



Reset device tree configuration



Reset device tree configuration

Stable: 06.02.2020 - 14:52 / Revision: 06.02.2020 - 14:50

Template:ArticleMainWriter Template:ArticleApprovedVersion

Contents

1 Article purpose	2
2 DT bindings documentation	2
3 DT configuration	2
3.1 DT configuration (STM32 level)	2
3.1.1 STM32MP1 Reset node	3
3.2 DT configuration (board level)	3
4 How to configure the DT using STM32CubeMX	3
5 References	4

1 Article purpose

This article explains how to configure the **RCC** internal peripheral when it is assigned to the Linux[®] OS. In that case, it is controlled by the **Reset framework**.

The configuration is performed using the **device tree** mechanism that provides a hardware description of the RCC peripheral used by the reset-stm32mp1 Linux driver and by the Reset framework.

2 DT bindings documentation

The Reset device tree bindings are composed of:

- generic DT bindings^[1] used by the Reset framework.
- vendor Reset DT bindings^[2] used by the reset-stm32mp1 driver: this binding document explains how to write device tree files for reset.

3 DT configuration

3.1 DT configuration (STM32 level)

The STM32MP1 Reset node is same node of Clock (they share same hardware IP) and is located in the *stm32mp157c.dtsi* ^[3]. See the **Device tree** for further explanation.

3.1.1 STM32MP1 Reset node

We need to specify the number of cells in a reset specifier.

For the STM32MP1, reset driver request only 1 cell. This is configured by setting the property 'reset-cells' of the rcc device tree node to 1.

```
rcc: rcc@50000000 {
    compatible = "st,stm32mp1-rcc", "syscon";
    #clock-cells = <1>;
    #reset-cells = <1>;
    reg = <0x50000000 0x1000>;
    ...
};
```



This device tree part is related to STM32MP1 microprocessors. It must be kept as-is, without being modified by the end-user.

3.2 DT configuration (board level)

If a Linux driver needs a reset signal, it should be declared in its DT node as shown below:

resets = <phandle> : List of phandle and reset specifier pairs, one pair for each reset signal that affects the device, or that the device manages.

- Example:

```
i2c2: i2c@40013000 {
    compatible = "st,stm32f7-i2c";
    reg = <0x40013000 0x400>;
    interrupt-names = "event", "error", "wakeup";
    interrupts-extended = <&intc GIC_SPI 33 IRQ_TYPE_LEVEL_HIGH>,
        <&intc GIC_SPI 34 IRQ_TYPE_LEVEL_HIGH>,
        <&exti 22 1>;
    clocks = <&rcc I2C2_K>;
    resets = <&rcc I2C2_R>;
    ...
};
```

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, which 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:

- [Documentation/devicetree/bindings/reset/reset.txt](#) , Reset device tree bindings
- [Documentation/devicetree/bindings/reset/st%2Cstm32mp1-rc.txt](#) , STM32MP1 Reset device tree bindings
- [stm32mp157c.dtsi](#) STM32MP157C device tree file