Clock overview

Contents

Stable: 15.10.2019 - 11:58 / Revision: 15.10.2019 - 11:57

Stable: 15.10.2019 - 11:37 / Revision: 15.10.2019 - 11:37

A quality version of this page, approved on *15 October 2019*, was based off this revision.

This article gives information about the Linux® Clock framework.
Clocks are generally provided by internal/external oscillators or PLLs.
They can pass through a gate, a muxing, a divider or a multiplier.
All peripheral clocks are organized as a tree.

All these elements are managed by the Common Clock framework.

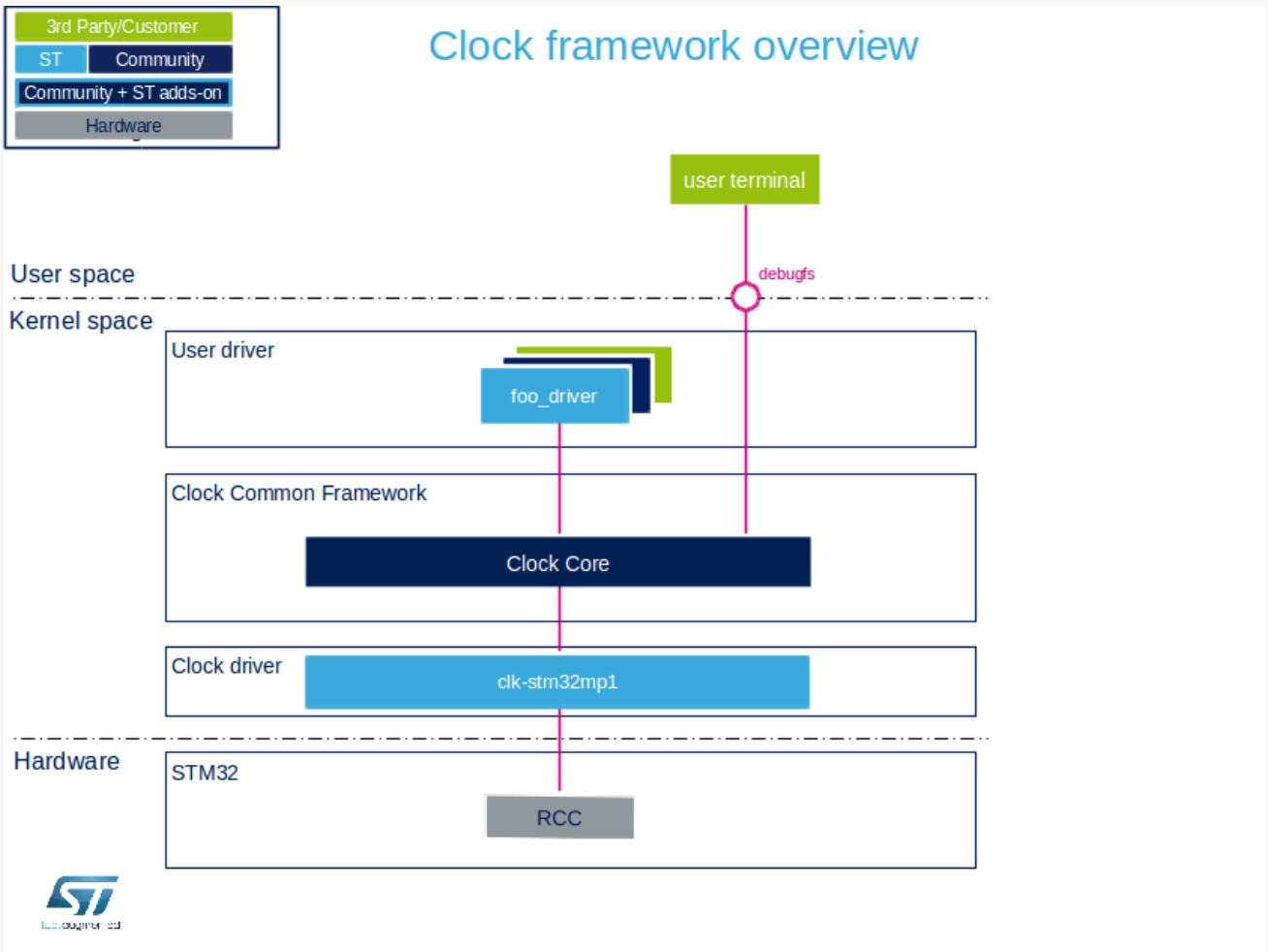
Contents

1 Framework purpose

The purpose of this article is to introduce the Common Clock Framework. It provides general information and, based on examples, explains how to use it.

Linux Common Clock framework offers a generic API for configuring and controlling the different system clocks.
Drivers can easily enable/disable clocks, change their frequency, change the parent of the clocks associated to peripherals without any knowledge of the clock characteristics.
All clock tree specificities (such as clock source selection, muxing, dividor and gate) are abstracted by the Common Clock framework and the associated system clock driver.

2 System overview



Clock Overview

2.1 Component description

- **foo driver:** any peripheral driver which needs to control and activate clock(s) associated to a given peripheral.
- **Clock core:** the Common Clock framework is the generic Linux kernel interface that controls the clock nodes available in the system.

  It features two generic interfaces:

  - The upper interface unifies the definition and control of the clocks for all Linux platforms. This agnostic API is used by peripheral drivers for configuring and controlling the clocks associated to a specific peripheral.

  - The lower interface allows the registration of platform specific functions in order to manage a platform specific clock tree.
- **clk-stm32mp1:** stm32mp1 specific clock driver that supports RCC clocks.
- **RCC:** reset and clock controler RCC.

2.2 API description

Documentation on the Common Clock framework can be found in the kernel documentation folder: https://www.kernel.org/doc/Documentation/driver-api/clk.rst

- Main kernel clock API
  - **devm_get_clk():** looks up and obtains from the device tree a managed reference to a clock producer, to a root clock or to a clock node.
  - **clk_prepare_enable():** selects a parent clock, configures the corresponding multiplexor and dividor, and enables the clock gating.
  - **clk_disable_unprepare():** unprepares and gate a clock.
  - **clk_get_rate():** obtains the current frequency (in Hz) for a given clock.
  - **clk_set_rate():** sets the frequency for a given clock. If a clock has several parents, the clock framework can change the parent in order to obtain a better frequency.
  - **clk_get_parent():** gets the parent clock source for a given clock
  - **clk_set_parent():** sets the parent clock source for a given clock
  - **...**

3 Configuration

3.1 Kernel configuration

By default the Common Clock framework is activated in the kernel configuration.

3.2 Device tree configuration

A detailed device tree configuration is described in Clock device tree configuration.

4 How to use the Common Clock framework

This paragraph describes how a standard peripheral driver can retrieve its clock configuration from the device tree and configure it.

The clocks associated to a given peripheral are declared in the device tree as described in https://www.kernel.org/doc/Documentation/devicetree/bindings/clock/clock-bindings.txt.

Specific platform define statements that abstract hardware clock offsets are defined in dt-bindings/clock/stm32mp1-clks.h header file.

Below an example of clock association to a foo driver:

```
foo: foo@adcdefgh {
       compatible = "foo-driver";
       ...
       clocks = <&rcc FOO_K>;
       ...
};
```

Before using the clock specified in the device tree node, the foo driver has to request it at probe execution.

```
static int foo_probe(struct platform_device *pdev)
{
        struct clk *clk;
        ...
        clk = devm_clk_get(&pdev->dev, NULL);
        if (IS_ERR(clk)) {
                ret = PTR_ERR(clk);
                dev_err(&pdev->dev, "clk get failed: %d\n", ret);
                goto err_master_put;
        }
        ...
}
```

The clock can then be configured and enabled using the Common Clock framework API.

If the peripheral needs several clocks, a name must be associated to each clock handle in the device tree node. The specified names must be used by the driver to retrieve the corresponding pointer for each clock.

Below an example of foo driver managing 2 clocks:

```
foo: foo@abcdefgh {
        compatible = "foo-driver";

        clocks = <&rcc FOO1_K>, <&rcc FOO2_K>;
        clock-names = "foo1", "foo2";
};
```

```
static int foo_probe(struct platform_device *pdev)
{
        priv->foo1clk = devm_clk_get(&pdev->dev, "foo1");
        if (IS_ERR(priv->foo1clk)) {
                ret = PTR_ERR(priv->foo1clk);
```

```
                        if (ret != -ENOENT) {
                                dev_err(&pdev->dev, "Can't get 'foo1' clock\n");
                                return ret;
                        }
                }
                  priv->foo2clk = devm_clk_get(&pdev->dev, "foo2")';
                if (IS_ERR(priv->foo2clk)) {
                        ret = PTR_ERR(priv->foo2clk);
                        if (ret != -ENOENT) {
                                dev_err(&pdev->dev, "Can't get 'foo2' clock\n");
                                return ret;
                        }
                }
          }
```

```
        ret = clk_prepare_enable(priv->foo1clk);
        if (ret < 0) {
                dev_err(dev, "foo1 clk enable failed\n");
                goto err_bclk_disable;
        }
```

5 How to trace and debug the framework

5.1 Tracing using dynamic debug

By default, no kernel log showing the clock activity. However the user can enable the dynamic debug for the clock framework driver:

```
Board $> dmesg -n8
Board $> echo "file drivers/clk/* +p" > /sys/kernel/debug/dynamic_debug/control
```

Refer to dynamic debug for more details.

5.1.1 How to monitor with debugfs

Information on clocks are available in Debugfs interface located in the '/sys/kernel/debug/clk' directory.

It helps viewing all the clocks registered in tree format.

Show clock tree:

```
Board $> cat /sys/kernel/debug/clk/clk_summary

   clock                         enable_cnt  prepare_cnt        rate   accuracy   phase
----------------------------------------------------------------------------------------
clk-ext-camera                          0            0    24000000          0 0
ck_usbo_48m                             1            1    48000000          0 0
   usbo_k                               1            1    48000000          0 0
ck_dsi_phy                              0            0           0          0 0
   dsi_k                                0            0           0          0 0
i2s_ckin                                0            0           0          0 0
clk-csi                                 0            0     4000000          0 0
   ck_csi                               0            0     4000000          0 0
      rng2_k                            0            0     4000000          0 0
      rng1_k                            0            0     4000000          0 0
clk-lsi                                 1            1       32000          0 0
   ck_lsi                               1            1       32000          0 0
      dac12_k                           0            0       32000          0 0
clk-lse                                 1            1       32768          0 0
   ck_lse                               1            1       32768          0 0
      ck_rtc                            1            1       32768          0 0
      cec_k                             0            0       32768          0 0
clk-hsi                                 1            1    64000000          0 0
   clk-hsi-div                          1            1    64000000          0 0
      ck_hsi                            2            2    64000000          0 0
         ck_mco1                        0            0    64000000          0 0
         uart8_k                        0            0    64000000          0 0
         uart7_k                        0            0    64000000          0 0
         uart6_k                        0            0    64000000          0 0
         uart5_k                        0            0    64000000          0 0
         uart4_k                        1            1    64000000          0 0
         usart3_k                       0            0    64000000          0 0
         usart2_k                       0            0    64000000          0 0
         usart1_k                       0            0    64000000          0 0
         i2c6_k                         0            0    64000000          0 0
         i2c4_k                         1            1    64000000          0 0
         i2c5_k                         0            0    64000000          0 0
         i2c3_k                         0            0    64000000          0 0
         i2c2_k                         0            0    64000000          0 0
```

```
         i2c1_k                    0          0    64000000          0 0
         spi6_k                    0          0    64000000          0 0
         spi5_k                    0          0    64000000          0 0
         spi4_k                    0          0    64000000          0 0
clk-hse                            1          1    24000000          0 0
   ck_hse                          6          6    24000000          0 0
      ck_hse_rtc                   0          0     1000000          0 0
      stgen_k                      1          1    24000000          0 0
      usbphy_k                     1          1    24000000          0 0
      ck_per                       0          0    24000000          0 0
         adc12_k                   0          0    24000000          0 0
      ref4                         1          1    24000000          0 0
         pll4                      1          1   508000000          0 0
            pll4_r                 0          0    56444445          0 0
            pll4_q                 1          1    50800000          0 0
               ltdc_px             1          1    50800000          0 0
               dsi_px              0          0    50800000          0 0
               fdcan_k             0          0    50800000          0 0
            pll4_p                 0          0    56444445          0 0
      ref3                         1          1    24000000          0 0
         pll3                      2          3   786431640          0 0
            pll3_r                 1          1    98303955          0 0
               sdmmc3_k            0          0    98303955          0 0
               sdmmc2_k            0          0    98303955          0 0
               sdmmc1_k            1          1    98303955          0 0
            pll3_q                 0          3    49151978          0 0
               adfsdm_k            0          0    49151978          0 0
               sai4_k              0          1    49151978          0 0
               sai3_k              0          0    49151978          0 0
               sai2_k              0          2    49151978          0 0
               sai1_k              0          0    49151978          0 0
               spi3_k              0          0    49151978          0 0
               spi2_k              0          0    49151978          0 0
               spi1_k              0          0    49151978          0 0
               spdif_k             0          1    49151978          0 0
            pll3_p                 1          1   196607910          0 0
               ck_mcu              6         19   196607910          0 0
                  dfsdm_k          0          1   196607910          0 0
                  gpiok            0          1   196607910          0 0
                  gpioj            0          1   196607910          0 0
                  gpioi            0          1   196607910          0 0
                  gpioh            0          1   196607910          0 0
                  gpiog            0          1   196607910          0 0
                  gpiof            0          1   196607910          0 0
                  gpioe            0          1   196607910          0 0
                  gpiod            0          1   196607910          0 0
                  gpioc            0          1   196607910          0 0
                  gpiob            0          1   196607910          0 0
                  gpioa            0          1   196607910          0 0
                  ipcc             2          2   196607910          0 0
                  hsem             0          0   196607910          0 0
                  crc2             0          0   196607910          0 0
                  rng2             0          0   196607910          0 0
                  hash2            0          0   196607910          0 0
                  cryp2            0          0   196607910          0 0
                  dcmi             0          0   196607910          0 0
                  sdmmc3           0          0   196607910          0 0
                  usbo             0          0   196607910          0 0
                  adc12            0          0   196607910          0 0
                  dmamux           1          1   196607910          0 0
                  dma2             1          1   196607910          0 0
                  dma1             1          1   196607910          0 0
                  pclk3            1          1    98303955          0 0
                     lptim5_k      0          0    98303955          0 0
                     lptim4_k      0          0    98303955          0 0
                     lptim3_k      0          0    98303955          0 0
                     lptim2_k      0          0    98303955          0 0
```

```
      hdp                  0              0     98303955          0 0
      pmbctrl              0              0     98303955          0 0
      tmpsens              0              0     98303955          0 0
      vref                 0              0     98303955          0 0
      syscfg               1              1     98303955          0 0
      sai4                 0              0     98303955          0 0
      lptim5               0              0     98303955          0 0
      lptim4               0              0     98303955          0 0
      lptim3               0              0     98303955          0 0
      lptim2               0              0     98303955          0 0
   pclk2                   0              0     98303955          0 0
      fdcan                0              0     98303955          0 0
      dfsdm                0              0     98303955          0 0
      sai3                 0              0     98303955          0 0
      sai2                 0              0     98303955          0 0
      sai1                 0              0     98303955          0 0
      usart6               0              0     98303955          0 0
      spi5                 0              0     98303955          0 0
      spi4                 0              0     98303955          0 0
      spi1                 0              0     98303955          0 0
      tim17                0              0     98303955          0 0
      tim16                0              0     98303955          0 0
      tim15                0              0     98303955          0 0
      tim8                 0              0     98303955          0 0
      tim1                 0              0     98303955          0 0
      ck2_tim              0              0    196607910          0 0
         tim17_k           0              0    196607910          0 0
         tim16_k           0              0    196607910          0 0
         tim15_k           0              0    196607910          0 0
         tim8_k            0              0    196607910          0 0
         tim1_k            0              0    196607910          0 0
   pclk1                   0              2     98303955          0 0
      lptim1_k             0              0     98303955          0 0
      mdio                 0              0     98303955          0 0
      dac12                0              1     98303955          0 0
      cec                  0              0     98303955          0 0
      spdif                0              0     98303955          0 0
      i2c5                 0              0     98303955          0 0
      i2c3                 0              0     98303955          0 0
      i2c2                 0              0     98303955          0 0
      i2c1                 0              0     98303955          0 0
      uart8                0              0     98303955          0 0
      uart7                0              0     98303955          0 0
      uart5                0              0     98303955          0 0
      uart4                0              0     98303955          0 0
      usart3               0              0     98303955          0 0
      usart2               0              0     98303955          0 0
      spi3                 0              0     98303955          0 0
      spi2                 0              0     98303955          0 0
      lptim1               0              0     98303955          0 0
      tim14                0              0     98303955          0 0
      tim13                0              0     98303955          0 0
      tim12                0              0     98303955          0 0
      tim7                 0              0     98303955          0 0
      tim6                 0              0     98303955          0 0
      tim5                 0              0     98303955          0 0
      tim4                 0              0     98303955          0 0
      tim3                 0              0     98303955          0 0
      tim2                 0              0     98303955          0 0
      ck1_tim              0              1    196607910          0 0
         tim14_k           0              0    196607910          0 0
         tim13_k           0              0    196607910          0 0
         tim12_k           0              0    196607910          0 0
         tim7_k            0              0    196607910          0 0
         tim6_k            0              1    196607910          0 0
         tim5_k            0              0    196607910          0 0
         tim4_k            0              0    196607910          0 0
```

```
                        tim3_k           0            0   196607910          0 0
                        tim2_k           0            0   196607910          0 0
      ref1                               2            2    24000000          0 0
         pll2                            2            2   533000000          0 0
            pll2_r                       1            1   533000000          0 0
            pll2_q                       0            0   533000000          0 0
               gpu_k                     0            0   533000000          0 0
            pll2_p                       1            1   266500000          0 0
               ck_axi                    9           10   266500000          0 0
                  ck_trace               0            0   133250000          0 0
                  ck_sys_dbg             0            0   266500000          0 0
                  qspi_k                 1            1   266500000          0 0
                  fmc_k                  0            0   266500000          0 0
                  ethstp                 0            0   266500000          0 0
                  usbh                   1            1   266500000          0 0
                  crc1                   0            0   266500000          0 0
                  sdmmc2                 0            0   266500000          0 0
                  sdmmc1                 0            0   266500000          0 0
                  qspi                   0            0   266500000          0 0
                  fmc                    0            0   266500000          0 0
                  ethmac                 1            1   266500000          0 0
                  ethrx                  1            1   266500000          0 0
                  ethtx                  1            1   266500000          0 0
                  gpu                    0            0   266500000          0 0
                  mdma                   1            1   266500000          0 0
                  bkpsram                0            0   266500000          0 0
                  rng1                   0            0   266500000          0 0
                  hash1                  0            0   266500000          0 0
                  cryp1                  0            0   266500000          0 0
                  gpioz                  0            1   266500000          0 0
                  tzc2                   0            0   266500000          0 0
                  tzc1                   0            0   266500000          0 0
                  pclk5                  1            1    66625000          0 0
                     stgen               0            0    66625000          0 0
                     bsec                0            0    66625000          0 0
                     iwdg1               0            0    66625000          0 0
                     tzpc                0            0    66625000          0 0
                     rtcapb              2            2    66625000          0 0
                     usart1              0            0    66625000          0 0
                     i2c6                0            0    66625000          0 0
                     i2c4                0            0    66625000          0 0
                     spi6                0            0    66625000          0 0
                  pclk4                  1            1   133250000          0 0
                     stgenro             0            0   133250000          0 0
                     usbphy              0            0   133250000          0 0
                     iwdg2               1            1   133250000          0 0
                     dsi                 0            0   133250000          0 0
                     ltdc                0            0   133250000          0 0
         pll1                            1            1   650000000          0 0
            pll1_p                       1            1   650000000          0 0
               ck_mpu                    1            1   650000000          0 0
                  ck_mco2                0            0   650000000          0 0
      clk-hse-div2                       0            0    12000000          0 0
ethptp_k                                 0            0           0          0 0
ethck_k
```

In addition, each clock has a dedicated directory in which you can find the information such as prepare count, enable count, rate and accuracy:

```
Board $>/sys/kernel/debug/clk# cd usart2_k
Board $>/sys/kernel/debug/clk/usart2_k# ls
clk_accuracy      clk_flags           clk_phase             clk_prepare_count
clk_enable_count  clk_notifier_count  clk_possible_parents  clk_rate
```

Additional information, such as flags, notifier count and possible parents (for clocks with multiple parents) are also available.
Just execute a 'cat' command to display them.

```
Board $>:/sys/kernel/debug/clk/usart2_k# cat clk_possible_parents
pclk1 pll4_q ck_hsi ck_csi ck_hse
```

6 Source code location

stm32mp1 clock driver source [1]
clock dt-binding interface [2]

7 To go further

See "how to build a clock tree" in STM32MP15_clock_tree.

8 References

- drivers/clk/clk-stm32mp1.c
- include/dt-bindings/clock/stm32mp1-clks.h

Linux® is a registered trademark of Linus Torvalds.

Application programming interface

Reset and Clock Control

Debug File System (See https://en.wikipedia.org/wiki/Debugfs for more details)

Stable: 01.12.2020 - 16:39 / Revision: 01.12.2020 - 16:25

A quality version of this page, approved on *1 December 2020*, was based off this revision.

Contents

1 Article purpose

This article explains how to configure the clocks in the STMP32MP1.

This article focuses on the non-secure world configuration with the device tree technology. Specific articles describe the bootlader stage clock configuration or the alternate ways when disabling the RCC TZEN hardening in non-secure RCC configuration article.

2 Clock device providers

There are 3 clock providers in the STM32MP1. Each are represented by node(s) in the device tree description.

- Fixed clocks, as input clocks with a fixed frequency.
- STM32MP1 RCC clocks, most of the system clocks.
- SCMI clocks, clocks that only the secure world can manipulate, and that the secure world exposes to non-secure world using the SCMI clock protocol and a client/server communication.

The node defines *#clock-cells = <1>;*.
In the device tree bindings, all clock providers must define a specific specifier for the number cells used with the clock device phandle, when referring to the clock. When *#clock-cells = 0*, the clock provider phandle does not need an extra argument. When *#clock-cells = 1*, the clock provider phandle is used with an argument. The STM32MP1 RCC clock and SCMI clock drivers both use stm32mp1 clock DT binding IDs defined in the STM32MP1 clock DT bindings[1].

2.1 STM32MP1 RCC clock device

The device tree defines the RCC clock controller device as a node with *compatible = "st,stm32mp1-rcc"* or *"st,stm32mp1-rcc-secure"* node.

- *"st,stm32mp1-rcc-secure"* complies with configuration where RCC TZEN secure hardening is enabled.
- *"st,stm32mp1-rcc"* complies with configuration where RCC TZEN secure hardening is disabled.

For example, below is an extract from the Linux kernel and U-Boot device tree representation.

```
        rcc: rcc@50000000 {
                compatible = "st,stm32mp1-rcc-secure", "st,stm32mp1-rcc", "syscon";
                reg = <0x50000000 0x1000>;
                #clock-cells = <1>;
                #reset-cells = <1>;

                clock-names = "hse", "hsi", "csi", "lse", "lsi";
                clocks = <&scmi0_clk CK_SCMI0_HSE>,
                         <&scmi0_clk CK_SCMI0_HSI>,
                         <&scmi0_clk CK_SCMI0_CSI>,
                         <&scmi0_clk CK_SCMI0_LSE>,
                         <&scmi0_clk CK_SCMI0_LSI>;
        };
```

Note in the file snipped above the STM32MP1 RCC clocks depend on system clocks registered by the SCMI device using the phandles *&scmi0_clk*. Declaring this dependency helps the Linux kernel and U-Boot boot loader to proprely handle the platform clocks.

2.2 SCMI clock device

The device tree defines SCMI clocks using *compatible = "arm,scmi"* nodes with subnodes specifying protocol@14 (*reg = <0x14>*). The node defines *#clock-cells = 1*. The node phandle (label *scmi1_clk* in example below) is used together with a clock ID, using macros *CK_SCMI1_** defined in the DT bindings[2]. One can refer to the article SCMI overview, chapter STM32MP15 SCMI clock and reset for more details.

```
        scmi-1 {
                compatible = "arm,scmi";
                #address-cells = <1>;
                #size-cells = <0>;
              (...)

                scmi1_clk: protocol@14 {
                        reg = <0x14>;
                        #clock-cells = <1>;
                };
        };
```

The Linux driver handles all STM32MP clocks with the Linux common clock framework. As does U-Boot with its generic clock udevice framework.

The configuration is performed using the device tree mechanism that provides a description of the fixed clocks if any, RCC peripheral clocks and SCMI clocks used.

3 DT bindings documentation

The RCC is a multi function device.

Each function is represented by a separate binding document:

- generic DT bindings[3] used by the Common Clock framework.
- vendor clock DT bindings[4] used by the clk-stm32mp1 driver: this binding document explains how to write device tree files for clocks.
- generic SCMI DT bindings[5] for the clock protocol support.

4 DT configuration

4.1 DT configuration (STM32 level)

The STM32MP1 Clock node is located in the *stm32mp151.dtsi*[6]. See Device tree for more explanations.

See the example of STM32MP1 RCC clock DT node above in this article, with *compatible = "st,stm32mp1-rcc-secure"*. The node specifies its dependency on the input clock using SCMI clock phandles *scmi0_clk*.

```
        rcc: rcc@50000000 {
                compatible = "st,stm32mp1-rcc-secure", "st,stm32mp1-rcc", "syscon";
                reg = <0x50000000 0x1000>;
                #clock-cells = <1>;
                #reset-cells = <1>;

                clock-names = "hse", "hsi", "csi", "lse", "lsi";
                clocks = <&scmi0_clk CK_SCMI0_HSE>,
                         <&scmi0_clk CK_SCMI0_HSI>,
                         <&scmi0_clk CK_SCMI0_CSI>,
                         <&scmi0_clk CK_SCMI0_LSE>,
                         <&scmi0_clk CK_SCMI0_LSI>;
        };
```

4.2 DT configuration (board level)

If a Linux driver needs a clock, it has to be added in its DT node:

clocks = <handle> It is the list of the handles to use in the device tree to refer to the target clock instance. The can be several clocks listed, separated with coma character ',', i.e. *clocks = <handle>, <handle>, <handle>;*

In the clock provider node, property *#clock-cells* defines how many 32-bit IDs are to be used with the device phandle to identify the clock.
When the clock provider defines *#clock-cells = 0*, *<handle>* is the single device tree node phandle reference *<&phandle>*.
When the clock provider defines *#clock-cells = 1*, *<handle>* is a pair *<&phandel id>'*

- Example:

```
    usart3: serial@4000f000 {
        compatible = "st,stm32h7-usart";
        reg = <0x4000f000 0x400>;
        interrupts-extended = <&intc GIC_SPI 39 IRQ_TYPE_LEVEL_HIGH>,
                              <&exti 28 1>;
        clocks = <&rcc USART3_K>;
        power-domains = <&pd_core>;
    };

    usart1: serial@5c000000 {
        compatible = "st,stm32h7-usart";
        reg = <0x5c000000 0x400>;
        interrupts-extended = <&intc GIC_SPI 39 IRQ_TYPE_LEVEL_HIGH>,
                              <&exti 28 1>;
        clocks = <&scmi0_clk CK_SCMI0_USART1>;
        power-domains = <&pd_core>;
    };
```

5 How to configure the DT using STM32CubeMX

The STM32CubeMX tool can be used to configure the STM32MP1 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.

6 References

Please refer to the following links for additional information:

- include/dt-bindings/clock/stm32mp1-clks.h STM32MP1 Clock DT bindings header file
- include/dt-bindings/clock/stm32mp1-clks.h STM32MP1 Reset DT bindings header file
- Documentation/devicetree/bindings/clock/clock-bindings.txt , Clock device tree bindings
- Documentation/devicetree/bindings/clock/st,stm32mp1-rcc.txt , STM32MP1 clock device tree bindings
- >Documentation/devicetree/bindings/arm/arm,scmi.txt SCMI DT bindings
- stm32mp151.dtsi STM32MP157C device tree file

Reset and Clock Control

System control and management interface

Device Tree

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see U-Boot_overview)

High Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

Multi Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

Low Speed Internal oscillator (STM32 clock source)

Generic Interrupt Controller

Serial Peripheral Interface

Stable: 04.02.2020 - 07:47 / Revision: 04.02.2020 - 07:34

A quality version of this page, approved on *4 February 2020*, was based off this revision.

Contents

1 Introduction

Debugfs is a simple-to-use RAM-based file system, specially designed for debugging purposes.

It is provided as a simple way for kernel developers to make information available to the user space.

It is part of a pseudo filesystem. Unlike procfs, which is only meant for process information, or sysfs, which has strict one-value-per-file rules, debugfs has no rules at all.

Developers are free to put any information into debugfs.

2 Installing debugfs on your target board

Debugfs is enabled and ready to be used in all STM32MPU Embedded Software distributions, via the Linux® kernel configuration **CONFIG_DEBUG_FS**, set to yes by default.

```
Symbol: DEBUG_FS
Location:
  Kernel Hacking --->
    Compile-time checks and compiler options -->
      [*] Debug Filesystem
```

Please refer to the Menuconfig or how to configure kernel article for instructions on modifying the configuration and recompiling the Linux kernel image in the distribution package context.

3 Getting started

3.1 How to mount debugfs

Debugfs is typically mounted at */sys/kernel/debug* with a command such as:

```
Board $> mount -t debugfs none /sys/kernel/debug
```

3.2 How to use debugfs in Linux kernel drivers

Debugfs is a simple way for kernel drivers to make information available to user space.

It can be manipulated using several calls from the C header file *include/linux/debugfs.h* , such as:

```
debugfs_create_file → for creating a file in the debug filesystem.
debugfs_create_dir → for creating a directory in the debug filesystem.
debugfs_create_symlink → for creating a symbolic link in the debug filesystem.
debugfs_create_u32 → for creating a file containing a single 32-bit unsigned integer value
debugfs_remove → for removing a debugfs entry from the debug filesytem.
...
```

For further details, refer to the Linux kernel documentation for debugfs[1].

For example, you can refer to an OpenSourceForU article: [2].

3.3 How to visualize debugfs information

On the target board, go the debugfs mount directory:

```
Board $> cd /sys/kernel/debug
```

There you will find all the debugfs entries created by Linux kernel frameworks

```
Board $> ls -l
```

```
drwxr-xr-x   2 root root 0 Jan  1  1970 49000000.usb-otg
drwxr-xr-x   4 root root 0 Jan  1  1970 asoc
drwxr-xr-x  40 root root 0 Jan  1  1970 bdi
drwxr-xr-x  10 root root 0 Jan  1  1970 block
drwxr-xr-x   2 root root 0 Jan  1  1970 bluetooth
drwxr-xr-x   3 root root 0 May 28 15:52 cec
drwxr-xr-x 225 root root 0 Jan  1  1970 clk
drwxr-xr-x   2 root root 0 Jan  1  1970 dma_buf
drwxr-xr-x   3 root root 0 Jan  1  1970 dri
drwxr-xr-x   2 root root 0 Jan  1  1970 dynamic_debug
drwxr-xr-x   2 root root 0 Jan  1  1970 extfrag
-rw-r--r--   1 root root 0 Jan  1  1970 fault_around_bytes
drwxr-xr-x   3 root root 0 May 28 15:52 gc
-r--r--r--   1 root root 0 Jan  1  1970 gpio
```

```
drwxr-xr-x   2 root root 0 Jan  1  1970 hid
drwxr-xr-x   2 root root 0 Jan  1  1970 ieee80211
drwxr-xr-x   7 root root 0 Jan  1  1970 iio
-r--r--r--   1 root root 0 Jan  1  1970 irq_domain_mapping
drwxr-xr-x   2 root root 0 Jan  1  1970 memblock
drwxr-xr-x   3 root root 0 Jan  1  1970 mmc0
drwxr-xr-x   3 root root 0 Jan  1  1970 mmc1
drwxr-xr-x  11 root root 0 Jan  1  1970 mtd
drwxr-xr-x   4 root root 0 Jan  1  1970 pinctrl
drwxr-xr-x   4 root root 0 Jan  1  1970 pm_genpd
drwxr-xr-x   2 root root 0 Jan  1  1970 pm_qos
-r--r--r--   1 root root 0 Jan  1  1970 pwm
drwxr-xr-x   2 root root 0 Jan  1  1970 ras
drwxr-xr-x  13 root root 0 Jan  1  1970 regmap
drwxr-xr-x  21 root root 0 Jan  1  1970 regulator
drwxr-xr-x   3 root root 0 Jan  1  1970 remoteproc
-r--r--r--   1 root root 0 Jan  1  1970 sleep_time
drwxr-xr-x   3 root root 0 Jan  1  1970 stmmaceth
-r--r--r--   1 root root 0 Jan  1  1970 suspend_stats
dr-xr-xr-x   3 root root 0 Jan  1  1970 tracing
drwxr-xr-x   2 root root 0 Jan  1  1970 ubi
drwxr-xr-x   2 root root 0 Jan  1  1970 ubifs
drwxr-xr-x   4 root root 0 Jan  1  1970 usb
-r--r--r--   1 root root 0 Jan  1  1970 wakeup_sources
```

You can then browse to find the requested information.

Please note that the different Linux software frameworks available in STM32MP15 Linux kernel overview provide specific trace and debug information in a "*How to trace and debug the framework*" dedicated chapter, incuding debugfs.

Some examples:

- To list the wakeup sources:

```
Board $> cat /sys/kernel/debug/wakeup_sources
name            active_count    event_count     wakeup_count    expire_count
active_since    total_time      max_time        last_change     prevent_suspend_time
5c002000.i2c:stpmu1@33:onkey    0               0               0
0               0               0               0               4075            0
0-0033          0               0               0               0
0               0               0               3949            0
alarmtimer      0               0               0               0
0               0               0               3901            0
5c004000.rtc    1               1               0               0
0               0               0               5246            0
deleted         0               0               0               0
0               0               0               0               0
```

- To obtain a summary of the clock settings:

```
Board $> cat /sys/kernel/debug/clk/clk_summary
  clock                         enable_cnt  prepare_cnt        rate   accuracy   phase
----------------------------------------------------------------------------------------
ck_usbo_48m                              1            1    48000000          0 0
   usbo_k                                1            1    48000000          0 0
ck_dsi_phy                               1            1           0          0 0
   dsi_k                                 1            1           0          0 0
i2s_ckin                                 0            0           0          0 0
clk-csi                                  1            1     4000000          0 0
   ck_csi                                1            1     4000000          0 0
      rng2_k                             0            0     4000000          0 0
```

```
      rng1_k                          0          0     4000000          0 0
clk-lsi                               1          1       32000          0 0
   ck_lsi                             1          1       32000          0 0
      dac12_k                         0          0       32000          0 0
clk-lse                               1          1       32768          0 0
   ck_lse                             1          1       32768          0 0
      ck_rtc                          2          2       32768          0 0
         rtc_lsco                     1          1       32768          0 0
      lptim5_k                        0          0       32768          0 0
      lptim4_k                        0          0       32768          0 0
      cec_k                           0          0       32768          0 0
clk-hsi                               1          1    64000000          0 0
   clk-hsi-div                        1          1    64000000          0 0
      ck_hsi                          2          2    64000000          0 0
         ck_mco1                      0          0    64000000          0 0
         uart8_k                      0          0    64000000          0 0
         uart7_k                      0          0    64000000          0 0
         uart6_k                      0          0    64000000          0 0
         uart5_k                      0          0    64000000          0 0
         uart4_k                      1          1    64000000          0 0
         usart3_k                     0          0    64000000          0 0
         usart2_k                     1          1    64000000          0 0
         usart1_k                     0          0    64000000          0 0
         i2c6_k                       0          0    64000000          0 0
         i2c4_k                       0          0    64000000          0 0
         i2c5_k                       0          0    64000000          0 0
         i2c3_k                       0          0    64000000          0 0
         i2c2_k                       0          0    64000000          0 0
         i2c1_k                       0          0    64000000          0 0
         spi6_k                       0          0    64000000          0 0
         spi5_k                       0          0    64000000          0 0
         spi4_k                       0          0    64000000          0 0
clk-hse                               1          1    24000000          0 0
   ck_hse                             7          7    24000000          0 0
      ck_hse_rtc                      0          0     1000000          0 0
      stgen_k                         1          1    24000000          0 0
      usbphy_k                        1          1    24000000          0 0
      ck_per                          1          1    24000000          0 0
         adc12_k                      1          1    24000000          0 0
      ref4                            1          1    24000000          0 0
         pll4                         1          1   594000000          0 0
            pll4_r                    0          0    74250000          0 0
            pll4_q                    1          1    31263158          0 0
               ltdc_px                1          1    31263158          0 0
               dsi_px                 0          0    31263158          0 0
               fdcan_k                0          0    31263158          0 0
            pll4_p                    0          0    74250000          0 0
      ref3                            1          1    24000000          0 0
         pll3                         2          3   786431640          0 0
            pll3_r                    1          1    98303955          0 0
               sdmmc3_k               0          0    98303955          0 0
               sdmmc2_k               1          1    98303955          0 0
               sdmmc1_k               0          0    98303955          0 0
            pll3_q                    0          1    49151978          0 0
               adfsdm_k               0          0    49151978          0 0
               sai4_k                 0          0    49151978          0 0
               sai3_k                 0          0    49151978          0 0
               sai2_k                 0          2    49151978          0 0
               sai1_k                 0          0    49151978          0 0
               spi3_k                 0          0    49151978          0 0
               spi2_k                 0          0    49151978          0 0
               spi1_k                 0          0    49151978          0 0
               spdif_k                0          0    49151978          0 0
            pll3_p                    1          1   196607910          0 0
               ck_mcu                 6         16   196607910          0 0
                  dfsdm_k             0          0   196607910          0 0
                  gpiok               0          0   196607910          0 0
```

gpioj	0	0	196607910	0 0
gpioi	0	1	196607910	0 0
gpioh	0	1	196607910	0 0
gpiog	0	1	196607910	0 0
gpiof	0	1	196607910	0 0
gpioe	0	1	196607910	0 0
gpiod	0	1	196607910	0 0
gpioc	0	1	196607910	0 0
gpiob	0	1	196607910	0 0
gpioa	0	1	196607910	0 0
ipcc	2	2	196607910	0 0
hsem	0	0	196607910	0 0
crc2	0	0	196607910	0 0
rng2	0	0	196607910	0 0
hash2	0	0	196607910	0 0
cryp2	0	0	196607910	0 0
dcmi	0	0	196607910	0 0
sdmmc3	0	0	196607910	0 0
usbo	0	0	196607910	0 0
adc12	1	1	196607910	0 0
dmamux	1	1	196607910	0 0
dma2	0	0	196607910	0 0
dma1	1	1	196607910	0 0
pclk3	1	1	98303955	0 0
lptim3_k	0	0	98303955	0 0
lptim2_k	0	0	98303955	0 0
hdp	0	0	98303955	0 0
pmbctrl	0	0	98303955	0 0
tmpsens	0	0	98303955	0 0
vref	0	0	98303955	0 0
syscfg	1	1	98303955	0 0
sai4	0	0	98303955	0 0
lptim5	0	0	98303955	0 0
lptim4	0	0	98303955	0 0
lptim3	0	0	98303955	0 0
lptim2	0	0	98303955	0 0
pclk2	0	0	98303955	0 0
fdcan	0	0	98303955	0 0
dfsdm	0	0	98303955	0 0
sai3	0	0	98303955	0 0
sai2	0	0	98303955	0 0
sai1	0	0	98303955	0 0
usart6	0	0	98303955	0 0
spi5	0	0	98303955	0 0
spi4	0	0	98303955	0 0
spi1	0	0	98303955	0 0
tim17	0	0	98303955	0 0
tim16	0	0	98303955	0 0
tim15	0	0	98303955	0 0
tim8	0	0	98303955	0 0
tim1	0	0	98303955	0 0
ck2_tim	0	0	196607910	0 0
tim17_k	0	0	196607910	0 0
tim16_k	0	0	196607910	0 0
tim15_k	0	0	196607910	0 0
tim8_k	0	0	196607910	0 0
tim1_k	0	0	196607910	0 0
pclk1	0	2	98303955	0 0
lptim1_k	0	0	98303955	0 0
mdio	0	0	98303955	0 0
dac12	0	0	98303955	0 0
cec	0	0	98303955	0 0
spdif	0	0	98303955	0 0
i2c5	0	0	98303955	0 0
i2c3	0	0	98303955	0 0
i2c2	0	0	98303955	0 0
i2c1	0	0	98303955	0 0

```
                  uart8                0            0    98303955          0 0
                  uart7                0            0    98303955          0 0
                  uart5                0            0    98303955          0 0
                  uart4                0            0    98303955          0 0
                  usart3               0            0    98303955          0 0
                  usart2               0            0    98303955          0 0
                  spi3                 0            0    98303955          0 0
                  spi2                 0            1    98303955          0 0
                  lptim1               0            0    98303955          0 0
                  tim14                0            0    98303955          0 0
                  tim13                0            0    98303955          0 0
                  tim12                0            0    98303955          0 0
                  tim7                 0            0    98303955          0 0
                  tim6                 0            0    98303955          0 0
                  tim5                 0            0    98303955          0 0
                  tim4                 0            0    98303955          0 0
                  tim3                 0            0    98303955          0 0
                  tim2                 0            0    98303955          0 0
                  ck1_tim              0            1   196607910          0 0
                     tim14_k            0            0   196607910           0 0
                     tim13_k            0            0   196607910           0 0
                     tim12_k            0            0   196607910           0 0
                     tim7_k           0            0   196607910          0 0
                     tim6_k           0            1   196607910          0 0
                     tim5_k           0            0   196607910          0 0
                     tim4_k           0            0   196607910          0 0
                     tim3_k           0            0   196607910          0 0
                     tim2_k           0            0   196607910          0 0
ref1                                  2            2    24000000          0 0
   pll2                               2            2   533000000          0 0
      pll2_r                          1            1   533000000          0 0
      pll2_q                          0            0   533000000          0 0
         gpu_k                        0            0   533000000          0 0
      pll2_p                          1            1   266500000          0 0
         ck_axi                       8            9   266500000          0 0
            ck_trace                  0            0   133250000          0 0
            ck_sys_dbg                0            0   266500000          0 0
            qspi_k                    0            0   266500000          0 0
            fmc_k                     0            0   266500000          0 0
            ethstp                    0            0   266500000          0 0
            usbh                      1            1   266500000          0 0
            crc1                      0            0   266500000          0 0
            sdmmc2                    0            0   266500000          0 0
            sdmmc1                    0            0   266500000          0 0
            qspi                      0            0   266500000          0 0
            fmc                       0            0   266500000          0 0
            ethmac                    1            1   266500000          0 0
            ethrx                     1            1   266500000          0 0
            ethtx                     1            1   266500000          0 0
            gpu                       0            0   266500000          0 0
            mdma                      1            1   266500000          0 0
            bkpsram                   0            0   266500000          0 0
            rng1                      0            0   266500000          0 0
            hash1                     0            0   266500000          0 0
            cryp1                     0            0   266500000          0 0
            gpioz                     0            1   266500000          0 0
            tzc2                      0            0   266500000          0 0
            tzc1                      0            0   266500000          0 0
            pclk5                     1            1    66625000          0 0
               stgen                  0            0    66625000          0 0
               bsec                   0            0    66625000          0 0
               iwdg1                  0            0    66625000          0 0
               tzpc                   0            0    66625000          0 0
               rtcapb                 2            2    66625000          0 0
               usart1                 0            0    66625000          0 0
               i2c6                   0            0    66625000          0 0
               i2c4                   0            0    66625000          0 0
```

```
                    spi6                0            0    66625000          0 0
                 pclk4                  1            1   133250000          0 0
                    stgenro             0            0   133250000          0 0
                    usbphy              0            0   133250000          0 0
                    iwdg2               1            1   133250000          0 0
                    dsi                 0            0   133250000          0 0
                    ltdc                0            0   133250000          0 0
         pll1                           1            1   650000000          0 0
            pll1_p                      1            1   650000000          0 0
               ck_mpu                   1            1   650000000          0 0
                  ck_mco2               0            0   650000000          0 0
      clk-hse-div2                      0            0    12000000          0 0
ethptp_k                                0            0           0          0 0
ethck_k                                 0            0           0          0 0
```

- To obtain a summary of the regulator settings:

```
Board $> cat /sys/kernel/debug/regulator/regulator_summary
regulator                       use open bypass voltage current     min     max
---------------------------------------------------------------------------------
regulator-dummy                   0   11      0     0mV     0mA     0mV     0mV
   49000000.usb-otg                                                 0mV     0mV
   49000000.usb-otg                                                 0mV     0mV
   5a000000.dsi.0                                                   0mV     0mV
   vddcore                        0    0      0  1200mV     0mA   800mV  1350mV
   vdd_ddr                        0    1      0  1350mV     0mA  1350mV  1350mV
      vtt_ddr                     0    0      0   675mV     0mA   675mV   675mV
   vdd                            0    1      0  3300mV     0mA  3300mV  3300mV
      vref                        1    1      0  2500mV     0mA  2500mV  2500mV
         48003000.adc                                               0mV     0mV
   v3v3                           0    5      0  3300mV     0mA  3300mV  3300mV
      0-004a                                                        0mV     0mV
      58007000.sdmmc                                             3300mV  3300mV
      58005000.sdmmc                                             3300mV  3300mV
      v1v8_audio                  0    3      0  1800mV     0mA  1800mV  1800mV
         0-004a                                                     0mV     0mV
         0-004a                                                     0mV     0mV
         0-004a                                                     0mV     0mV
      v1v2_hdmi                   0    1      0  1200mV     0mA  1200mV  1200mV
         0-0039                                                     0mV     0mV
   v3v3_hdmi                      0    1      0  3300mV     0mA  3300mV  3300mV
      0-0039                                                        0mV     0mV
   vdd_usb                        2    2      0  3300mV     0mA  3300mV  3300mV
      phy-5a006000.usbphyc.1                                        0mV     0mV
      phy-5a006000.usbphyc.0                                        0mV     0mV
   vdda                           0    0      0  2900mV     0mA  2900mV  2900mV
   bst_out                        0    2      0  5000mV     0mA     0mV     0mV
      vbus_otg                    0    0      0  5000mV     0mA     0mV     0mV
      vbus_sw                     0    0      0  5000mV     0mA     0mV     0mV
reg11                             1    1      0  1100mV     0mA  1100mV  1100mV
   5a006000.usbphyc                                             0mV     0mV
reg18                             2    2      0  1800mV     0mA  1800mV  1800mV
   5a006000.usbphyc                                             0mV     0mV
   5a000000.dsi                                                 0mV     0mV
usb33                             1    1      0  3300mV     0mA  3300mV  3300mV
   49000000.usb-otg                                             0mV     0mV
vref_ddr                          0    0      0   675mV     0mA     0mV     0mV
```

- To list the pin control settings:

```
Board $> cat /sys/kernel/debug/pinctrl/soc\:pin-controller@50002000/pinconf-pins
Pin config settings per pin
Format: pin (name): configs
pin 0 (PA0): analog
pin 1 (PA1): alternate 11 (ETH1_GMII_RX_CLK ETH1_MII_RX_CLK ETH1_RGMII_RX_CLK
ETH1_RMII_REF_CLK) - push pull - floating - low speed
pin 2 (PA2): alternate 11 (ETH1_MDIO) - push pull - floating - very high speed
pin 3 (PA3): alternate 14 (LCD_B5) - push pull - floating - high speed
pin 4 (PA4): analog
pin 5 (PA5): analog
pin 6 (PA6): analog
pin 7 (PA7): alternate 11 (ETH1_GMII_RX_DV ETH1_MII_RX_DV ETH1_RGMII_RX_CTL
ETH1_RMII_CRS_DV) - push pull - floating - low speed
pin 8 (PA8): analog
pin 9 (PA9): analog
pin 10 (PA10): output - high - push pull - floating - low speed
pin 11 (PA11): analog
pin 12 (PA12): analog
pin 13 (PA13): output - low - open drain - floating - very high speed
pin 14 (PA14): input - high - floating
pin 15 (PA15): analog
pin 16 (PB0): alternate 11 (ETH1_GMII_RXD2 ETH1_MII_RXD2 ETH1_RGMII_RXD2) - push pull -
floating - low speed
pin 17 (PB1): alternate 11 (ETH1_GMII_RXD3 ETH1_MII_RXD3 ETH1_RGMII_RXD3) - push pull -
floating - low speed
pin 18 (PB2): alternate 8 (UART4_RX) - push pull - floating - low speed
pin 19 (PB3): alternate 9 (SDMMC2_D2) - push pull - pull up - very high speed
pin 20 (PB4): alternate 9 (SDMMC2_D3) - push pull - pull up - very high speed
pin 21 (PB5): analog
pin 22 (PB6): alternate 5 (CEC) - open drain - floating - low speed
pin 23 (PB7): analog
pin 24 (PB8): alternate 14 (LCD_B6) - push pull - floating - high speed
pin 25 (PB9): analog
pin 26 (PB10): analog
pin 27 (PB11): alternate 11 (ETH1_GMII_TX_EN ETH1_MII_TX_EN ETH1_RGMII_TX_CTL ETH1_RMII_TX
_EN) - push pull - floating - very high speed
pin 28 (PB12): analog
pin 29 (PB13): analog
pin 30 (PB14): alternate 9 (SDMMC2_D0) - push pull - pull up - very high speed
pin 31 (PB15): alternate 9 (SDMMC2_D1) - push pull - pull up - very high speed
pin 32 (PC0): alternate 14 (LCD_R5) - push pull - floating - high speed
pin 33 (PC1): alternate 11 (ETH1_MDC) - push pull - floating - very high speed
pin 34 (PC2): alternate 11 (ETH1_GMII_TXD2 ETH1_MII_TXD2 ETH1_RGMII_TXD2) - push pull -
floating - very high speed
pin 35 (PC3): analog
pin 36 (PC4): alternate 11 (ETH1_GMII_RXD0 ETH1_MII_RXD0 ETH1_RGMII_RXD0 ETH1_RMII_RXD0)
- push pull - floating - low speed
pin 37 (PC5): alternate 11 (ETH1_GMII_RXD1 ETH1_MII_RXD1 ETH1_RGMII_RXD1 ETH1_RMII_RXD1)
- push pull - floating - low speed
pin 38 (PC6): analog
pin 39 (PC7): analog
pin 40 (PC8): analog
pin 41 (PC9): analog
pin 42 (PC10): analog
pin 43 (PC11): analog
pin 44 (PC12): analog
pin 45 (PC13): analog
pin 46 (PC14): analog
pin 47 (PC15): analog
pin 48 (PD0): analog
pin 49 (PD1): analog
pin 50 (PD2): analog
pin 51 (PD3): alternate 7 (USART2_CTS USART2_NSS) - push pull - floating - low speed
pin 52 (PD4): alternate 7 (USART2_RTS USART2_DE) - push pull - floating - very high speed
pin 53 (PD5): alternate 7 (USART2_TX) - push pull - floating - very high speed
```

```
pin 54 (PD6): alternate 7 (USART2_RX) - push pull - floating - low speed
pin 55 (PD7): analog
pin 56 (PD8): alternate 14 (LCD_B7) - push pull - floating - high speed
pin 57 (PD9): alternate 14 (LCD_B0) - push pull - floating - high speed
pin 58 (PD10): alternate 14 (LCD_B3) - push pull - floating - high speed
pin 59 (PD11): output - low - push pull - floating - low speed
pin 60 (PD12): alternate 5 (I2C1_SCL) - open drain - floating - low speed
pin 61 (PD13): analog
pin 62 (PD14): analog
pin 63 (PD15): analog
pin 64 (PE0): alternate 10 (SAI2_MCLK_A) - push pull - floating - high speed
pin 65 (PE1): analog
pin 66 (PE2): alternate 11 (ETH1_GMII_TXD3 ETH1_MII_TXD3 ETH1_RGMII_TXD3) - push pull -
floating - very high speed
pin 67 (PE3): alternate 9 (SDMMC2_CK) - push pull - pull up - very high speed
pin 68 (PE4): output - high - push pull - floating - low speed
pin 69 (PE5): alternate 14 (LCD_G0) - push pull - floating - high speed
pin 70 (PE6): alternate 14 (LCD_G1) - push pull - floating - high speed
pin 71 (PE7): analog
pin 72 (PE8): analog
pin 73 (PE9): analog
pin 74 (PE10): analog
pin 75 (PE11): analog
pin 76 (PE12): analog
pin 77 (PE13): analog
pin 78 (PE14): analog
pin 79 (PE15): alternate 14 (LCD_R7) - push pull - floating - high speed
pin 80 (PF0): analog
pin 81 (PF1): analog
pin 82 (PF2): input - high - floating
pin 83 (PF3): analog
pin 84 (PF4): analog
pin 85 (PF5): analog
pin 86 (PF6): analog
pin 87 (PF7): analog
pin 88 (PF8): analog
pin 89 (PF9): analog
pin 90 (PF10): alternate 14 (LCD_DE) - push pull - floating - high speed
pin 91 (PF11): alternate 10 (SAI2_SD_B) - push pull - floating - low speed
pin 92 (PF12): analog
pin 93 (PF13): analog
pin 94 (PF14): analog
pin 95 (PF15): alternate 5 (I2C1_SDA) - open drain - floating - low speed
pin 96 (PG0): analog
pin 97 (PG1): input - high - floating
pin 98 (PG2): analog
pin 99 (PG3): analog
pin 100 (PG4): alternate 11 (ETH1_GMII_GTX_CLK ETH1_RGMII_GTX_CLK) - push pull - floating
- very high speed
pin 101 (PG5): alternate 11 (ETH1_GMII_CLK125 ETH1_RGMII_CLK125) - push pull - floating -
very high speed
pin 102 (PG6): alternate 10 (SDMMC2_CMD) - push pull - pull up - very high speed
pin 103 (PG7): alternate 14 (LCD_CLK) - push pull - floating - high speed
pin 104 (PG8): analog
pin 105 (PG9): output - high - push pull - floating - low speed
pin 106 (PG10): alternate 14 (LCD_B2) - push pull - floating - high speed
pin 107 (PG11): alternate 6 (UART4_TX) - push pull - floating - low speed
pin 108 (PG12): alternate 14 (LCD_B1) - push pull - floating - high speed
pin 109 (PG13): alternate 11 (ETH1_GMII_TXD0 ETH1_MII_TXD0 ETH1_RGMII_TXD0
ETH1_RMII_TXD0) - push pull - floating - very high speed
pin 110 (PG14): alternate 11 (ETH1_GMII_TXD1 ETH1_MII_TXD1 ETH1_RGMII_TXD1
ETH1_RMII_TXD1) - push pull - floating - very high speed
pin 111 (PG15): analog
pin 112 (PH0): analog
pin 113 (PH1): analog
pin 114 (PH2): alternate 14 (LCD_R0) - push pull - floating - high speed
pin 115 (PH3): alternate 14 (LCD_R1) - push pull - floating - high speed
```

```
pin 116 (PH4): output - high - push pull - floating - low speed
pin 117 (PH5): analog
pin 118 (PH6): analog
pin 119 (PH7): analog
pin 120 (PH8): alternate 14 (LCD_R2) - push pull - floating - high speed
pin 121 (PH9): alternate 14 (LCD_R3) - push pull - floating - high speed
pin 122 (PH10): alternate 14 (LCD_R4) - push pull - floating - high speed
pin 123 (PH11): analog
pin 124 (PH12): alternate 14 (LCD_R6) - push pull - floating - high speed
pin 125 (PH13): alternate 14 (LCD_G2) - push pull - floating - high speed
pin 126 (PH14): alternate 14 (LCD_G3) - push pull - floating - high speed
pin 127 (PH15): alternate 14 (LCD_G4) - push pull - floating - high speed
pin 128 (PI0): alternate 14 (LCD_G5) - push pull - floating - high speed
pin 129 (PI1): alternate 14 (LCD_G6) - push pull - floating - high speed
pin 130 (PI2): alternate 14 (LCD_G7) - push pull - floating - high speed
pin 131 (PI3): analog
pin 132 (PI4): alternate 14 (LCD_B4) - push pull - floating - high speed
pin 133 (PI5): alternate 10 (SAI2_SCK_A) - push pull - floating - medium speed
pin 134 (PI6): alternate 10 (SAI2_SD_A) - push pull - floating - medium speed
pin 135 (PI7): alternate 10 (SAI2_FS_A) - push pull - floating - medium speed
pin 136 (PI8): analog
pin 137 (PI9): alternate 14 (LCD_VSYNC) - push pull - floating - high speed
pin 138 (PI10): alternate 14 (LCD_HSYNC) - push pull - floating - high speed
pin 139 (PI11): analog
```

- To obtain the GPIO settings:

```
Board $> cat /sys/kernel/debug/gpio
gpiochip0: GPIOs 0-15, parent: platform/soc:pin-controller@50002000, GPIOA:
 gpio-10  (                    |reset               ) out lo

gpiochip1: GPIOs 16-31, parent: platform/soc:pin-controller@50002000, GPIOB:

gpiochip2: GPIOs 32-47, parent: platform/soc:pin-controller@50002000, GPIOC:

gpiochip3: GPIOs 48-63, parent: platform/soc:pin-controller@50002000, GPIOD:
 gpio-59  (                    |heartbeat           ) out lo

gpiochip4: GPIOs 64-79, parent: platform/soc:pin-controller@50002000, GPIOE:
 gpio-68  (                    |reset               ) out hi

gpiochip5: GPIOs 80-95, parent: platform/soc:pin-controller@50002000, GPIOF:

gpiochip6: GPIOs 96-111, parent: platform/soc:pin-controller@50002000, GPIOG:
 gpio-105 (                    |reset               ) out hi

gpiochip7: GPIOs 112-127, parent: platform/soc:pin-controller@50002000, GPIOH:
 gpio-116 (                    |reset               ) out hi

gpiochip8: GPIOs 128-143, parent: platform/soc:pin-controller@50002000, GPIOI:

gpiochip9: GPIOs 400-415, parent: platform/soc:pin-controller-z@54004000, GPIOZ:
```

4 To go further

4.1 Regmap (register map) cache

Debugfs contains a register cache (mirror) for drivers/peripherals based on regmap API[3].

Regmap is a register map abstraction API of the Linux kernel. It is mainly used for serial bus device drivers (I2C and SPI), but can also be used for internal peripheral drivers.

Many of these drivers contain some very similar code for accessing connected-hardware device registers.
The regmap kernel API proposes a solution which factors out this code from the drivers, saving code and making it much easier to share infrastructure.

- Path of the regmap register cache in debugfs

```
Board $> cd /sys/kernel/debug/regmap
```

- List of devices for which drivers are based on regmap API:

```
Board $> ls -la .
total 0
drwxr-xr-x 18 root root 0 Jan  1  1970 .
drwx------ 32 root root 0 Jan  1  1970 ..
drwxr-xr-x  2 root root 0 Jan  1  1970 0-001b
drwxr-xr-x  2 root root 0 Jan  1  1970 0-0042
drwxr-xr-x  2 root root 0 Jan  1  1970 2-0033
drwxr-xr-x  2 root root 0 Jan  1  1970 40004000.timer
drwxr-xr-x  2 root root 0 Jan  1  1970 4000d000.audio-controller
drwxr-xr-x  2 root root 0 Nov 30 12:19 40016000.cec
drwxr-xr-x  2 root root 0 Jan  1  1970 40017000.dac
drwxr-xr-x  2 root root 0 Jan  1  1970 4400b004.audio-controller
drwxr-xr-x  2 root root 0 Jan  1  1970 4400b024.audio-controller
drwxr-xr-x  2 root root 0 Jan  1  1970 4400d000.dfsdm
drwxr-xr-x  2 root root 0 Jan  1  1970 50027004.audio-controller
drwxr-xr-x  2 root root 0 Jan  1  1970 dummy-interrupt-controller@5000d000
drwxr-xr-x  2 root root 0 Jan  1  1970 dummy-pwr@50001000
drwxr-xr-x  2 root root 0 Jan  1  1970 dummy-rcc@50000000
drwxr-xr-x  2 root root 0 Jan  1  1970 dummy-syscon@50020000
drwxr-xr-x  2 root root 0 Jan  1  1970 dummy-tamp@5c00a000
```

- Check for the corresponding driver: example for **2-0033** regmap entry

```
Board $> cat 2-0033/name
stpmic1
```

Then, on the STM32MPU device tree files, you can check for the device, and the corresponding driver

```
pmic: stpmic@33 {
        compatible = "st,stpmic1";
        reg = <0x33>;
        interrupts-extended = <&exti_pwr 55 IRQ_TYPE_EDGE_FALLING>;
        interrupt-controller;
        #interrupt-cells = <2>;
        status = "okay";
```

- Regmap entries stating by ***dummy-***

This kind of entry is set when there are no associated /dev entries (devtmpfs).

To find the corresponding node in the device tree, you can look for the suffix name after ***dummy-***:

Example for dummy-**interrupt-controller@5000d000**

```
exti: interrupt-controller@5000d000 {
        compatible = "st,stm32mp1-exti", "syscon";
        interrupt-controller;
        #interrupt-cells = <2>;
        reg = <0x5000d000 0x400>;
};
```

- List of registers: example for **2-0033** regmap entry

```
Board $> cat 2-0033/registers
01: 10
02: 00
03: 00
04: 00
05: 60
06: 10
10: 04
11: 00
12: 00
13: 00
14: 00
15: c0
16: 00
17: 00
18: 04
19: 00
1a: 00
1b: 00
1c: 00
1d: 0f
1e: 04
20: 61
21: 79
22: d9
23: d9
24: 01
25: 51
26: 4c
27: 7d
28: 01
29: 51
2a: 25
30: 61
31: 50
32: d9
33: d9
34: 00
35: 24
36: 24
37: 24
38: 01
39: 51
3a: 04
40: 35
```

```
50: 00
51: 00
52: 00
53: 80
60: 00
61: 00
62: 00
63: 00
70: 00
71: 00
72: 00
73: 00
80: fc
81: 8f
82: c4
83: cf
90: fc
91: 8f
92: c4
93: cf
a0: 03
a1: 70
a2: 3b
a3: 00
b0: 00
b1: 00
b2: 00
b3: 80
```

The first column representes the register address, and the second column the register value.

For register definitions, please refer to the device specification.

5 References

- Documentation/filesystems/debugfs.txt
- https://opensourceforu.com/2010/10/debugging-linux-kernel-with-debugfs
- https://elinux.org/images/a/a3/Regmap-_The_Power_of_Subsystems_and_Abstractions.pdf

- Useful external links

Document link	Document Type	Description
Wikipedia debugfs	Standard	Wikipedia
Guide to debugfs	Standard	Guide
regmap: Reducing the Redundancy in Linux Code	Article	Guide

Random Access Memory (Early computer memories generally hadserial access. Memories where any given address can be accessed when desired were then called "random access" to distinguish them from the memories where contents can only be accessed in a fixed order. The term is used today for volatile random-acces ssemiconductor memories.)

Process File System (See https://en.wikipedia.org/wiki/Procfs for more details)

System File System (See https://en.wikipedia.org/wiki/Sysfs for more details)

Debug File System (See https://en.wikipedia.org/wiki/Debugfs for more details)

Linux® is a registered trademark of Linus Torvalds.

Receive

Consumer Electronics Control (HDMI standard)

Transmit

Compatibility Test Suite (Android specific) or Clear to send (in UART context)

Serial clock line

Secure digital

Serial DAta line

General-Purpose Input/Output (A realization of open ended transmission between devices on an embedded level. These pins available on a processor can be programmed to be used to either accept input or provide output to external devices depending on user desires and applications requirements.)

Application programming interface

Inter-Integrated Circuit (Bi-directional 2-wire bus standard for efficient inter-IC control.)

Serial Peripheral Interface

Stable: 02.11.2020 - 10:48 / Revision: 19.10.2020 - 12:09

A quality version of this page, approved on *2 November 2020*, was based off this revision.

Contents

1 Introduction

As prerequisite to reading this article, please refer to the Dmesg and Linux kernel log page.

"Dynamic debug is designed to allow you to dynamically enable/disable kernel code to obtain additional kernel information. Currently, if **CONFIG_DYNAMIC_DEBUG** is set, all **pr_debug()/dev_dbg()** calls can be dynamically enabled per-callsite." extracted from the Linux kernel documentation[1].

The related debugfs entry is usually:

```
/sys/kernel/debug/dynamic_debug/control
```

Note that the verbose **dev_vdbg()** calls cannot be dynamically activated.

When the dynamic debug traces are activated, the trace results are printed in **dmesg (or /proc/kmsg)**, and in the console if console loglevel is set to 8.

2 More technical information

The dynamic debug trace configuration is done through a **control** file in the **debugfs** filesystem: *<debugfs>/dynamic_debug /control*

The command includes keywords and flag elements (for details see the Linux kernel documentation[1]).

- Keywords

Possible keywords are:

```
func : function name
file : source filename
module : module name
format : format string
line : line number (including ranges of line numbers)
```

The colored keywords above are illustrated by examples in the next chapter.

- Flags

The flag specification comprises a change operation followed by one or more flag characters. The change operation is one of the characters:

```
- : remove the given flags
+ : add the given flags
= : set the flags to the given flags
```

Possible flags are:

```
f : Include the function name in the printed message
l : Include line number in the printed message
m : Include module name in the printed message
p : Causes a printk() message to be emitted to dmesg
t : Include thread ID in messages not generated from interrupt context
```

3 Examples

- Track all dev_*dbg/pr_debug() in a **file** (you can add several files if necessary):

```
Board $> mount -t debugfs none /sys/kernel/debug
Board $> echo "file stm32-adc.c +p" > /sys/kernel/debug/dynamic_debug/control
```

Note that just the file name or full file path can be given, here *stm32-adc.c* or *drivers/iio/adc/stm32-adc.c*

- Track only one **line** with dev_dbg() in a **file** (you can add several files and several lines if necessary, please use the last line number of the function call):

```
Board $> echo "file stm32-adc.c line 1438 +p" > /sys/kernel/debug/dynamic_debug/control
```

- For an entire "**module** (module means ~.ko, so not applicable for a statically linked driver)":

```
Board $> echo "module cfg80211 +p" > /sys/kernel/debug/dynamic_debug/control
```

- If you want to list all available traces *(warning: it is a long file so you may need to use "tee" or another solution to save it)*:

```
Board $> cat /sys/kernel/debug/dynamic_debug/control | tee /tmp/dynamic_log.log
```

- For instance, if you are looking for a particular **file** to find a particular **line**:

```
Board $> cat /sys/kernel/debug/dynamic_debug/control | grep adc
drivers/iio/adc/stm32-adc.c:1515 [stm32_adc]stm32_adc_conf_scan_seq =p "%s chan %d to %s%
d\012"
drivers/iio/adc/stm32-adc.c:1438 [stm32_adc]stm32_adc_awd_set =p "%s chan%d htr:%d ltr:%
d\012"
drivers/iio/adc/stm32-adc.c:2182 [stm32_adc]stm32_adc_dma_start =p "%s size=%d watermark=%
d\012"
drivers/iio/adc/stm32-adc.c:2304 [stm32_adc]stm32_adc_trigger_handler =p "%s bufi=%d\012"
drivers/iio/adc/stm32-adc.c:2443 [stm32_adc]stm32_adc_chan_of_init =p "Configured to use
injected\012"
drivers/iio/adc/stm32-adc.c:2364 [stm32_adc]stm32_adc_of_get_resolution =p "Using %u bits
resolution\012"
```

- Multiple commands can be written together, separated by ';' or '\n'.

```
Board $> echo "file stm32-adc.c +p; file stm32-adc-core.c +p" > /sys/kernel/debug
/dynamic_debug/control
```

- A another method is to use a wildcard. The match rule supports * (matches zero or more characters) and ? (matches exactly one character). For example, you can match all USB drivers:

```
Board $> echo "file drivers/usb/* +p" > /sys/kernel/debug/dynamic_debug/control
```

4 Synchronous tracing on the console

In the case of a crash, or impossibility to call dmesg, it is sometimes useful to have traces synchronously emitted on the console.
Only error, warning and informational traces are emitted synchronously on the console (that is, loglevel=5), so if you need to see the lower level traces too, you need to change the console loglevel to "8".

```
<enable the conditional traces>
Board $> echo 8 > /proc/sys/kernel/printk
or
Board $> dmesg -n 8
or
Board $> dmesg -n debug
```

Please follow this article to get a serial console for the target: How to get Terminal

Warning

As all traces are now synchronously emitted, real-time is affected

If you want to return to the default console log level, you have to get this default value from the procfs entry */proc/sys /kernel/printk*:

```
Board $> cat /proc/sys/kernel/printk
8       4       1       7
Board $> dmesg -n 7
Board $> cat /proc/sys/kernel/printk
7       4       1       7
```

5 Debug messages during boot process

In order to activate debug messages during the boot process, even before userspace and debugfs exist, use the kernel's command-line parameter: **dyndbg**

For instance, the kernel *bootargs* can be modified in the following ways:

- Mount a boot partition from the Linux kernel console, and then update the extlinux.conf file using the vi editor (see man page [2], or introduction page[3]). For example:

```
Board $> mount /dev/mmcblk0p4 /boot
Board $> vi /boot/mmc0_stm32mp157c-ev1_extlinux/extlinux.conf
```

or

- Edit the extlinux.conf file by using UMS (USB Mass Storage): see How to use USB mass storage in U-Boot for details.

To mount partitions (mmc 0:microSD card / mmc 1: eMMC):

- Press any key to stop at U-Boot execution when booting the board.

```
Board $> ...
Board $> Hit any key to stop autoboot:  0
Board $> STM32MP>
```

- Then

```
STM32MP> ums 0 mmc 0
```

- Check for the boot partition mounted on your host PC (/media/*$USER*/bootfs)
- Edit the extlinux file corresponding to your setup (/media/*$USER*/bootfs/mmc0_extlinux/stm32mp157f-dk2_extlinux.conf

- Update the kernel command line, adding the dyndbg parameter:

```
root=PARTUUID=e91c4e10-16e6-4c0e-bd0e-77becf4a3582 rootwait rw console=ttySTM0,115200 dynd
bg="file drivers/usb/core/hub.c +p"
```

Save and quit file update, and then reboot the board.

Note: to display these debug messages in the console, in addition to the dmesg, add **loglevel=8** in the kernel command line.

- Reboot the board and check for a kernel command-line, and that debug messages are present in the dmesg output

6 References

- 1.01.1 Documentation/admin-guide/dynamic-debug-howto.rst
- http://ex-vi.sourceforge.net/vi.html
- http://ex-vi.sourceforge.net/viin/paper.html

- Useful external links

Document link	Document Type	Description
The dynamic debugging interface (lwn.net)	User guide	http://lwn.net
Documentation/dynamic-debug-howto.txt (lwn.txt)	User guide	http://lwn.net
Dynamic debug howto (kernel.org)	Standard	http://www.kernel.org

Debug File System (See https://en.wikipedia.org/wiki/Debugfs for more details)

Process File System (See https://en.wikipedia.org/wiki/Procfs for more details)

User-space Mode Setting

former spelling for e•MMC ('e' in italic)

Das U-Boot -- the Universal Boot Loader (see U-Boot_overview)

Stable: 25.09.2020 - 09:10 / Revision: 25.09.2020 - 09:09

A quality version of this page, approved on *25 September 2020*, was based off this revision.

Contents

1 Article purpose

The purpose of this article is to:

- briefly introduce the RCC peripheral and its main features
- indicate the level of security supported by this hardware block
- explain, when necessary, how to configure the RCC peripheral.

2 Peripheral overview

The **RCC** peripheral is used to control the internal peripherals, as well as the **reset** signals and **clock** distribution. The RCC gets several internal (LSI, HSI and CSI) and external (LSE and HSE) clocks. They are used as clock sources for the hardware blocks, either directly or indirectly, via the four PLLs (PLL1, PLL2, PLL3 and PLL4) that allow to achieve high frequencies.

2.1 Features

Refer to the STM32MP15 reference manuals for the complete list of features, and to the software components, introduced below, to see which features are really implemented.

2.2 Security support

The RCC is a **secure** peripheral. There are two levels of security, which are controlled via two bits in the RCC_TZCR register (only accessible in secure mode):

- **TZEN** allows to set some RCC registers in secure mode, in particular registers for configuring PLL1 and PLL2, in order to secure a TrustZone perimeter for the Cortex®-A7 secure core and its peripherals.
- **MCKPROT** allows extending the TZEN secure clock control perimeter to PLL3 and to the MCU subsystem, so to the Cortex® -M4 and its bus clock.

Please note that all RCC registers can be read from the non-secure world.

3 Peripheral usage and associated software

3.1 Boot time

The RCC security level differs for each boot chain:

- the trusted boot chain sets TZEN to 1 an MCKPROT to 0
- the basic boot chain sets TZEN to 0 an MCKPROT to 0

The RCC is used by all the boot components: the ROM code, the FSBL, the SSBL and up to the Linux® kernel. Nevertheless, the main initialization step is performed by the FSBL that is responsible for the clock tree initialization: it consists in configuring all the input clocks, the PLL and the clock sources that are selected as kernel clocks for all peripherals. The whole configuration is carried out by the device tree.

The STM32CubeMX tool allows configuring in one place the clock tree that will be applied at boot time and used at runtime, so it is highly recommended to use it to generate your device tree. Moreover, the STM32CubeMX integrates all the information documented in the STM32MP15 reference manuals, making this configuration step straighforward.

3.2 Runtime

3.2.1 Overview

The RCC peripheral is shared at runtime:

- the Arm®Cortex®-A7 secure core controls all the secure registers (refer to TZEN and MCKPROT bit descriptions) through the RCC OP-TEE driver. The access to some secure registers from the Cortex®-A7 non-secure core can be achieved via runtime secure services implemented in the secure monitor (from the OP-TEE if it is present, otherwise from the TF-A).
- the Arm®Cortex®-A7 non-secure core controls the clock management via the clock framework, and the reset management via the reset framework in Linux®.
- the Arm®Cortex®-M4 core controls all the clock and reset managements in STM32Cube with the RCC HAL driver

Concurrent control from each context is possible because the above managements are performed via independent registers.

3.2.2 Software frameworks

Domain	Peripheral	Software frameworks			Comment
Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Power & Thermal	RCC	OP-TEE RCC driver	Reset framework Clock framework	STM32Cube RCC driver	

3.2.3 Peripheral configuration

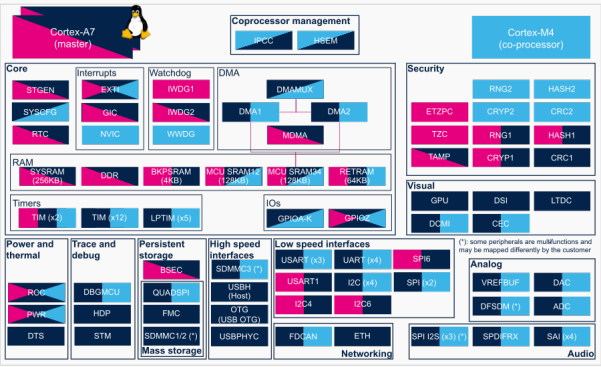
The configuration is applied by the firmware running in the context to which the peripheral is assigned. The configuration can be done alone via the STM32CubeMX tool for all internal peripherals, and then manually completed (particularly for external peripherals), according to the information given in the corresponding software framework article.

3.2.4 Peripheral assignment

Check boxes illustrate the possible peripheral allocations supported by STM32 MPU Embedded Software:

- means that the peripheral can be assigned () to the given runtime context.
- is used for system peripherals that cannot be unchecked because they are statically connected in the device.

Refer to How to assign an internal peripheral to a runtime context for more information on how to assign peripherals manually or via STM32CubeMX.
The present chapter describes STMicroelectronics recommendations or choice of implementation. Additional possiblities might be described in STM32MP15 reference manuals
.



Internal peripherals

Domain	Periphera l	Runtime allocation				Comment
Instance	Cortex-A7 secure (OP-TEE)	Cortex-A7 non-secure (Linux)	Cortex-M4 (STM32Cube)			
Power & Thermal	RCC	RCC				

4 How to go further

The RCC is interfaced with the HDP internal peripheral, thus offering the flexibility to monitor the main RCC state signals on the debug pins.

Please refer to the STM32MP15 reference manuals for the full list of signals that can be monitored.

5 References

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

Multi Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed External oscillator (STM32 clock source)

TrustZone®

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Cortex®

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Read Only Memory

First Stage Boot Loader

Second Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Arm® is a registered trademark of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Open Portable Trusted Execution Environment

Stable: 25.09.2020 - 09:10 / Revision: 25.09.2020 - 09:07

A quality version of this page, approved on *25 September 2020*, was based off this revision.

All the peripherals receive one or several clocks that are generated via RCC internal peripheral. RCC relies on several clocks sources (LSI, LSE, HSI, HSE, CSI) and four PLL in order to provide adequate input frequencies to all the peripherals. The clock tree covers all the system clock distribution aspects, from the clock sources to the consumer peripherals (internal and external), except clock gating management that is locally controlled by each peripheral driver.

Contents

1 Overview

The clock tree is managed via RCC internal peripheral hardware block and it is configured at different steps from the Cortex-A7:

- When the device is **reset**, all RCC registers take their reset values: the four PLL are disabled and most of the clock source selectors are pointing to the HSI.
- The ROM Code configures the minimum clock tree needed to boot on the selected boot device.
- The **FSBL** (Cf. Boot chain overview) completely configures the clock tree as expected, all the way up to Linux, thanks to the configuration given in the device tree.
  - U-Boot SPL (basic boot chain) binding is available in doc/device-tree-bindings/clock/st,stm32mp1.txt
  - TF-A (trusted boot chain) binding follows the same syntax as U-Boot SPL
- **Linux** is responsible for the clock tree at runtime and it may partly modify it thanks to information taken from the device tree:
  - Linux binding is available in Documentation/devicetree/bindings/clock/clock-bindings.txt (and surrounding files): 'fixed-clock' compatible, 'clocks' and 'assigned-clocks' properties are important concepts to understand the management of clocks providers/consumers.

Notice that STM32Cube (running on the Cortex-M4) does not control the clock tree setup so the configuration applied by the Cortex-A7 must provide suitable clocks to Cortex-M4, like it is explained in resource manager description. One exception to this rule is the engineering boot, allowing to directly load (and debug) the Cortex-M4 whereas the Cortex-A7 execution is stalled in the ROM code.

2 How to build a clock tree?

Building a clock tree is quite complex as it needs to take into account the constraints set by each internal and external peripheral, including external clock sources.

We encourage the use of STM32CubeMX to build the clock tree, and avoid having to know all internal peripherals details: the tool allows to select the peripherals that will be present on the board, fix the clock sources frequencies and automatically find an optimized clock tree. It is then able to generate the device tree that is directly consumed by the boot chain and Linux.

3 ST boards clock tree

Information

The PLL1 configuration is optional in the ecosystem release v2.0.0 and your board may support dynamic voltage and frequency scaling (DVFS).

See How to change the CPU frequency article for more information about the CPU frequency setting.

3.1 STM32MP157x-EV1 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-EV1 evaluation board. Linux eventual runtime modifications are not covered here.

3.1.1 Clock tree

The following table shows what STM32MP157x-EV1 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

Component	Parent	Frequency	Used?	Comment
LSI	N.A.	0.032000 MHz	yes	Mandatory for IWDG, DAC
DAC	LSI	0.032000 MHz	yes	
RNG1	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
RNG2	LSI	0.032000 MHz	yes	Input frequency should be as low as possible
IWDG1	LSI	0.032000 MHz	yes	
IWDG2	LSI	0.032000 MHz	yes	
LSE	N.A.	0.032768 MHz	yes	Mandatory for DTS
RTC	LSE	0.032768 MHz	yes	
TAMP	LSE	0.032768 MHz	yes	
CEC	LSE	0.032768 MHz	yes	
LPTIM4	LSE	0.032768 MHz	yes	
LPTIM5	LSE	0.032768 MHz	yes	
HSI	N.A.	64.000000 MHz	yes	
SPI4	HSI	64.000000 MHz	no	
SPI5	HSI	64.000000 MHz	no	
SPI6	HSI	64.000000 MHz	yes	
I2C4	HSI	64.000000 MHz	yes	PMIC
I2C6	HSI	64.000000 MHz	no	
I2C1	HSI	64.000000 MHz	no	
I2C2	HSI	64.000000 MHz	yes	Rpi and peripherals
I2C3	HSI	64.000000 MHz	no	
I2C5	HSI	64.000000 MHz	yes	Rpi
USART1	HSI	64.000000 MHz	yes	
USART2	HSI	64.000000 MHz	no	
USART3	HSI	64.000000 MHz	yes	Rpi
UART4	HSI	64.000000 MHz	yes	Linux console
UART5	HSI	64.000000 MHz	no	
USART6	HSI	64.000000 MHz	no	
UART7	HSI	64.000000 MHz	no	
UART8	HSI	64.000000 MHz	no	
MCO1	HSI	64.000000 MHz	no	Available so can be used
HSE	N.A.	24.000000 MHz	yes	

DSIPLL	HSE	125.000000 MHz	no	DSI DPHY PLL
DSIBL	DSIPLL	125.000000 MHz	no	DSI lanebyte clock
RTCDIV	HSE	1.000000 MHz	yes	Only used when RTC source is HSE
ck_per	HSE	24.000000 MHz	yes	
ADC	ck_per	24.000000 MHz	yes	Can use internal divider to be < 40MHz
PLL1	HSE	xxx MHz	yes	
PLL1P	PLL1	xxx MHz	yes	
MPUDIV	PLL1P	xxx MHz	yes	
Cortex-A7	PLL1P	xxx MHz	yes	650 MHz or 800 MHz, see the information box above
MCO2	Cortex-A7	xxx MHz	no	Available so can be used
PLL2	HSE	1066.000000 MHz	yes	
PLL2P	PLL2	266.500000 MHz	yes	
AXI	PLL2P	266.500000 MHz	yes	< 266MHz
FMC	AXI	266.500000 MHz	yes	NAND flash
QSPI	AXI	266.500000 MHz	yes	NOR flash
SYSRAM	AXI	266.500000 MHz	yes	
ROM	AXI	266.500000 MHz	yes	
AHB5	AXI	266.500000 MHz	yes	< 266MHz
CRYP1	AHB5	266.500000 MHz	yes	
HASH1	AHB5	266.500000 MHz	yes	
GPIOZ	AHB5	266.500000 MHz	yes	
BKPSRAM	AHB5	266.500000 MHz	yes	
AHB6	AXI	266.500000 MHz	yes	< 266MHz
CRC1	AHB6	266.500000 MHz	yes	
MDMA	AHB6	266.500000 MHz	yes	
USBH	AHB6	266.500000 MHz	yes	USB Host
APB4	AHB6	133.250000 MHz	yes	
APB5	AHB6	66.625000 MHz	yes	
BSEC	APB5	66.625000 MHz	yes	< 67MHz
ETZPC	APB5	66.625000 MHz	yes	
TZC	APB5	66.625000 MHz	yes	
DBGAPB	AXI	133.250000 MHz	yes	JTAG & Coresight
DBGMCU	DBGAPB	66.625000 MHz	yes	
STM	DBGAPB	66.625000 MHz	yes	
PLL2Q	PLL2	533.000000 MHz	yes	
GPU	PLL2Q	533.000000 MHz	yes	< 533MHz
PLL2R	PLL2	533.000000 MHz	yes	
DDR	PLL2R	533.000000 MHz	yes	< 533MHz
PLL3	HSE	417.755859 MHz	yes	
PLL3P	PLL3	208.877930 MHz	yes	
MLAHB	PLL3P	208.877930 MHz	yes	< 209MHz
Cortex-M4	MLAHB	208.877930 MHz	yes	
SRAM1	MLAHB	208.877930 MHz	yes	
SRAM2	MLAHB	208.877930 MHz	yes	
SRAM3	MLAHB	208.877930 MHz	yes	
RETRAM	MLAHB	208.877930 MHz	yes	
AHB1	MLAHB	208.877930 MHz	yes	< 209MHz
AHB2	MLAHB	208.877930 MHz	yes	< 209MHz
DMA1	AHB2	208.877930 MHz	yes	
DMA2	AHB2	208.877930 MHz	yes	
DMAMUX	AHB2	208.877930 MHz	yes	
APB1	MLAHB	104.438965 MHz	yes	
LPTIM1	APB1	104.438965 MHz	yes	
WWDG	APB1	104.438965 MHz	yes	
APB2	MLAHB	104.438965 MHz	yes	
TIM2	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM3	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM4	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM5	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM6	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM7	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM12	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM13	MLAHB	208.877930 MHz	yes	TIM Group 1
TIM14	MLAHB	208.877930 MHz	yes	TIM Group 1

```
        TIM1            |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM8            |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM15           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM16           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM17           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        APB3            |MLAHB    |  104.438965 MHz |  yes   |
          LPTIM2        |APB3     |  104.438965 MHz |  yes   |
          LPTIM3        |APB3     |  104.438965 MHz |  yes   |
          SYSCFG        |APB3     |  104.438965 MHz |  yes   |
          VREFBUF       |APB3     |  104.438965 MHz |  yes   |
          DTS           |APB3     |  104.438965 MHz |  yes   |
          HDP           |APB3     |  104.438965 MHz |  yes   |
        AHB3            |MLAHB    |  208.877930 MHz |  yes   | < 209MHz
          CRC2          |AHB3     |  208.877930 MHz |  yes   |
          CRYP2         |AHB3     |  208.877930 MHz |  yes   |
          HASH2         |AHB3     |  208.877930 MHz |  yes   |
          DCMI          |AHB3     |  208.877930 MHz |  yes   | Camera
          IPCC          |AHB3     |  208.877930 MHz |  yes   | Mailbox
        AHB4            |MLAHB    |  208.877930 MHz |  yes   | < 209MHz
          PWR           |AHB4     |  208.877930 MHz |  yes   |
          RCC           |AHB4     |  208.877930 MHz |  yes   |
          GPIOA-K       |AHB4     |  208.877930 MHz |  yes   |
          EXTI          |AHB4     |  208.877930 MHz |  yes   |
    PLL3Q               |PLL3     |   24.573874 MHz |  yes   |
      SPI1              |PLL3Q    |   24.573874 MHz |  no    |
      SPI2              |PLL3Q    |   24.573874 MHz |  no    |
      SPI3              |PLL3Q    |   24.573874 MHz |  no    |
      DFSDM             |PLL3Q    |   24.573874 MHz |  yes   | Digital micro
      SAI1              |PLL3Q    |   24.573874 MHz |  no    |
      SAI2              |PLL3Q    |   24.573874 MHz |  yes   | AudCodec 48kHz (use PLL3R
for 44.1kHz)
      SAI3              |PLL3Q    |   24.573874 MHz |  no    |
      SAI4              |PLL3Q    |   24.573874 MHz |  yes   | SPDIF TX 48kHz (use PLL3R
for 44.1kHz)
    PLL3R               |PLL3     |   11.290699 MHz |  yes   |
  PLL4                  |HSE      |  594.000000 MHz |  yes   |
    PLL4P               |PLL4     |   99.000000 MHz |  yes   |
      SDMMC1            |PLL4P    |   99.000000 MHz |  yes   | µSD card
      SDMMC2            |PLL4P    |   99.000000 MHz |  yes   | eMMC
      SDMMC3            |PLL4P    |   99.000000 MHz |  yes   |
      SPDIF             |PLL4P    |   99.000000 MHz |  yes   | SPDIF RX
    PLL4Q               |PLL4     |   74.250000 MHz |  yes   |
      LCD               |PLL4Q    |   74.250000 MHz |  yes   | LTDC & DSI display pixel
clock
    PLL4R               |PLL4     |   74.250000 MHz |  yes   |
      FDCAN             |PLL4R    |   74.250000 MHz |  yes   | Should be as high as
possible and < 100MHz
  STGEN                 |HSE      |   24.000000 MHz |  yes   |
  USBPHYC               |HSE      |   24.000000 MHz |  yes   | USB PHY Ctrl for USB Host
and OTG
    USBPLL              |USBPHYC  |   48.000000 MHz |  yes   |
      USBO              |USBPLL   |   48.000000 MHz |  yes   | USB OTG
CSI                     |N.A.     |    4.000000 MHz |  yes   | Mandatory for IO
compensation
ETH                     |N.A.     |    0.000000 MHz |  no    | ETH clocked by RGMII PHY
------------------------|---------|-----------------|--------
|-----------------------------------|
```

3.1.1.1 *Device tree*

Here are the corresponding device tree **rcc_clk** sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
        st,clksrc = <
                CLK_MPU_PLL1P
                CLK_AXI_PLL2P
                CLK_MCU_PLL3P
                CLK_PLL12_HSE
                CLK_PLL3_HSE
                CLK_PLL4_HSE
                CLK_RTC_LSE
                CLK_MCO1_DISABLED
                CLK_MCO2_DISABLED
        >;

        st,clkdiv = <
                1 /*MPU*/
                0 /*AXI*/
                0 /*MCU*/
                1 /*APB1*/
                1 /*APB2*/
                1 /*APB3*/
                1 /*APB4*/
                2 /*APB5*/
                23 /*RTC*/
                0 /*MCO1*/
                0 /*MCO2*/
        >;

        st,pkcs = <
                CLK_CKPER_HSE
                CLK_FMC_ACLK
                CLK_QSPI_ACLK
                CLK_ETH_DISABLED
                CLK_SDMMC12_PLL4P
                CLK_DSI_DSIPLL
                CLK_STGEN_HSE
                CLK_USBPHY_HSE
                CLK_SPI2S1_PLL3Q
                CLK_SPI2S23_PLL3Q
                CLK_SPI45_HSI
                CLK_SPI6_HSI
                CLK_I2C46_HSI
                CLK_SDMMC3_PLL4P
                CLK_USBO_USBPHY
                CLK_ADC_CKPER
                CLK_CEC_LSE
                CLK_I2C12_HSI
                CLK_I2C35_HSI
                CLK_UART1_HSI
                CLK_UART24_HSI
                CLK_UART35_HSI
                CLK_UART6_HSI
                CLK_UART78_HSI
                CLK_SPDIF_PLL4P
                CLK_FDCAN_PLL4R
                CLK_SAI1_PLL3Q
                CLK_SAI2_PLL3Q
                CLK_SAI3_PLL3Q
                CLK_SAI4_PLL3Q
                CLK_RNG1_LSI
                CLK_RNG2_LSI
                CLK_LPTIM1_PCLK1
                CLK_LPTIM23_PCLK3
                CLK_LPTIM45_LSE
        >;

        /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
```

```
        pll2: st,pll@1 {
                cfg = < 2 65 1 0 0 PQR(1,1,1) >;
                frac = < 0x1400 >;
                u-boot,dm-pre-reloc;
        };

        /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */
        pll3: st,pll@2 {
                cfg = < 1 33 1 16 36 PQR(1,1,1) >;
                frac = < 0x1a04 >;
                u-boot,dm-pre-reloc;
        };

        /* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
        pll4: st,pll@3 {
                cfg = < 3 98 5 7 7 PQR(1,1,1) >;
                u-boot,dm-pre-reloc;
        };
```

3.2 STM32MP157x-DK2 case

This chapter shows the boot time clock tree set by the FSBL on STM32MP157x-DK2 DISCO board. Linux eventual runtime modifications are not covered here.

3.2.1 Clock tree

The following table shows what STM32MP157x-DK2 clock tree looks like, as a result of the boot chain execution with the device tree built with STM32CubeMX.

```
Component               | Parent  |   Frequency     | Used?  |
Comment                           |
------------------------|---------|-----------------|--------
|-----------------------------------|
LSI                     |N.A.     |    0.032000 MHz |  yes   | Mandatory for IWDG, DAC
  DAC                   |LSI      |    0.032000 MHz |  no    |
  RNG1                  |LSI      |    0.032000 MHz |  yes   | Input frequency should be
as low as possible
  RNG2                  |LSI      |    0.032000 MHz |  yes   | Input frequency should be
as low as possible
  IWDG1                 |LSI      |    0.032000 MHz |  yes   |
  IWDG2                 |LSI      |    0.032000 MHz |  yes   |
LSE                     |N.A.     |    0.032768 MHz |  yes   | Mandatory for DTS
  RTC                   |LSE      |    0.032768 MHz |  yes   |
  TAMP                  |LSE      |    0.032768 MHz |  yes   |
  CEC                   |LSE      |    0.032768 MHz |  yes   |
  LPTIM4                |LSE      |    0.032768 MHz |  yes   |
  LPTIM5                |LSE      |    0.032768 MHz |  yes   |
HSI                     |N.A.     |   64.000000 MHz |  yes   |
  SPI4                  |HSI      |   64.000000 MHz |  yes   | Arduino
  SPI5                  |HSI      |   64.000000 MHz |  yes   | Rpi
  SPI6                  |HSI      |   64.000000 MHz |  no    |
  I2C4                  |HSI      |   64.000000 MHz |  yes   | PMIC
  I2C6                  |HSI      |   64.000000 MHz |  no    |
  I2C1                  |HSI      |   64.000000 MHz |  yes   | Rpi and peripherals
  I2C2                  |HSI      |   64.000000 MHz |  no    |
  I2C3                  |HSI      |   64.000000 MHz |  no    |
  I2C5                  |HSI      |   64.000000 MHz |  yes   | Rpi and Arduino
  USART1                |HSI      |   64.000000 MHz |  no    |
  USART2                |HSI      |   64.000000 MHz |  yes   | Bluetooth
  USART3                |HSI      |   64.000000 MHz |  yes   | Rpi
  UART4                 |HSI      |   64.000000 MHz |  yes   | Linux console
  UART5                 |HSI      |   64.000000 MHz |  no    |
```

```
  USART6                |HSI       |    64.000000 MHz |  no    |
  UART7                 |HSI       |    64.000000 MHz |  yes   | Arduino
  UART8                 |HSI       |    64.000000 MHz |  no    |
  MCO1                  |HSI       |    64.000000 MHz |  no    | Available so can be used
HSE                     |N.A.      |    24.000000 MHz |  yes   |
  DSIPLL                |HSE       |   125.000000 MHz |  no    | DSI DPHY PLL
    DSIBL               |DSIPLL    |   125.000000 MHz |  no    | DSI lanebyte clock
  RTCDIV                |HSE       |     1.000000 MHz |  yes   | Only used when RTC source
is HSE
  ck_per                |HSE       |    24.000000 MHz |  yes   |
    ADC                 |ck_per    |    24.000000 MHz |  yes   | Can use internal divider to
be < 40MHz
  PLL1                  |HSE       |          xxx MHz |  yes   |
    PLL1P               |PLL1      |          xxx MHz |  yes   |
      MPUDIV            |PLL1P     |          xxx MHz |  yes   |
      Cortex-A7         |PLL1P     |          xxx MHz |  yes   | 650 MHz or 800 MHz, see the
information box above
        MCO2            |Cortex-A7|          xxx MHz |  no    | Available so can be used
  PLL2                  |HSE       | 1066.000000 MHz |  yes   |
    PLL2P               |PLL2      |  266.500000 MHz |  yes   |
      AXI               |PLL2P     |  266.500000 MHz |  yes   | < 266MHz
        FMC             |AXI       |  266.500000 MHz |  no    |
        QSPI            |AXI       |  266.500000 MHz |  no    |
        SYSRAM          |AXI       |  266.500000 MHz |  yes   |
        ROM             |AXI       |  266.500000 MHz |  yes   |
        AHB5            |AXI       |  266.500000 MHz |  yes   | < 266MHz
          CRYP1         |AHB5      |  266.500000 MHz |  yes   |
          HASH1         |AHB5      |  266.500000 MHz |  yes   |
          GPIOZ         |AHB5      |  266.500000 MHz |  yes   |
          BKPSRAM       |AHB5      |  266.500000 MHz |  yes   |
        AHB6            |AXI       |  266.500000 MHz |  yes   | < 266MHz
          CRC1          |AHB6      |  266.500000 MHz |  yes   |
          MDMA          |AHB6      |  266.500000 MHz |  yes   |
          USBH          |AHB6      |  266.500000 MHz |  yes   | USB Host
          APB4          |AHB6      |  133.250000 MHz |  yes   |
          APB5          |AHB6      |   66.625000 MHz |  yes   |
            BSEC        |APB5      |   66.625000 MHz |  yes   | < 67MHz
            ETZPC       |APB5      |   66.625000 MHz |  yes   |
            TZC         |APB5      |   66.625000 MHz |  yes   |
        DBGAPB          |AXI       |  133.250000 MHz |  yes   | JTAG & Coresight
          DBGMCU        |DBGAPB    |   66.625000 MHz |  yes   |
          STM           |DBGAPB    |   66.625000 MHz |  no    |
    PLL2Q               |PLL2      |  533.000000 MHz |  yes   |
      GPU               |PLL2Q     |  533.000000 MHz |  yes   | < 533MHz
    PLL2R               |PLL2      |  533.000000 MHz |  yes   |
      DDR               |PLL2R     |  533.000000 MHz |  yes   | < 533MHz
  PLL3                  |HSE       |  417.755859 MHz |  yes   |
    PLL3P               |PLL3      |  208.877930 MHz |  yes   |
      MLAHB             |PLL3P     |  208.877930 MHz |  yes   | < 209MHz
        Cortex-M4       |MLAHB     |  208.877930 MHz |  yes   |
        SRAM1           |MLAHB     |  208.877930 MHz |  yes   |
        SRAM2           |MLAHB     |  208.877930 MHz |  yes   |
        SRAM3           |MLAHB     |  208.877930 MHz |  yes   |
        RETRAM          |MLAHB     |  208.877930 MHz |  yes   |
        AHB1            |MLAHB     |  208.877930 MHz |  yes   | < 209MHz
        AHB2            |MLAHB     |  208.877930 MHz |  yes   | < 209MHz
          DMA1          |AHB2      |  208.877930 MHz |  yes   |
          DMA2          |AHB2      |  208.877930 MHz |  yes   |
          DMAMUX        |AHB2      |  208.877930 MHz |  yes   |
        APB1            |MLAHB     |  104.438965 MHz |  yes   |
          LPTIM1        |APB1      |  104.438965 MHz |  yes   |
          WWDG          |APB1      |  104.438965 MHz |  yes   |
        APB2            |MLAHB     |  104.438965 MHz |  yes   |
        TIM2            |MLAHB     |  208.877930 MHz |  yes   | TIM Group 1
        TIM3            |MLAHB     |  208.877930 MHz |  yes   | TIM Group 1
        TIM4            |MLAHB     |  208.877930 MHz |  yes   | TIM Group 1
        TIM5            |MLAHB     |  208.877930 MHz |  yes   | TIM Group 1
```

```
        TIM6            |MLAHB    |  208.877930 MHz |  yes   | TIM Group 1
        TIM7            |MLAHB    |  208.877930 MHz |  yes   | TIM Group 1
        TIM12           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 1
        TIM13           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 1
        TIM14           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 1
        TIM1            |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM8            |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM15           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM16           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        TIM17           |MLAHB    |  208.877930 MHz |  yes   | TIM Group 2
        APB3            |MLAHB    |  104.438965 MHz |  yes   |
          LPTIM2        |APB3     |  104.438965 MHz |  yes   |
          LPTIM3        |APB3     |  104.438965 MHz |  yes   |
          SYSCFG        |APB3     |  104.438965 MHz |  yes   |
          VREFBUF       |APB3     |  104.438965 MHz |  yes   |
          DTS           |APB3     |  104.438965 MHz |  yes   |
          HDP           |APB3     |  104.438965 MHz |  no    |
        AHB3            |MLAHB    |  208.877930 MHz |  yes   | < 209MHz
          CRC2          |AHB3     |  208.877930 MHz |  yes   |
          CRYP2         |AHB3     |  208.877930 MHz |  yes   |
          HASH2         |AHB3     |  208.877930 MHz |  yes   |
          DCMI          |AHB3     |  208.877930 MHz |  no    |
          IPCC          |AHB3     |  208.877930 MHz |  yes   | Mailbox
        AHB4            |MLAHB    |  208.877930 MHz |  yes   | < 209MHz
          PWR           |AHB4     |  208.877930 MHz |  yes   |
          RCC           |AHB4     |  208.877930 MHz |  yes   |
          GPIOA-K       |AHB4     |  208.877930 MHz |  yes   |
          EXTI          |AHB4     |  208.877930 MHz |  yes   |
    PLL3Q               |PLL3     |   24.573874 MHz |  yes   |
      SPI1              |PLL3Q    |   24.573874 MHz |  yes   | SPI2S1 for BT PCM
      SPI2              |PLL3Q    |   24.573874 MHz |  yes   | SPI2S2 for HDMI
      SPI3              |PLL3Q    |   24.573874 MHz |  no    |
      DFSDM             |PLL3Q    |   24.573874 MHz |  no    |
      SAI1              |PLL3Q    |   24.573874 MHz |  no    |
      SAI2              |PLL3Q    |   24.573874 MHz |  yes   | AudCodec 48kHz (use PLL3R
for 44.1kHz)
      SAI3              |PLL3Q    |   24.573874 MHz |  no    |
      SAI4              |PLL3Q    |   24.573874 MHz |  no    |
    PLL3R               |PLL3     |   11.290699 MHz |  yes   |
  PLL4                  |HSE      |  594.000000 MHz |  yes   |
    PLL4P               |PLL4     |   99.000000 MHz |  yes   |
      SDMMC1            |PLL4P    |   99.000000 MHz |  yes   | µSD card
      SDMMC2            |PLL4P    |   99.000000 MHz |  yes   | Wifi
      SDMMC3            |PLL4P    |   99.000000 MHz |  yes   | Rpi
      SPDIF             |PLL4P    |   99.000000 MHz |  no    |
    PLL4Q               |PLL4     |   74.250000 MHz |  yes   |
      LCD               |PLL4Q    |   74.250000 MHz |  yes   | LTDC & DSI display pixel
clock
    PLL4R               |PLL4     |   74.250000 MHz |  yes   |
      FDCAN             |PLL4R    |   74.250000 MHz |  no    |
  STGEN                 |HSE      |   24.000000 MHz |  yes   |
  USBPHYC               |HSE      |   24.000000 MHz |  yes   | USB PHY Ctrl for USB Host
and OTG
    USBPLL              |USBPHYC  |   48.000000 MHz |  yes   |
      USBO              |USBPLL   |   48.000000 MHz |  yes   | USB OTG
CSI                     |N.A.     |    4.000000 MHz |  yes   | Mandatory for IO
compensation
ETH                     |N.A.     |    0.000000 MHz |  no    | ETH clocked by RGMII PHY
------------------------|---------|-----------------|--------
|-----------------------------------|
```

3.2.2 Device tree

Here are the corresponding device tree **rcc_clk** sub node properties consumed by the first stage boot loader (FSBL) to configure the clock tree above:

```
        st,clksrc = <
                CLK_MPU_PLL1P
                CLK_AXI_PLL2P
                CLK_MCU_PLL3P
                CLK_PLL12_HSE
                CLK_PLL3_HSE
                CLK_PLL4_HSE
                CLK_RTC_LSE
                CLK_MCO1_DISABLED
                CLK_MCO2_DISABLED
        >;

        st,clkdiv = <
                1 /*MPU*/
                0 /*AXI*/
                0 /*MCU*/
                1 /*APB1*/
                1 /*APB2*/
                1 /*APB3*/
                1 /*APB4*/
                2 /*APB5*/
                23 /*RTC*/
                0 /*MCO1*/
                0 /*MCO2*/
        >;

        st,pkcs = <
                CLK_CKPER_HSE
                CLK_FMC_ACLK
                CLK_QSPI_ACLK
                CLK_ETH_DISABLED
                CLK_SDMMC12_PLL4P
                CLK_DSI_DSIPLL
                CLK_STGEN_HSE
                CLK_USBPHY_HSE
                CLK_SPI2S1_PLL3Q
                CLK_SPI2S23_PLL3Q
                CLK_SPI45_HSI
                CLK_SPI6_HSI
                CLK_I2C46_HSI
                CLK_SDMMC3_PLL4P
                CLK_USBO_USBPHY
                CLK_ADC_CKPER
                CLK_CEC_LSE
                CLK_I2C12_HSI
                CLK_I2C35_HSI
                CLK_UART1_HSI
                CLK_UART24_HSI
                CLK_UART35_HSI
                CLK_UART6_HSI
                CLK_UART78_HSI
                CLK_SPDIF_PLL4P
                CLK_FDCAN_PLL4R
                CLK_SAI1_PLL3Q
                CLK_SAI2_PLL3Q
                CLK_SAI3_PLL3Q
                CLK_SAI4_PLL3Q
                CLK_RNG1_LSI
                CLK_RNG2_LSI
                CLK_LPTIM1_PCLK1
                CLK_LPTIM23_PCLK3
                CLK_LPTIM45_LSE
        >;

        /* VCO = 1066.0 MHz => P = 266 (AXI), Q = 533 (GPU), R = 533 (DDR) */
```

```
        pll2: st,pll@1 {
                cfg = < 2 65 1 0 0 PQR(1,1,1) >;
                frac = < 0x1400 >;
                u-boot,dm-pre-reloc;
        };

        /* VCO = 417.8 MHz => P = 209, Q = 25, R = 11 */
        pll3: st,pll@2 {
                cfg = < 1 33 1 16 36 PQR(1,1,1) >;
                frac = < 0x1a04 >;
                u-boot,dm-pre-reloc;
        };

        /* VCO = 594.0 MHz => P = 99, Q = 74, R = 74 */
        pll4: st,pll@3 {
                cfg = < 3 98 5 7 7 PQR(1,1,1) >;
                u-boot,dm-pre-reloc;
        };
```

Reset and Clock Control

Low Speed Internal oscillator (STM32 clock source)

Low Speed External oscillator (STM32 clock source)

High Speed Internal oscillator (STM32 clock source) or High Speed Synchronous Serial Interface (MIPI® Alliance standard)

High Speed External oscillator (STM32 clock source)

Multi Speed Internal oscillator (STM32 clock source)

Cortex®

First Stage Boot Loader

Linux® is a registered trademark of Linus Torvalds.

Das U-Boot -- the Universal Boot Loader (see U-Boot_overview)

Secondary Program Loader, *Also known as* ***U-Boot SPL***

Read Only Memory

Dynamic voltage and frequency scaling

Central processing unit

Independent Watchdog

Digital-to-analog converter (Electronic circuit that converts a binary number into a continuously varying value.)

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

Real Time Clock

Tamper

Consumer Electronics Control (HDMI standard)

Power Management Integrated Circuit

Display Serial Interface (MIPI® Alliance standard)

Analog-to-digital converter. The process of converting a sampled analog signal to a digital code that represents the amplitude of the original signal sample.

USB Host (STM32 specific)

Boot and Security and OTP control

Extended TrustZone Protection Controller

TrustZone® address space Controller for DDR

Arm® and TrustZone® are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

debug and test protocol, named from the Joint Test Action Group that developed it

System Trace Module

Graphics Processing Units

Doubledata rate (memory domain)

System Configuration

voltage reference buffer (STM32 specific)

Hardware Debug Port

Digital Camera Memory Interface

Inter-Processor Communication Controller

External Interrupt

Digital Filter for Sigma-Delta Modulator

Transmit

former spelling for e•MMC ('e' in italic)

Receive

LCD TFT Display Controller (STM32 specific)

System Time Generator

USB On-The-Go (Capability/type of USB port, acting primarily as USB device, to also act as USB host. Also known as USB OTG.)

input/output

Ethernet

Microprocessor Unit

Microcontroller Unit (MCUs have internal flash memory and are intended to operate with a minimum amount of external support ICs. They commonly are a self-contained, system-on-chip (SoC) designs.)

Discovery kit

BlueTooth

High-Definition Multimedia Interface (HDMI standard)