DMAMUX device tree configuration

Contents

Stable: 19.03.2021 - 17:21 / Revision: 19.03.2021 - 12:35

Stable: 18.06.2021 - 17.11 / Revision: 18.06.2021 - 12.00

Contents

1 Article purpose

This article explains how to configure the DMAMUX internal peripheral when it is assigned to the Linux®OS. In that case, it is controlled by the Dmaengine overview.

The configuration is performed using the Device tree mechanism that provides a hardware description of the DMAMUX internal peripheral, used by the STM32 DMAMUX Linux driver and by the DMA framework.

The hardware description is a combination of:

- STM32 DMAMUX peripheral
- and STM32 DMAMUX client

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

Complete device tree bindings can be found at this location: [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)

At device level, DMAMUX is declared as follows:

```
dmamux1: dma-router@48002000 {
    compatible = "st,stm32h7-dmamux";
    reg = <0x48002000 0x40>;
    #dma-cells = <3>;
    dma-requests = <128>;
    dma-masters = <&dma1 &dma2>;
    dma-channels = <16>;
    clocks = <&rcc DMAMUX>;
    resets = <&rcc DMAMUX_R>;
};
```

The DTS file is located under arch/arm/boot/dts/stm32mp151.dtsi

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

3.2 DT configuration (board level)

No board device tree configuration is required.
The whole configuration remains at STM32 level.

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:

- Documentation/devicetree/bindings/dma/st,stm32-dmamux.yaml

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Device Tree Source (in software context) or Digital Temperature Sensor (in peripheral context)

Stable: 04.02.2020 - 15:42 / Revision: 04.02.2020 - 15:34

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

Return to DMAMUX 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: 22.03.2021 - 08:50 / Revision: 22.03.2021 - 08:43

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

Return to Dmaengine overview.

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.