +x build_ubuntu_kernel.sh

./build_ubuntu_kernel.sh
```

Once the build completes, the kernel deb files will be present under the “build/kernel-debs” folder.

§§