



Ethernet device tree configuration



Contents



A quality version of this page, approved on 6 October 2020, was based off this revision.

Contents

| | |
|---|----|
| 1 Article purpose | 4 |
| 2 DT bindings documentation | 5 |
| 3 DT configuration | 6 |
| 3.1 DT configuration (STM32 level) | 6 |
| 3.2 Ethernet DT configuration (board level) | 6 |
| 3.3 DT configuration examples | 7 |
| 3.3.1 RMII with Crystal on PHY (Reference clock (standard RMII clock name) is provided by a Phy Crystal) | 7 |
| 3.3.2 RMII with 25MHz on ETH_CLK (no PHY Crystal), REF_CLK from PHY (Reference clock (standard RMII clock name) is provided by a RCC SoC internal clock) | 8 |
| 3.3.3 RMII with 50MHz on ETH_CLK (no PHY Crystal), internal REF_CLK from RCC (Reference clock (standard RMII clock name) is provided by a RCC SoC internal clock) | 9 |
| 3.3.4 RGMII with Crystal on PHY, CLK125 from PHY (Reference clock (standard RGMII clock name) is provided by a Phy Crystal) | 11 |
| 3.3.5 RGMII with 25MHz on ETH_CLK (no PHY Crystal), CLK125 from PHY (Reference clock (standard RGMII clock name) is provided by a RCC SoC internal clock) | 12 |
| 3.3.6 RGMII with Crystal on PHY, no 125Mhz from PHY | 13 |
| 4 How to configure Ethernet using CubeMX | 15 |
| 5 References | 16 |



1 Article purpose

This article explains how to configure the Ethernet when it is assigned to the Linux[®]OS. In that case, it is controlled by the Ethernet framework

The configuration is performed using the [device tree](#) mechanism that provides a hardware description of the Ethernet peripheral, used by the STM32 DWMAC driver



2 DT bindings documentation

The *Ethernet* is a multifunction device.

Each function is represented by a separate binding document:

- "Generic" Ethernet device tree bindings ^[1]
- specific STM32 ETH device tree bindings ^[2]



3 DT configuration

This hardware description is a combination of the **STM32 microprocessor** device tree files (*.dtsi* extension) and **board** device tree files (*.dts* extension). See the [Device tree](#) for an explanation of the device tree file split.

3.1 DT configuration (STM32 level)

Ethernet peripheral nodes are located in `stm32mp151.dtsi` ^[3] file with a disabled status and some required properties such as:

- Physical base address and size of the device register map
- STMMAC interrupts
- `stmmaceth` clock and Rx, Tx clocks

This is a set of properties that may not vary for a given STM32MP device, such as: register addresses, interrupts, clocks, ...

```

ethernet0: ethernet@5800a000 {
    compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
    reg = <0x5800a000 0x2000>;
    reg-names = "stmmaceth";
    interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_NONE>;
    interrupt-names = "macirq";
    clock-names = "stmmaceth",
                  "mac-clk-tx",
                  "mac-clk-rx",
                  "ethstp";
    clocks = <&rcc ETHMAC>,
            <&rcc ETHTX>,
            <&rcc ETHRX>,
            <&rcc ETHSTP>;
    st,syscon = <&syscfg 0x4>;
    snps,mixed-burst;
    snps,pbl = <2>;
    snps,axi-config = <&stmmac_axi_config_0>;
    snps,tso;
    power-domains = <&pd_core>;
    status = "disabled";
};

```

The required and optional properties are fully described in the [bindings](#) files.



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

3.2 Ethernet DT configuration (board level)

The device tree board file (*.dts*) contains all hardware configurations related to board design. The DT node ("**ethernet**") should be updated to:

- Enable the Ethernet block by setting **status = "okay"**.
- Configure the pins in use via `pinctrl`, through `pinctrl-0` (default pins), `pinctrl-1` (sleep pins) and `pinctrl-names`.
- Configure Ethernet interface used **phy-mode = "rgmii"**, (rmii, mii, gmii).
- Configure Ethernet max speed **max-speed = <1000>**..



```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rgmii_pins_a>;
    pinctrl-1 = <&ethernet0_rgmii_pins_sleep_a>;
    pinctrl-names = "default", "sleep";
    phy-mode = "rgmii";
    max-speed = <1000>;
    phy-handle = <&phy0>;

    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@1 {
            reg = <1>;
        };
    };
};

```

3.3 DT configuration examples

The example below shows how to configure and enable an Ethernet instance at board level:

```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rmii_pins_a>;
    /* enable ethernet0 */
    /* configure pinctrl modes for ethernet0 */
    /*
    pinctrl-1 = <&ethernet0_rmii_pins_sleep_a>;
    as sleep pinctrl configuration for ethernet0 */
    pinctrl-names = "default", "sleep";
    phy-mode = "rmii";
    /* configure ethernet phy mode for
    ethernet0 */
    max-speed = <100>;
    /* configure ethernet max speed for
    ethernet0 */
    phy-handle = <&phy0>;

    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@1 {
            reg = <1>;
            /* configure ethernet phy @ for ethernet0
        */
    };
};
};

```

How to configure Ethernet for :

3.3.1 RMII with Crystal on PHY (Reference clock (standard RMII clock name) is provided by a Phy Crystal)

```

ethernet0: ethernet@5800a000 {
    compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
    reg = <0x5800a000 0x2000>;
    reg-names = "stmmaceth";
    interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_LEVEL_HIGH>,
        <&intc GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>,
        <&exti 70 I>;
    interrupt-names = "macirq",
        "eth_wake_irq",

```



```

        "stm32_pwr_wakeup";
clock-names = "stmmaceth",
              "mac-clk-tx",
              "mac-clk-rx",
              "ethstp";
clocks = <&rcc ETHMAC>,
        <&rcc ETHTX>,
        <&rcc ETHRX>,
        <&rcc ETHSTP>;
st,syscon = <&syscfg 0x4>;
snps,mixed-burst;
snps,pbl = <2>;
snps,en-tx-lpi-clockgating;
snps,axi-config = <&stmmac_axi_config_0>;
snps,tso;
power-domains = <&pd_core>;
status = "disabled";
};

```

```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rmii_pins_a>;
    pinctrl-1 = <&ethernet0_rmii_pins_sleep_a>;
    pinctrl-names = "default", "sleep";
    phy-mode = "rmii";
    max-speed = <100>;
    phy-handle = <&phy0>;
    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@0 {
            reg = <0>;
        };
    };
};
};

```

3.3.2 RMII with 25MHz on ETH_CLK (no PHY Crystal), REF_CLK from PHY (Reference clock (standard RMII clock name) is provided by a RCC SoC internal clock)

```

ethernet0: ethernet@5800a000 {
    compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
    reg = <0x5800a000 0x2000>;
    reg-names = "stmmaceth";
    interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_LEVEL_HIGH>,
                        <&intc GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>,
                        <&exti 70 I>;
    interrupt-names = "macirq",
                    "eth_wake_irq",
                    "stm32_pwr_wakeup";
    clock-names = "stmmaceth",
                "eth-ck",
                "mac-clk-tx",
                "mac-clk-rx",
                "ethstp";
    clocks = <&rcc ETHMAC>,
            <&rcc ETHCK_K>,
            <&rcc ETHTX>,
            <&rcc ETHRX>,
            <&rcc ETHSTP>;
    st,syscon = <&syscfg 0x4>;
    snps,mixed-burst;
};

```




```

snps,pbl = <2>;
snps,en-tx-lpi-clockgating;
snps,axi-config = <&stmmac_axi_config_0>;
snps,tso;
power-domains = <&pd_core>;
status = "disabled";
};

```

```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rmii_pins_a>;
    pinctrl-1 = <&ethernet0_rmii_pins_sleep_a>;
    pinctrl-names = "default", "sleep";
    phy-mode = "rmii";
    max-speed = <100>;
    phy-handle = <&phy0>;
    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@0 {
            reg = <0>;
        };
    };
};

```

+ update stm32mp15-pinctrl.dtsi ^[4] to add ETHCK pin in ethernet0_rmii_pins_* node:

For example:

```
<STM32_PINMUX('G', 8, AF2)>, /* ETH_RMII_ETHCK */
```

+ Need also to update TFA devicetree to generate 25Mhz clock (from PLL4P or PLL3Q):

for example if PLL4P in ed1 board: update fdt/stm32mp15xx-edx.dtsi

```

st,pkcs = <
    CLK_CKPER_HSE
    CLK_FMC_ACLK
    CLK_QSPI_ACLK
    - CLK_ETH_DISABLED
    + CLK_ETH_PLL4P
    ...
    /* VCO = 600.0 MHz => P = 25, Q = 50, R = 50 */
    pll4: st,pll@3 {
        compatible = "st,stm32mp1-pll";
        reg = <3>;
        cfg = < 1 49 23 11 11 PQR(1,1,1) >;
    };
};

```

3.3.3 RMII with 50MHz on ETH_CLK (no PHY Crystal), internal REF_CLK from RCC (Reference clock (standard RMII clock name) is provided by a RCC SoC internal clock)

```

ethernet0: ethernet@5800a000 {
    compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
    reg = <0x5800a000 0x2000>;
    reg-names = "stmmaceth";
    interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_LEVEL_HIGH>,
        <&intc GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>,
};

```



```

        <&exti 70 1>;
interrupt-names = "macirq",
                  "eth_wake_irq",
                  "stm32_pwr_wakeup";
clock-names = "stmmaceth",
              "eth-ck",
              "mac-clk-tx",
              "mac-clk-rx",
              "ethstp";
clocks = <&rcc ETHMAC>,
        <&rcc ETHCK_K>,
        <&rcc ETHTX>,
        <&rcc ETHRX>,
        <&rcc ETHSTP>;
st,syscon = <&syscfg 0x4>;
snps,mixed-burst;
snps,pbl = <2>;
snps,en-tx-lpi-clockgating;
st,eth_ref_clk_sel;                               /* In case of U-Boot */
or
st,eth-ref-clk-sel;                               /* In case of Linux Kernel */
snps,axi-config = <&stmmac_axi_config_0>;
snps,tso;
power-domains = <&pd_core>;
status = "disabled";
};

```

```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rmii_pins_a>;
    pinctrl-1 = <&ethernet0_rmii_pins_sleep_a>;
    pinctrl-names = "default", "sleep";
    phy-mode = "rmii";
    max-speed = <100>;
    phy-handle = <&phy0>;
    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@0 {
            reg = <0>;
        };
    };
};
};

```

+ update stm32mp15-pinctrl.dtsi to add ETHCK pin in ethernet0_rmii_pins_* node:

For example:

```
<STM32_PINMUX('G', 8, AF2)>, /* ETH_RMII_ETHCK */
```

+ Need also to update TFA to generate 50Mhz clock (from PLL4P or PLL3Q):

for example if PLL4P in ed1 board: update fdts/stm32mp15xx-edx.dtsi

```

st,pkcs = <
    CLK_CKPER_HSE
    CLK_FMC_ACLK
    CLK_QSPI_ACLK
    - CLK_ETH_DISABLED
    + CLK_ETH_PLL4P
    ...

```



```

/* VCO = 508.0 MHz => P = 50, Q = 60, R = 60 */
pll4: st,pll@3 {
    compatible = "st,stm32mp1-pll";
    reg = <3>;
    cfg = < 1 49 11 9 9 PQR(1,1,1) >;
};

```

3.3.4 RGMII with Crystal on PHY, CLK125 from PHY (Reference clock (standard RGMII clock name) is provided by a Phy Crystal)

```

ethernet0: ethernet@5800a000 {
    compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
    reg = <0x5800a000 0x2000>;
    reg-names = "stmmaceth";
    interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_LEVEL_HIGH>,
        <&intc GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>,
        <&exti 70 1>;
    interrupt-names = "macirq",
        "eth_wake_irq",
        "stm32_pwr_wakeup";
    clock-names = "stmmaceth",
        "mac-clk-tx",
        "mac-clk-rx",
        "ethstp";
    clocks = <&rcc ETHMAC>,
        <&rcc ETHTX>,
        <&rcc ETHRX>,
        <&rcc ETHSTP>;
    st,syscon = <&syscfg 0x4>;
    snps,mixed-burst;
    snps,pbl = <2>;
    snps,en-tx-lpi-clockgating;
    snps,axi-config = <&stmmac_axi_config_0>;
    snps,tso;
    power-domains = <&pd_core>;
    status = "disabled";
};

```

```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rgmii_pins_a>;
    pinctrl-1 = <&ethernet0_rgmii_pins_sleep_a>;
    pinctrl-names = "default", "sleep";
    phy-mode = "rgmii";
    max-speed = <1000>;
    phy-handle = <&phy0>;
    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@0 {
            reg = <0>;
        };
    };
};

```



3.3.5 RGMII with 25MHz on ETH_CLK (no PHY Crystal), CLK125 from PHY (Reference clock (standard RGMII clock name) is provided by a RCC SoC internal clock)

```

ethernet0: ethernet@5800a000 {
    compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
    reg = <0x5800a000 0x2000>;
    reg-names = "stmmaceth";
    interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_LEVEL_HIGH>,
                        <&intc GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>,
                        <&exti 70 1>;
    interrupt-names = "macirq",
                    "eth_wake_irq",
                    "stm32_pwr_wakeup";
    clock-names = "stmmaceth",
                "eth-ck",
                "mac-clk-tx",
                "mac-clk-rx",
                "ethstp";
    clocks = <&rcc ETHMAC>,
            <&rcc ETHCK K>,
            <&rcc ETHTX>,
            <&rcc ETHRX>,
            <&rcc ETHSTP>;
    st,syscon = <&syscfg 0x4>;
    snps,mixed-burst;
    snps,pbl = <2>;
    snps,en-tx-lpi-clockgating;
    snps,axi-config = <&stmmac_axi_config_0>;
    snps,tso;
    power-domains = <&pd_core>;
    status = "disabled";
};

```

```

&ethernet0 {
    status = "okay";
    pinctrl-0 = <&ethernet0_rgmii_pins_a>;
    pinctrl-1 = <&ethernet0_rgmii_pins_sleep_a>;
    pinctrl-names = "default", "sleep";
    phy-mode = "rgmii";
    max-speed = <1000>;
    phy-handle = <&phy0>;
    mdio0 {
        #address-cells = <1>;
        #size-cells = <0>;
        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@0 {
            reg = <0>;
        };
    };
};

```

+ update stm32mp15-pinctrl.dtsi to add ETHCK pin in ethernet0_rgmii_pins_* node:

For example:

```
<STM32_PINMUX('G', 8, AF2)>, /* ETH_RGMII_ETHCK */
```

+ Need also to update TFA to generate 25Mhz clock (from PLL4P or PLL3Q):

for example if PLL4P in ed1 board: update fdts/stm32mp15xx-edx.dtsi



```

st,pkcs = <
  CLK_CKPER_HSE
  CLK_FMC_ACLK
  CLK_QSPI_ACLK
  - CLK_ETH_DISABLED
  + CLK_ETH_PLL4P
...
/* VCO = 600.0 MHz => P = 25, Q = 50, R = 50 */
pll4: st,pll@3 {
  compatible = "st,stm32mp1-pll";
  reg = <3>;
  cfg = < 1 49 23 11 11 PQR(1,1,1) >;
};

```

3.3.6 RGMII with Crystal on PHY, no 125Mhz from PHY

```

ethernet0: ethernet@5800a000 {
  compatible = "st,stm32mp1-dwmac", "snps,dwmac-4.20a";
  reg = <0x5800a000 0x2000>;
  reg-names = "stmmaceth";
  interrupts-extended = <&intc GIC_SPI 61 IRQ_TYPE_LEVEL_HIGH>,
    <&intc GIC_SPI 62 IRQ_TYPE_LEVEL_HIGH>,
    <&exti 70 1>;
  interrupt-names = "macirq",
    "eth_wake_irq",
    "stm32_pwr_wakeup";
  clock-names = "stmmaceth",
    "eth-ck",
    "mac-clk-tx",
    "mac-clk-rx",
    "ethstp";
  clocks = <&rcc ETHMAC>,
    <&rcc ETHCK_K>,
    <&rcc ETHTX>,
    <&rcc ETHRX>,
    <&rcc ETHSTP>;
  st,syscon = <&syscfg 0x4>;
  snps,mixed-burst;
  snps,pbl = <2>;
  snps,en-tx-lpi-clockgating;
  st,eth_clk_sel; /* In case of U-Boot */
  or
  st,eth-clk-sel; /* In case of Linux Kernel */
  snps,axi-config = <&stmmac_axi_config_0>;
  snps,tso;
  power-domains = <&pd_core>;
  status = "disabled";
};

```

```

&ethernet0 {
  status = "okay";
  pinctrl-0 = <&ethernet0_rgmii_pins_a>;
  pinctrl-1 = <&ethernet0_rgmii_pins_sleep_a>;
  pinctrl-names = "default", "sleep";
  phy-mode = "rgmii";
  max-speed = <1000>;
  phy-handle = <&phy0>;
  mdio0 {
    #address-cells = <1>;
    #size-cells = <0>;

```



```

        compatible = "snps,dwmac-mdio";
        phy0: ethernet-phy@0 {
            reg = <0>;
        };
    };
};

```

+ update stm32mp15-pinctrl.dtsi to delete CLK125 pin (also no need of ETHCK pin) in ethernet0_rgmii_pins_* node:

+ Need also to update TFA to generate 125Mhz clock (from PLL4P or PLL3Q):

for example if PLL4P in ed1 board: update fdts/stm32mp15xx-edx.dtsi

```

st,pkcs = <
    CLK_CKPER_HSE
    CLK_FMC_ACLK
    CLK_QSPI_ACLK
    - CLK_ETH_DISABLED
    + CLK_ETH_PLL4P
    ...
    /* VCO = 750.0 MHz => P = 125, Q = 62.5, R = 62.5 */
    pll4: st,pll@3 {
        compatible = "st,stm32mp1-pll";
        reg = <3>;
        cfg = < 3 124 5 11 11 PQR(1,1,1) >;
    };
};

```



4 How to configure Ethernet using CubeMX

The STM32CubeMX tool can be used to configure the STM32MPU device and get the corresponding platform configuration device tree files.

The STM32CubeMX may not support all the properties described in the above DT bindings documentation paragraph. If so, the tool inserts **user sections** in the generated device tree. These sections can then be edited to add some properties and they are preserved from one generation to another. Refer to STM32CubeMX user manual for further information.



5 References

- [Documentation/devicetree/bindings/net/stmmac.txt](#)
- [Documentation/devicetree/bindings/net/stm32-dwmac.txt](#)
- [arch/arm/boot/dts/stm32mp151.dtsi](#) , STM32MP151 device tree file
- [arch/arm/boot/dts/stm32mp15-pinctrl.dtsi](#) , STM32MP15 pinctrl device tree file

Linux[®] is a registered trademark of Linus Torvalds.

Operating System

Device Tree

Ethernet

Generic Interrupt Controller

Serial Peripheral Interface

Reset and Clock Control

High Speed External oscillator (STM32 clock source)

Das U-Boot -- the Universal Boot Loader (see [U-Boot_overview](#))