CEC device tree configuration

Contents

Stable: 25.02.2021 - 17:39 / Revision: 25.02.2021 - 17:51

A quality version of this page, approved on *25 February 2021*, was based off this revision.

Contents

1 Article purpose

This article explains how to configure the *CEC* [1] **when the peripheral is assigned to the Linux®OS**.

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

The Device tree provides a hardware description of the CEC [1] used by the STM32 *CEC 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 CEC is represented by the STM32 CEC device tree bindings [3].

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 CEC device tree node is declared in stm32mp151.dtsi [4]. It describes the hardware register address, clocks, interrupts and power domains.

```
cec: cec@40016000 {
        compatible = "st,stm32-cec";
        reg = <0x40016000 0x400>;
        interrut-names = "event", "wakeup";
        interrupts-extended = <&intc GIC_SPI 94 IRQ_TYPE_LEVEL_HIGH>,
                              <&exti 69 1>;
        clocks = <&rcc CEC>, <&rcc CK_LSE>;
        clock-names = "cec", "hdmi-cec";
        power-domains = <&pd_core>;
        status = "disabled";
};
```

Warning

This device tree part is related to STM32 microprocessors. It must be kept as is, without being modified by the end-user. It is strongly advised not to change this configuration except for the "hdmi-cec" clock source.

3.2 DT configuration (board level)

The CEC device tree node contains the pinctrl description and the "okay" status:

```
&cec {
        pinctrl-names = "default", "sleep";
        pinctrl-0 = <&cec_pins_a>;
        pinctrl-1 = <&cec_pins_sleep_a>;
        status = "okay";
};
```

You can find a full example of the STM32MP15 Evaluation board device tree in stm32mp15xx-evx.dtsi [5].

You can find a full example of the CEC pins [6] in the stm32mp15-pinctrl.dtsi [7]

```
                        cec_pins_a: cec-0 {
                                pins {
                                        pinmux = <STM32_PINMUX('A', 15, AF4)>; /* HDMI_CEC
 */
```

```
                                        bias-disable;
                                        drive-open-drain;
                                        slew-rate = <0>;
                                };
                        };

                        cec_pins_sleep_a: cec-sleep-0 {
                                pins {
                                        pinmux = <STM32_PINMUX('A', 15, ANALOG)>; /* HDMI_
CEC */
                                };
                        };
```

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:

- 1.01.1 CEC internal peripheral
- Device tree
- st,stm32-cec.txt Linux kernel bindings
- Linux kernel STM32MP15x device tree (stm32mp151.dtsi)
- Linux kernel STM32MP157 Evaluation board device tree (stm32mp15xx-evx.dtsi)
- Pinctrl device tree configuration
- Linux kernel STM32MP15x pinctrl device tree (stm32mp15-pinctrl.dtsi)

Consumer Electronics Control (HDMI standard)

Linux® is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Generic Interrupt Controller

Serial Peripheral Interface

Low Speed External oscillator (STM32 clock source)

High-Definition Multimedia Interface (HDMI standard)