RTC device tree configuration

Contents

Stable: 20.11.2020 - 17:14 / Revision: 20.11.2020 - 17:14

Stable: 20.11.2020 - 17:17 / Revision: 20.11.2020 - 17:17

Contents

1 Article purpose

This article explains how to configure the **RTC** internal peripheral when it is assigned to the Linux®OS. In this case, it is controlled by the **RTC framework**.

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

2 DT bindings documentation

The **RTC** is represented by the *STM32 RTC device tree bindings*[1]

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 **RTC** node is declared in the file *stm32mp151.dtsi* [2]. It describes the hardware register address, clocks and interrupts.

```
rtc: rtc@5c004000 {
        compatible = "st,stm32mp1-rtc";
        reg = <0x5c004000 0x400>;                                   --> Register location and
length
        clocks = <&rcc RTCAPB>, <&rcc RTC>;
        clock-names = "pclk", "rtc_ck";
        interrupts-extended = <&intc GIC_SPI 3 IRQ_TYPE_LEVEL_HIGH>,
                              <&exti 19 1>;
        status = "disabled";
};
```

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)

This part is used to enable the **RTC** used on a board, which is done by setting the **status** property to **okay**.

An "st,lsco" property is available to select and enable the RTC output on which RTC low-speed clock is output. The valid ouput values are defined in [3]. A pinctrl state named "default" can be defined to reserve a pin for the RTC output.

3.3 DT configuration examples

```
#include <dt-bindings/rtc/rtc-stm32.h>
...
&rtc {
        st,lsco = <RTC_OUT2_RMP>;
        pinctrl-0 = <&rtc_out2_rmp_pins_a>;
        pinctrl-names = "default";
};
```

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.
STM32CubeMX might 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:

- Device tree bindings
- STM32MP151 device tree
- STM32 RTC bindings constants

Linux® is a registered trademark of Linus Torvalds.

Operating System

Real Time Clock

Device Tree

Generic Interrupt Controller

Serial Peripheral Interface

Stable: 19.03.2021 - 08:52 / Revision: 19.03.2021 - 08:49

Invalid target: no reviewed revision corresponds to the given ID.

Return to Device tree.

Stable: 19.02.2020 - 10:01 / Revision: 04.02.2020 - 15:23

Invalid target: no reviewed revision corresponds to the given ID.

Return to RTC internal peripheral.

Stable: 23.09.2020 - 13:22 / Revision: 12.06.2020 - 13:25

Invalid target: no reviewed revision corresponds to the given ID.

Return to STM32CubeMX.