

GPIO device tree configuration

Stable: 11.02.2019 - 12:21 / Revision: 21.01.2019 - 14:51

A [quality version](#) of this page, [approved](#) on *11 February 2019*, was based off this revision.
It was rated: **Expert:** Approved **Technical writer:** Approved **Maintainer:** Approved

Contents

1 Purpose	1
2 DT bindings documentation	1
3 DT configuration	2
3.1 DT configuration (STM32 level)	2
3.2 DT configuration (board level)	2
3.3 DT configuration examples	2
3.3.1 How to add a GPIO dedicated to a device connected on the board	2
4 How to configure GPIOs using STM32CubeMX	3
5 References	3

1 Purpose

The purpose of this article is to explain how to configure the [GPIO internal peripheral](#) through the **GPIOLib framework when this peripheral is assigned to Linux® OS**. The configuration is performed using the device tree^[1].

To better understand I/O management, it is recommended to read the I/O pins overview article^[2].

This article also provides an example explaining how to add a new GPIO in the device tree.

2 DT bindings documentation

The STM32 GPIO bindings are composed of:

- Bindings for GPIO controller^[3]
- Bindings for devices using a GPIO^[4]

It is recommended to read these reference documents if you have never played with GPIO DT.

3 DT configuration

3.1 DT configuration (STM32 level)

The GPIO controller node (gpio controller) is located in the pin controller node that is declared inside the pinctrl dtsi file, *stm32mp157-pinctrl.dtsi*^[5]. See [Device tree](#) for more explanations about device tree files split.

There is one gpio controller node per GPIO bank.

Refer to [Pin controller configuration](#) and to GPIO controller node bindings^[3] for explanations on the gpio controller node.



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

3.2 DT configuration (board level)

Generic guidelines for adding a GPIO to a client device can be found in the document "GPIO bindings for board"^[4].

- Below an example of basic GPIO usage extracted from "GPIO bindings for board"^[4]:

GPIO mappings are defined in the **consumer device node**, through a property named *<function>-gpios*, where *<function>* is requested by the Linux driver through `gpiod_get()`.

```
foo_device {
    ...
    led-gpios = <&gpioa 15 GPIO_ACTIVE_HIGH>, /* red */
               <&gpioa 16 GPIO_ACTIVE_HIGH>, /* green */
               <&gpioa 17 GPIO_ACTIVE_HIGH>; /* blue */
    power-gpios = <&gpiob 1 GPIO_ACTIVE_LOW>;
};
```

Where:

- *&gpiox*: handle on gpio-controller node.
- *15*: line offset in gpio-controller.
- *GPIO_ACTIVE_HIGH*: flag to be used for the GPIO. The list of all available GPIO flags can be found in Linux kernel source code^[6].

3.3 DT configuration examples

3.3.1 How to add a GPIO dedicated to a device connected on the board

The example below shows how to add a GPIOB5 (PB5 on schematics) to a `foo_device`.

```
foo_device {  
    ...  
    x-gpios = <&gpio0 5 GPIO_ACTIVE_HIGH>; /* x function */  
    ...  
};
```

Note: GPIO_ACTIVE_HIGH means that the GPIO line is driven by the logical level 1 , while GPIO_ACTIVE_LOW means that it is driven by the logical level 0.

4 How to configure GPIOs 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

Please refer to the following links for additional information:

1. ↑ [Device tree](#), Device tree mechanism
2. ↑ [Overview of GPIO pins](#), Overview of GPIO pins article
3. ↑ [3.0 3.1 Kernel device tree bindings documentation - gpio.txt](#), GPIO device tree bindings
4. ↑ [4.0 4.1 4.2 GPIO bindings for board](#)
5. ↑ [stm32mp157-pinctrl.dtsi](#) , STM32MP157C Pinctrl device tree file
6. ↑ [include/dt-bindings/gpio/gpio.h](#)