

# How to customize kernel for Android

Stable: 18.10.2019 - 12:35 / Revision: 18.10.2019 - 12:35

This page explains how to change the kernel configuration and device tree. It is intended for Distribution Package users.

## Contents

1 Prerequisites .....	1
2 Updating the kernel configuration .....	1
2.1 Save the current configuration .....	2
2.2 Create a new kernel configuration .....	2
2.3 Test new kernel configuration .....	2
2.4 Applying the newly generated kernel configuration .....	2
3 Changing the kernel command line .....	3
4 Changing the Device Tree .....	3

## 1 Prerequisites

The environment must be installed using the Distribution Package adapted to your selected microprocessor device. See the list of Android [Distribution Package](#).

You must get kernel sources as explained in [How to build kernel for Android](#).

To be able to execute following instructions you need to work from your distribution root directory and initialise your environment:

```
PC $> source build/envsetup.sh
PC $> bspsetup
PC $> lunch aosp-<BoardId>-userdebug
```



The `bspsetup` command needs to be run only one time for the distribution

## 2 Updating the kernel configuration



The kernel configuration file used is merged between the following files (this operation is performed at the beginning of the kernel build process):

- The Android configuration files named `android-base.config` and `android-recommended.config` files available in the `device/stm/<STM32Series>-kernel/source/kconfig/<kernel version>/` directory.

- The android-soc.config file available in the device /stm/<STM32Series>-kernel/source/kconfig/<kernel version>/ directory.

By using menuconfig, you can update the configuration file for your needs. Some guidelines are given below.

## 2.1 Save the current configuration

Before making any changes to the kernel configuration, it is advised to build the kernel with the current configuration:

```
PC $> build_kernel
```

A default config file named defconfig.default is generated in out-bsp/<STM32Series>/KERNEL\_OBJ/ directory.

## 2.2 Create a new kernel configuration

Start menuconfig:

```
PC $> build_kernel menuconfig
```

You can load a .config file and modify any entry using this interface. Please read on the top of the windows the instructions to navigate.

Do not forget to hit save.

This will generate a .config file located in out-bsp/<STM32Series>/KERNEL\_OBJ/, and will also generate a defconfig file in the same place.

## 2.3 Test new kernel configuration

To test your new configuration you need to rebuild the kernel, then update the prebuilts and flash, like explain in more details in [How to build kernel for Android](#).

## 2.4 Applying the newly generated kernel configuration

To make your change permanent, you need first to compare the new defconfig file with the defconfig.default file generated at first build.

You can use a graphical tool like meld.

```
PC $> meld out-bsp/<STM32Series>/KERNEL_OBJ/defconfig out-bsp/<STM32Series>/KERNEL_OBJ/def
```

Then report the changes to the android-soc.config file located in device/stm/<STM32Series>-kernel/source/kconfig/<kernel version>/.

And force the regeneration of the .config:

```
PC $> build_kernel defaultconfig
```

*Note: the defconfig.default is then updated.*

To test your configuration you need to rebuild the kernel, then update the prebuilts and flash, like explain in more details in [How to build kernel for Android](#).

### 3 Changing the kernel command line

You can customize the kernel command line call at boot time. Arguments are defined in the BoardConfig.mk file located in `device/stm/<STM32Series>/<BoardId>/`.

Change the value of the **BOARD\_KERNEL\_CMDLINE** variable to your needs.

To apply changes, you need to rebuild the bootimage:

```
PC $> make bootimage-nodeps
```

And then flash the boot partition.



To control the command line after compilation you can execute:

```
PC $> ./system/core/mkbooting/unpack_booting
```

Check the "command line args" value from the output, it must match your changes.

### 4 Changing the Device Tree

The kernel device tree used is set in the dt.mk file available in the `device/stm/<STM32Series>/build/tasks/` directory. For more information please refer to the [Device tree](#) and [How to create your board device tree](#) pages.

This device trees are located inside the Linux kernel source code previously loaded and can be modified directly in `device/stm/<STM32Series>-kernel/linux-<STM32Series>/arch/arm/boot/dts`.

After modification you need to rebuild the device tree by:

```
PC $> build_kernel -i dtb
```

```
PC $> make -j
```

And then re-flash the DT partition.