CRC device tree configuration

Contents

Stable: 13.05.2020 - 08:24 / Revision: 13.05.2020 - 08:20

Stable: 18.06.2020 - v3.2 r1 Revision: 18.06.2020 - v3.2

Contents

1 Article purpose

The purpose of this article is to explain how to configure the *CRC*[1] **when the peripheral is assigned to Linux®OS**.

The configuration is performed using the **device tree mechanism**[2].

The Device tree provides a hardware description of the *CRC*[1], used by STM32 *CRC Linux driver*.

If the peripheral is assigned to another execution context, refer to How to assign an internal peripheral to a runtime context article for guidelines on peripheral assignment and configuration.

2 DT bindings documentation

The *CRC* [1] is represented by the *STM32 CRC device tree bindings*[3]

3 DT configuration

This hardware description is a combination of STM32 microprocessor and board device tree files. See Device tree for explanations on device tree file split.

The **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 CRC node is declared in stm32mp151.dtsi[4]. It provides the hardware registers base address and the clock.

```
crc1: crc@58009000 {
        compatible = "st,stm32f7-crc";
        reg = <0x58009000 0x400>;
        clocks = <&rcc CRC1>;
        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 CRC used on a board. This is done by setting the **status** property to **okay**.

3.3 DT configuration examples

```
&crc1 {
        status = "okay";
};
```

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

Please refer to the following links for additional information:

- 1.01.11.2 CRC internal peripheral
- Device tree
- Documentation/devicetree/bindings/crypto/st,stm32-crc.txt
- STM32MP151 device tree file

Cyclic redundancy check calculation unit

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Stable: 12.02.2020 - 16:39 / Revision: 12.02.2020 - 16:37

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

Return to CRC internal peripheral.

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: 08.03.2021 - 16:13 / Revision: 16.02.2021 - 17:11

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

Return to How to assign an internal peripheral to a runtime context.

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.