SPI device tree configuration

Stable: 15.04.2019 - 14:55 / Revision: 05.04.2019 - 15:36

Contents

1 Article purpose

This article explains how to configure the *SPI internal peripheral*[1] **when the peripheral is assigned to Linux® OS**, and in particular:

- how to configure the STM32 SPI peripheral
- how to configure the STM32 external SPI devices present either on the board or on a hardware extension.

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

It is used by the *STM32 SPI Linux® driver* that registers relevant information in the SPI framework.

2 DT bindings documentation

The SPI bus and its associated device are represented by:

- The *Generic device tree bindings for SPI buses*[3]
- The *STM32 SPI controller device tree bindings*[4]

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, each SPI controller is declared as follows:

```
spi1: spi@44004000 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "st,stm32h7-spi";
        reg = <0x44004000 0x400>;
        interrupts = <GIC_SPI 35 IRQ_TYPE_LEVEL_HIGH>;
        clocks = <&rcc SPI1_K>;
        resets = <&rcc SPI1_R>;
        dmas = <&dmamux1 37 0x400 0x05>,
               <&dmamux1 38 0x400 0x05>;
        dma-names = "rx", "tx";
        power-domains = <&pd_core>;
        status = "disabled";
};
```

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

Refer to the DTS file: stm32mp157c.dtsi[5]

3.2 DT configuration (board level)

```
&spi1 {
        pinctrl-names = "default", "sleep";
        pinctrl-0 = <&spi1_pins_a>;
        pinctrl-1 = <&spi1_sleep_pins_a>;
        cs-gpios = <&gpioz 3 0>;
        status = "okay";

        foo@0 {
                compatible = "spi-foo";
                #address-cells = <1>;
                #size-cells = <0>;
                reg = <0>; /* CS #0 */
                spi-max-frequency = <10000000>;
        };
};
```

There are two levels of configuration:

- Configuration of the SPI bus properties:
  - **pinctrl-0&1** configuration depends on hardware board configuration and on how the SPI devices are connected to MOSI, MISO and Clk pins.
    More details about pin configuration are available here: Pinctrl device tree configuration
  - **cs-gpios** represents the list of GPIOs used as chip selects. Native controller chip select defined by a NULL value is not supported by STM32MP1 SPI driver.
    More details about GPIO configuration are available here: GPIO device tree configuration

- **dmas**: by default, DMAs are specified for all SPI instances. This is up to the user to **remove** them if they are not needed. ***/delete-property/*** is used to remove DMA usage for SPI. Both **/delete-property/dma-names** and **/delete-property/dma** have to be inserted to get rid of DMAs.

- Configuration of the properties of the SPI device connected on the bus:
  - **compatible** represents the name of the SPI device driver.
  - **reg** represents the index of the gpio chip select associated to this SPI device.
  - **spi-max-frequency** represents the maximum SPI clocking speed for the device (in Hz).

For more information about SPI bus and SPI device bindings, please refer to spi-bus.txt[3]

3.3 DT configuration example

Example: of an external TPM device:

```
&spi1 {
        pinctrl-names = "default", "sleep";
        pinctrl-0 = <&spi1_pins_a>;
        pinctrl-1 = <&spi1_sleep_pins_a>;
        cs-gpios = <&gpioz 3 0>;
        status = "okay";

        st33zp24@0 {
                compatible = "st,st33htpm-spi";
                #address-cells = <1>;
                #size-cells = <0>;
                reg = <0>; /* CS #0 */
                spi-max-frequency = <10000000>;
        };
};
```

The above example registers a TPM device on spi1 bus, selected by chip select 0 also known as GPIO-Z3.
This instance is compatible with the driver registered with the same compatible property (st,st33htpm-spi).

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. ↑ SPI internal peripheral
2. ↑ Device tree
3. ↑ [3.03.1] Documentation/devicetree/bindings/spi/spi-bus.txt , Generic device tree bindings for SPI buses
4. ↑ Documentation/devicetree/bindings/spi/spi-stm32.txt
5. ↑ arch/arm/boot/dts/stm32mp157c.dtsi

Serial Peripheral Interface

Operating System

Device Tree

Generic Interrupt Controller

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

General-Purpose Input/Output

Direct Memory Access

Trusted Platform Module