



## GPU device tree configuration



# GPU device tree configuration

Stable: 07.10.2019 - 09:36 / Revision: 04.10.2019 - 16:00

## Contents

1 Article purpose .....	2
2 DT bindings documentation .....	2
3 DT configuration .....	2
<b>3.1 DT configuration (STM32 level) .....</b>	<b>2</b>
4 How to configure the DT using STM32CubeMX .....	3
5 References .....	3

## 1 Article purpose

This article explains how to configure the **GPU** internal peripheral when it is assigned to the Linux<sup>®</sup>OS.

The configuration is performed using the **device tree** mechanism that provides a hardware description of the GPU internal peripheral, used by the STM32 GPU Linux driver.

## 2 DT bindings documentation

The GPU is represented by the STM32 GPU device tree bindings <sup>[1]</sup>.

## 3 DT configuration

This hardware description is a combination of the **STM32 microprocessor** device tree files (*.dtsi* extension) and **board** device tree files (*.dts* extension). See the [Device tree](#) for an explanation of the device tree file split.

**STM32CubeMX** can be used to generate the board device tree. Refer to [How to configure the DT using STM32CubeMX](#) for more details.

### 3.1 DT configuration (STM32 level)

The GPU device tree node is declared in *stm32mp157c.dtsi* <sup>[2]</sup>. The declaration (shown below) defines the hardware registers base address, the **interrupt**, the **clocks** and the **reset**.

```
gpu: gpu@59000000 {
    compatible = "vivante,gc";
    reg = <0x59000000 0x800>;
    interrupts = <GIC_SPI 109 IRQ_TYPE_LEVEL_HIGH>;
```



```
};  
    clocks = <&rcc GPU>, <&rcc GPU_K>;  
    clock-names = "bus", "core";  
    resets = <&rcc GPU_R>;  
    status = "disabled";
```



**This device tree part is related to STM32 microprocessors. It must be kept as is, without being modified by the end-user.**

## 4 How to configure the DT using STM32CubeMX

The STM32CubeMX tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.

The STM32CubeMX may not support all the properties described in the above [DT bindings documentation](#) paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to [STM32CubeMX user manual](#) for further information.

## 5 References

- [Linux kernel bindings for "vivante,gc"](#)
- [Linux kernel STM32MP157C device tree \(stm32mp157c.dtsi\)](#)

Operating System

Graphics Processing Units

Device Tree

Generic Interrupt Controller

Serial Peripheral Interface