

Lierda IC530&IC610 ST cubemx 应用指导

产品名称: ST-A7-IC530 工业核心板

产品型号: L-IDMICS0-AA545

版本: Rev1.0

日期: 25/03/15

状态: 受控版本

法律声明

若接收利尔达科技集团股份有限公司(以下称为“利尔达”)的此份文档,即表示您已经同意以下条款。若不同意以下条款,请停止使用本文档。

本文档版权归利尔达科技集团股份有限公司所有,保留任何未在本文档中明示授予的权利。文档中涉及利尔达的专有信息。未经利尔达事先书面许可,任何单位和个人不得复制、传递、分发、使用和泄漏该文档以及该文档包含的任何图片、表格、数据及其他信息。

本产品符合有关环境保护和人身安全方面的设计要求,产品的存放、使用和弃置应遵照产品手册、相关合同或者相关法律、法规的要求进行。

本公司保留在不预先通知的情况下,对此手册中描述的产品进行修改和改进的权利;同时保留随时修订或收回本手册的权利。



文件修订历史

文档版本	变更日期	修订人	审核人	变更内容
Rev1.0	25-03-15	YQA		初始版本
Rev1.1	25-7-11	YQA		添加 ic610 cubemx6.14

Lierda
利 尔 达

目录

法律声明	1
文件修订历史	2
目录	3
1 引言	4
2 cubemx 版本选择及安装	4
2.1 版本选择	4
2.2 Cubemx 安装	4
3 Cubemx 使用	12
3.1 打开 cubemx 源码	12
3.2 Cubemx 下 gpio 复用说明	15
3.3 Cubemx 使能 usart2 及自动生成代码	17
3.4 dts 说明	22
3.5 dts 解析	23
3.5.1 includes	23
3.5.2 memory	23
3.5.3 USER CODE BEGIN root	24
3.5.4 pinctrl	26
3.5.5 外设 usercode	27
3.6 Debug 串口修改	28
3.7 串口复用为 485	29

1 引言

本文介绍了 st32mpu 下 cubemx 的使用，因 ic610 当前 cubemx 不支持对全部源码的开发，故实例介绍中，采用的是 stm32mp1 系列的 ic530 cubemx 源码。

2 cubemx 版本选择及安装

2.1 版本选择

Cubemx 官方下载链接：<https://www.st.com/en/development-tools/stm32cubemx.html>

Cubemx 当前最新版本为 6.12.1，不同版本创建的 project 存在不兼容，故版本选择时必须与本开发板选择的版本一致。

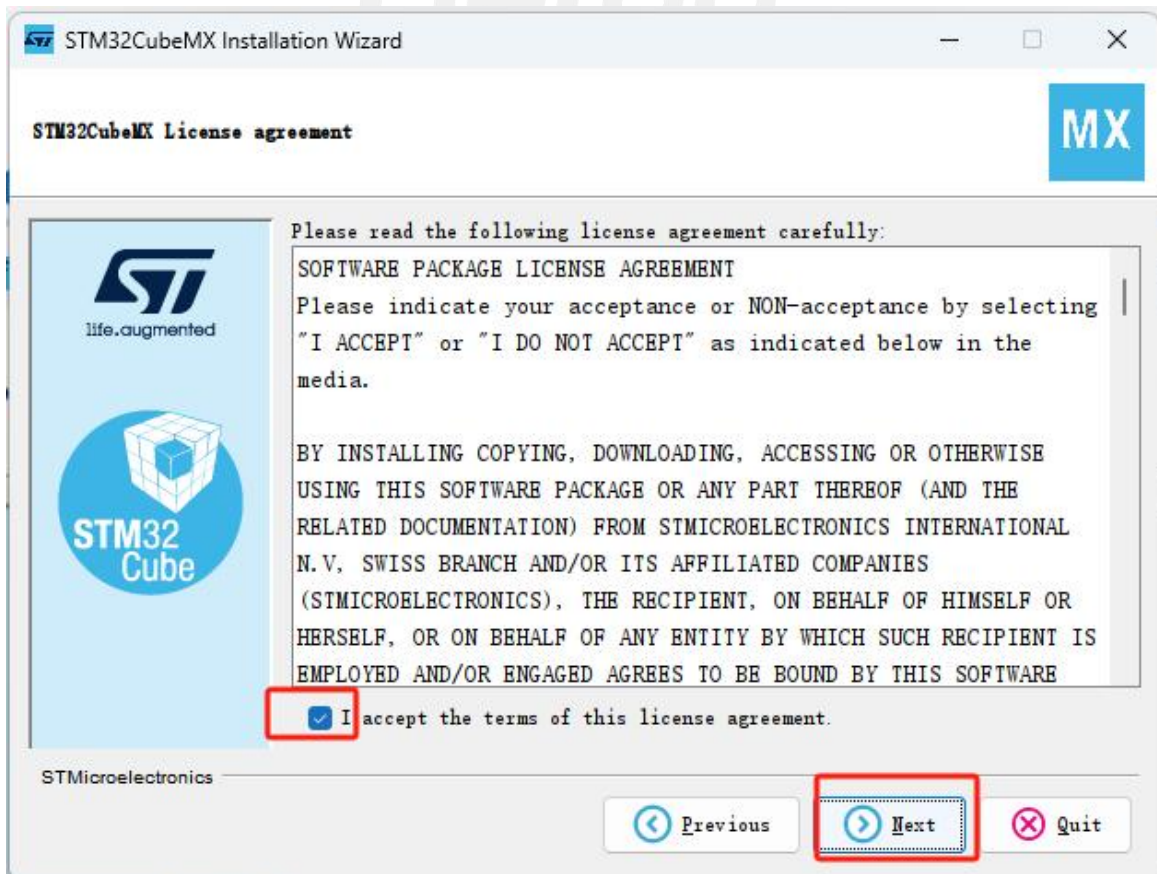
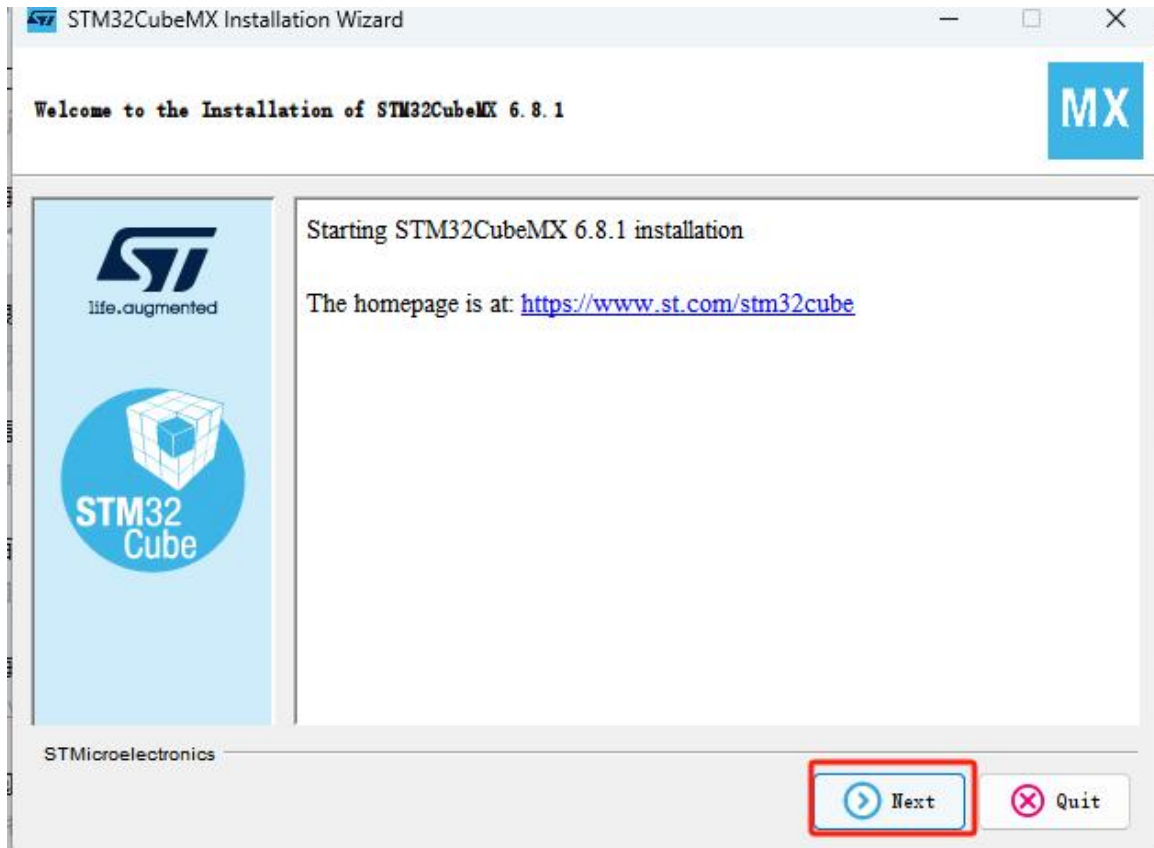
IC530: STM32MP1 系列，project 及源码构建使用的 cubemx 版本为 6.8.1。

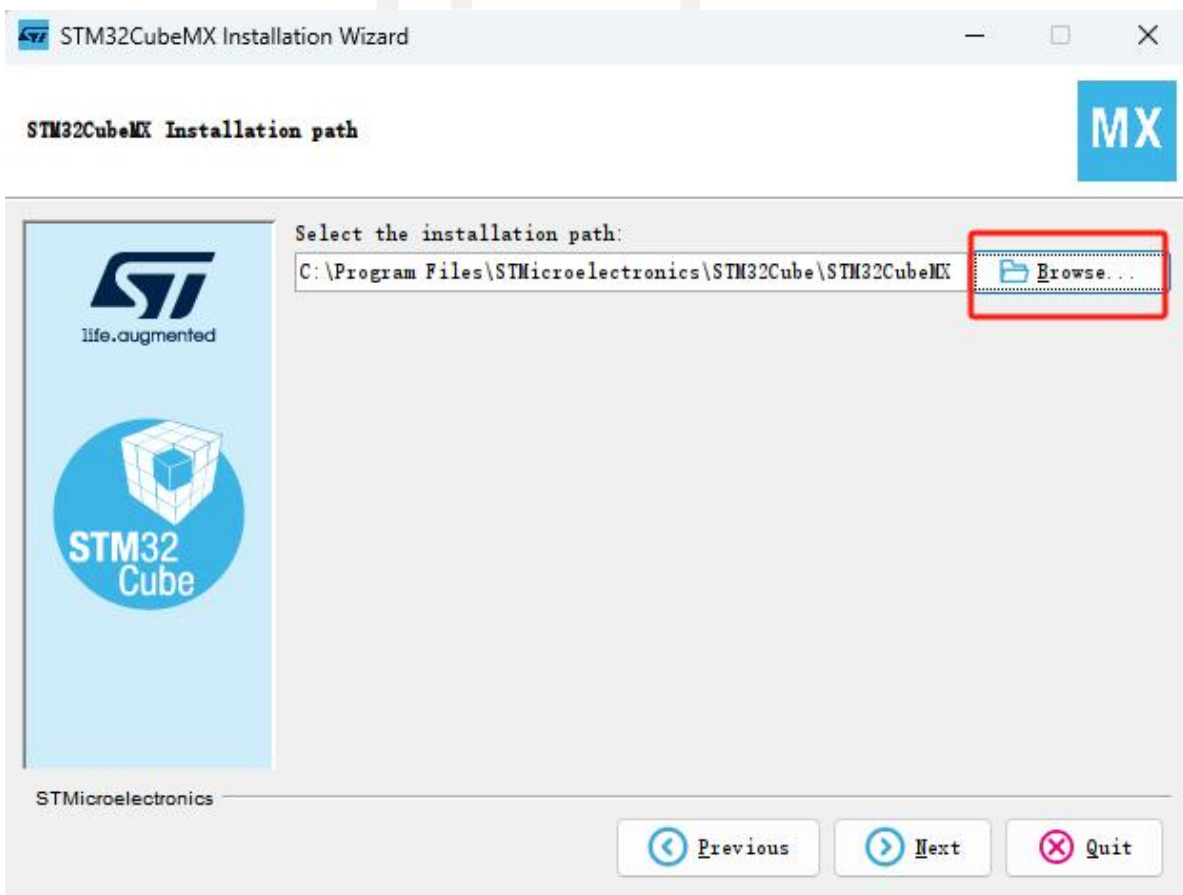
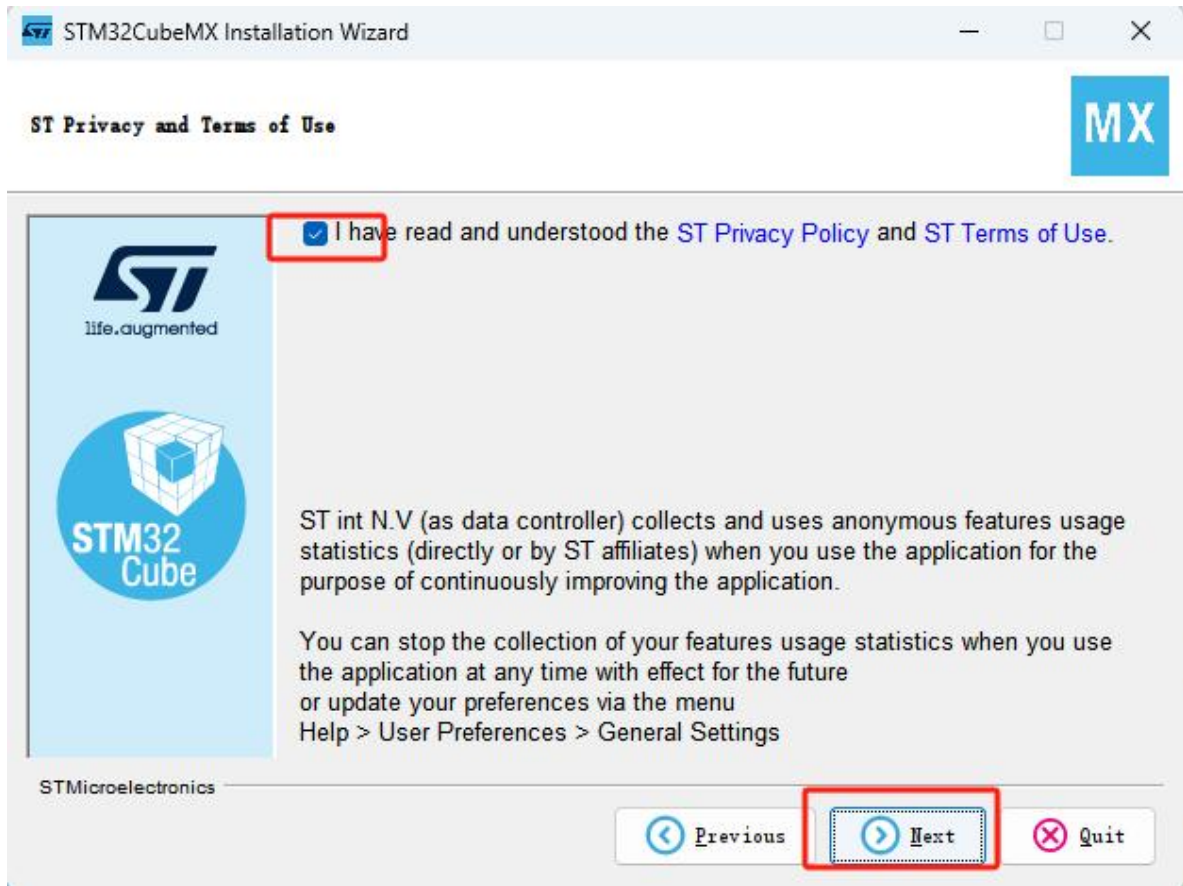
IC610: 使用 cubemx 构建工程及源码，cubemx 版本 6.14.0。

2.2 Cubemx 安装

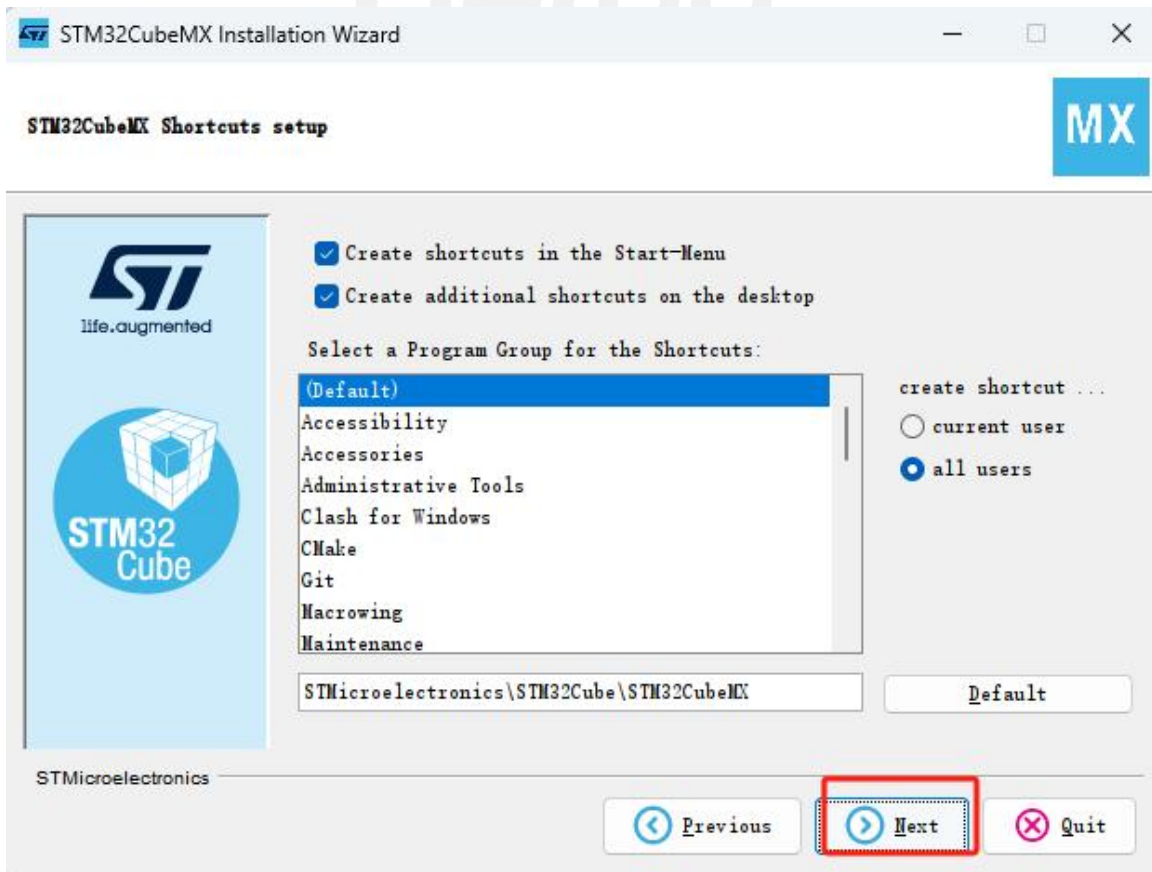
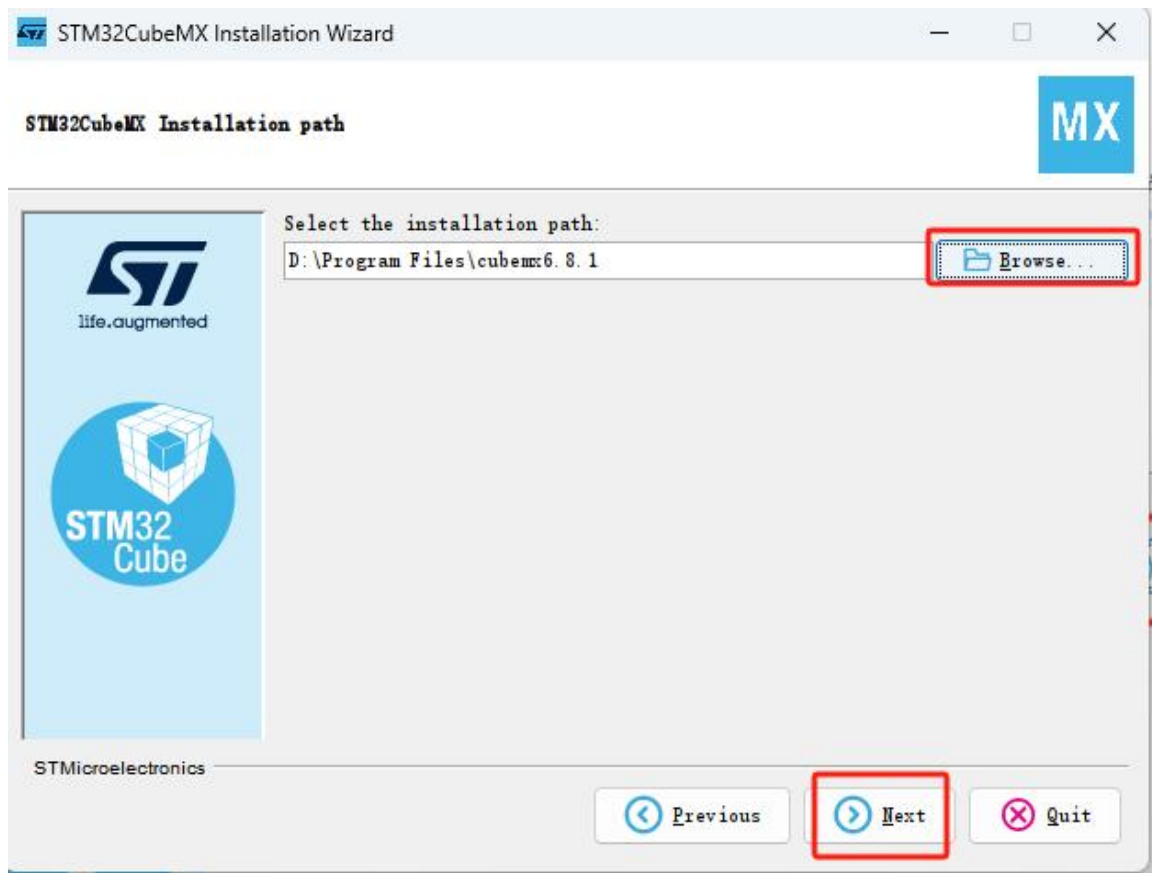
以 ic530 为例，使用 6.8.1，Cubemx 官方下载链接下载该版本或者下载 ic530 sdk 直接解压 tools 下 en.stm32cubemx-win-v6-8-1.zip 并安装。Windows 下 cubemx 可以安装到不同路径下多个版本，如当前版本安装路径为 D:\Program Files\cubemx6.8.1，在 D:\Program Files\下创建目录 cubemx6.8.1。

双击 SetupSTM32CubeMX-6.8.1-Win.exe 进行安装。

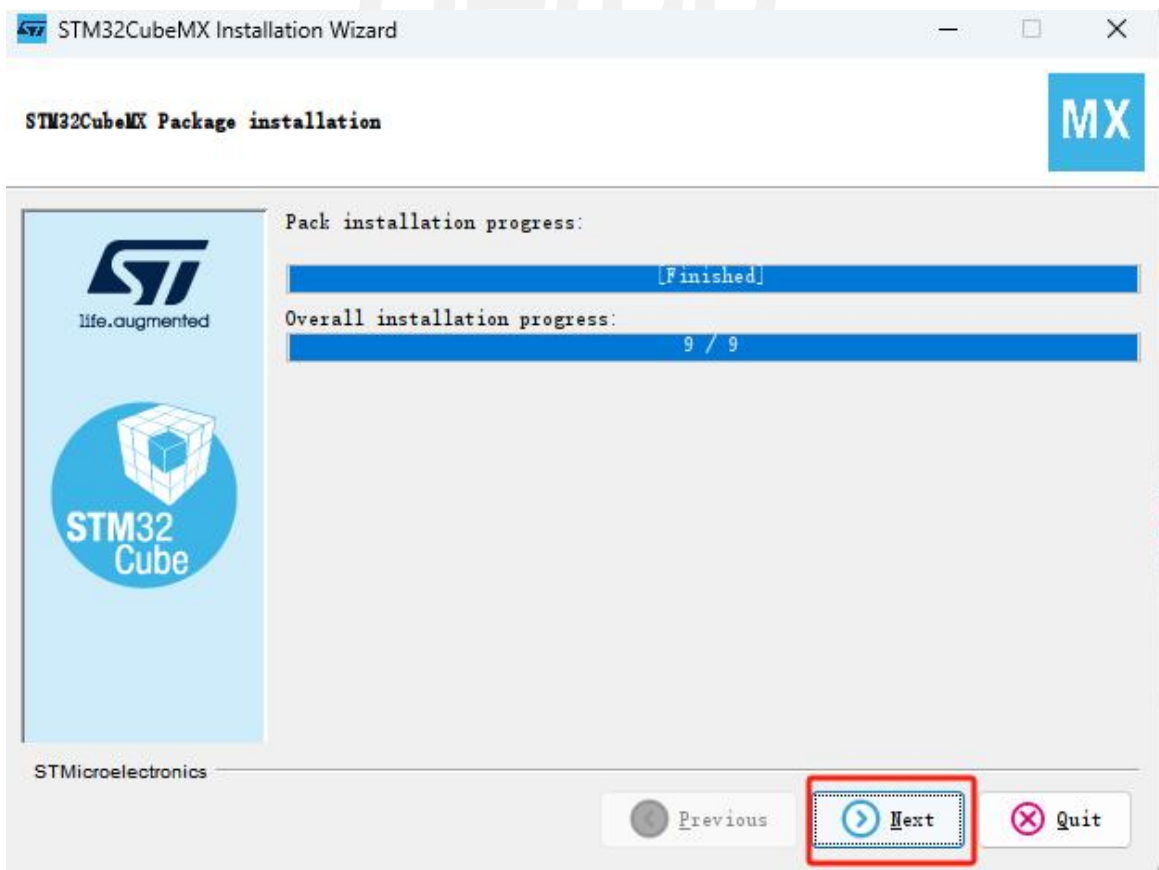
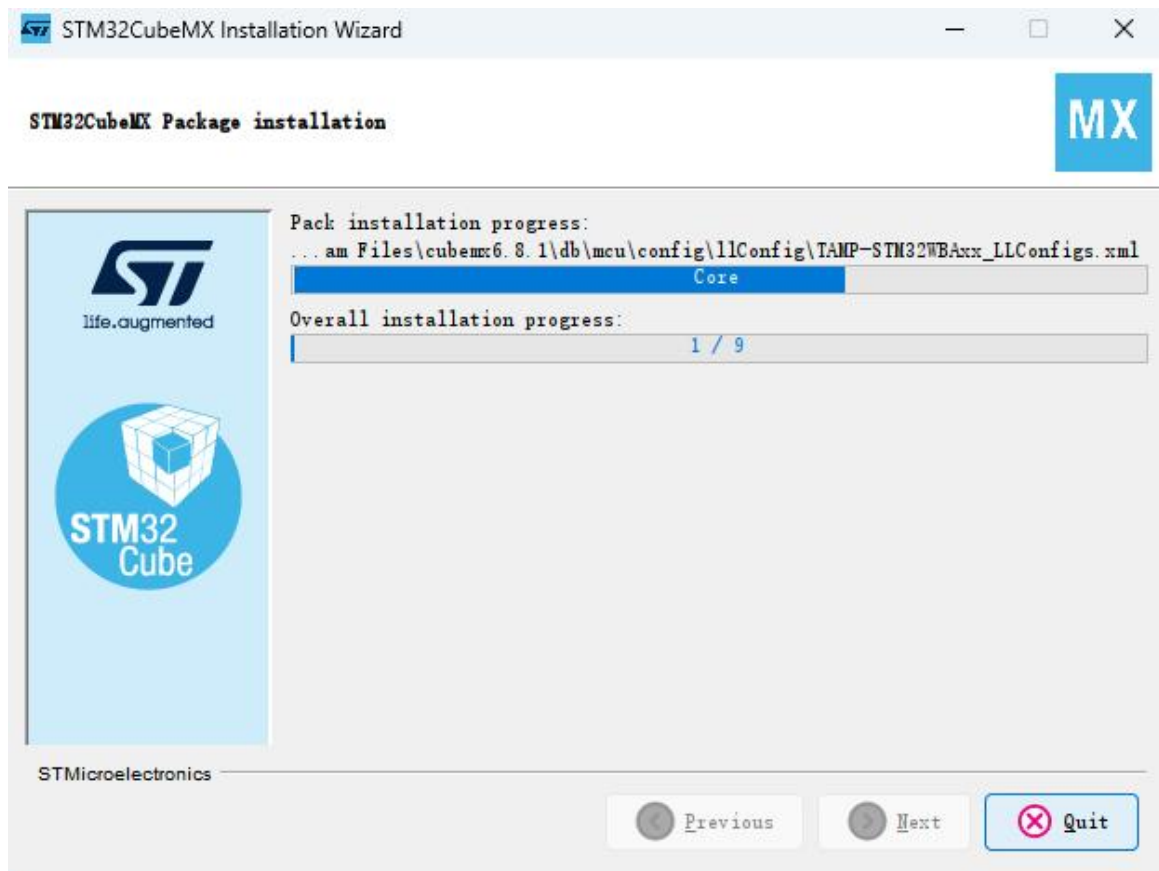




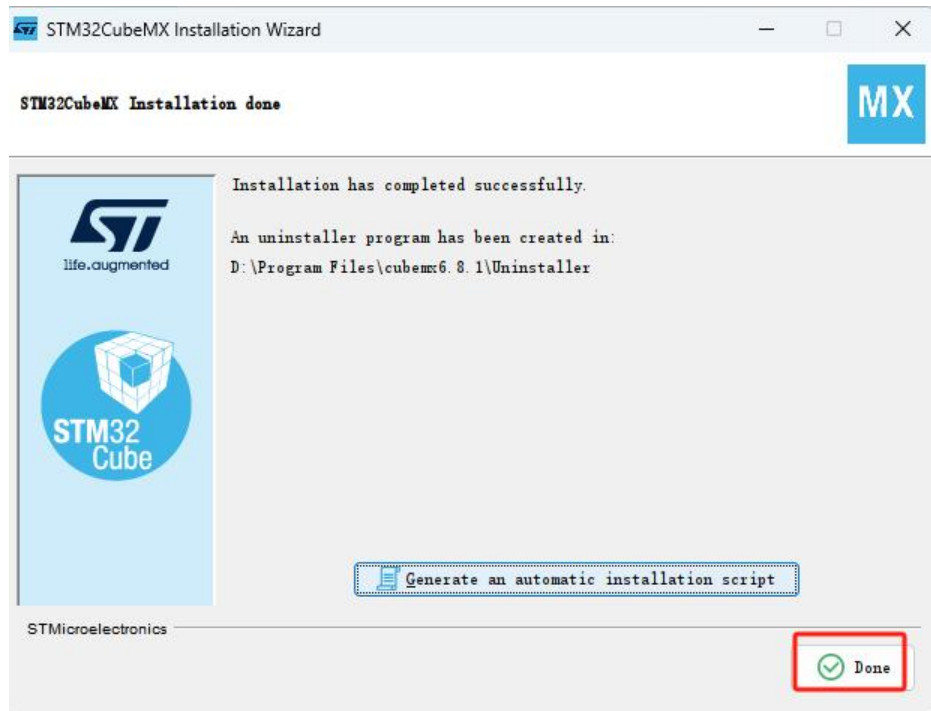
选择安装路径 D:\Program Files\cubemx6.8.1。



安装中:



如下表示安装完成：



完整完成后桌面自动生成 cubemx 快捷方式。

右击属性，查看信息如下。



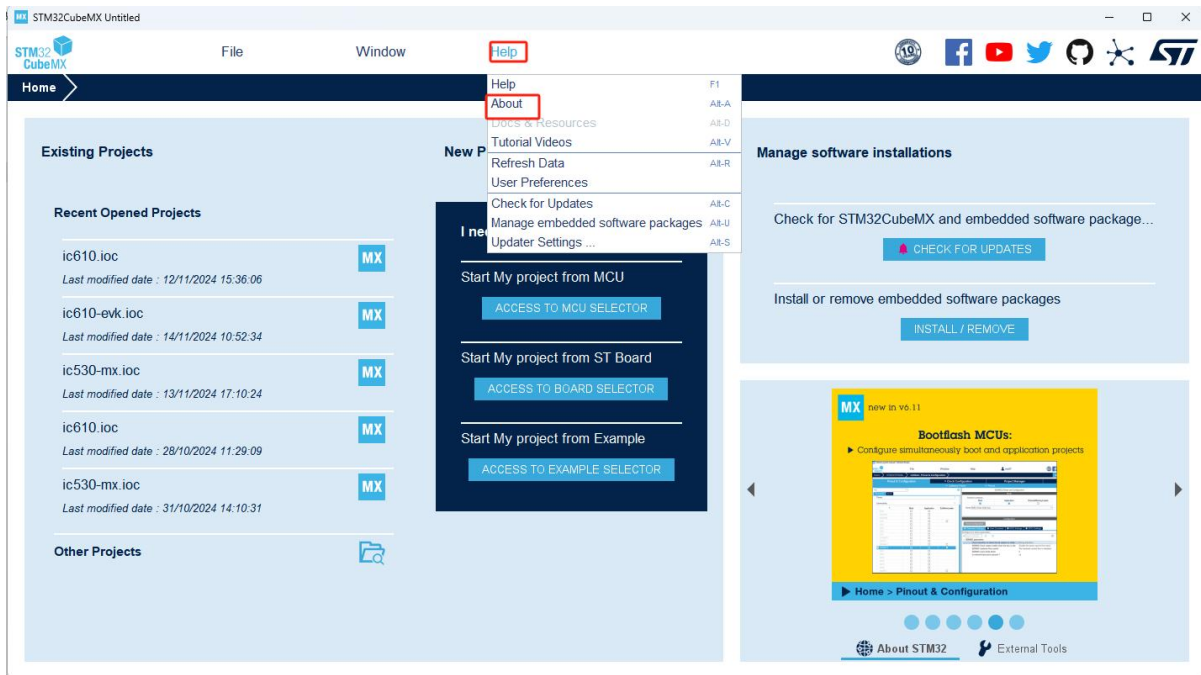
该快捷方式为安装 cubemx 自动生成，若再次安装其他版本，则该快捷方式会被替换，进入 D:\Program Files\cubemx6.8.1\右击 STM32CubeMX.exe 创建快捷方式。



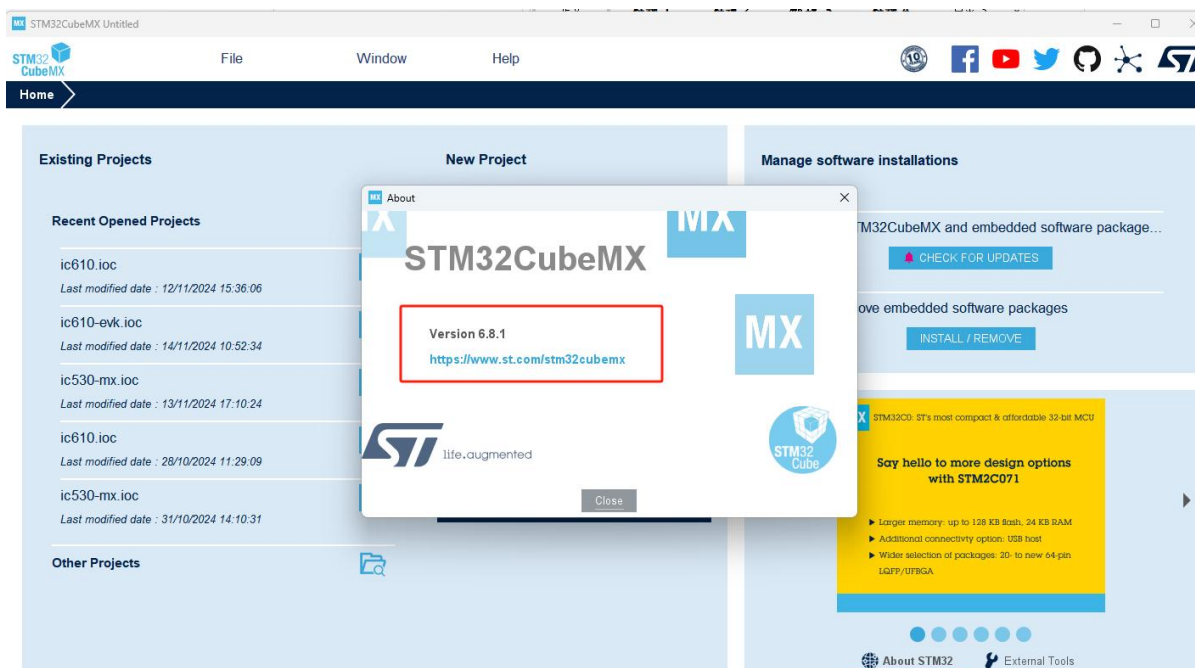
生成快捷方式 “STM32CubeMX.exe - 快捷方式” 重命名为 “STM32CubeMX-6.8.1.exe ” 将 STM32CubeMX-6.8.1.exe 剪切至桌面，则 cubemx6.8.1 安装完成。



双击 STM32CubeMX-6.8.1.exe 打开后，点击 “help” “About” 查看版本信息。



如下，版本安装正确。

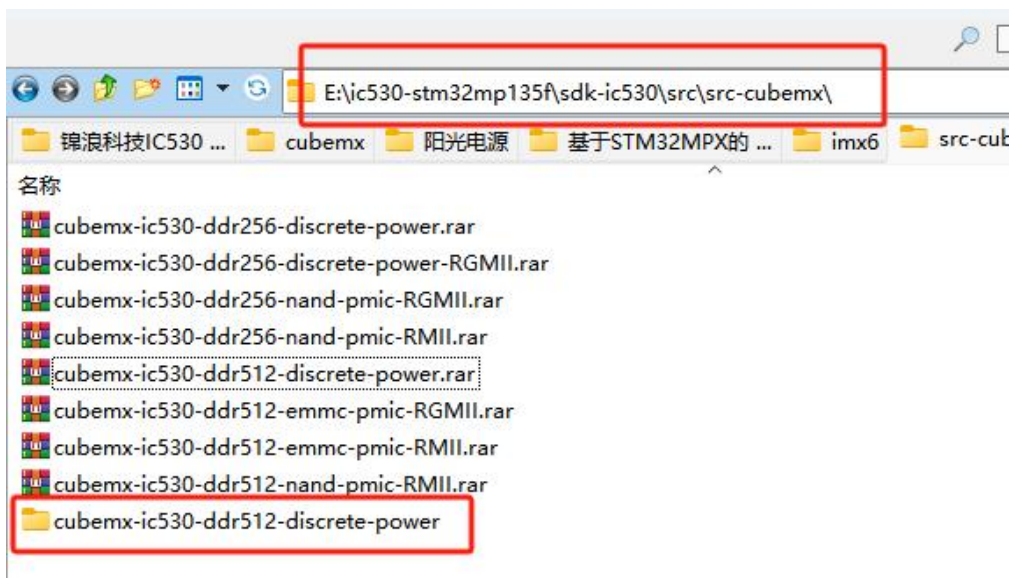


3 Cubemx 使用

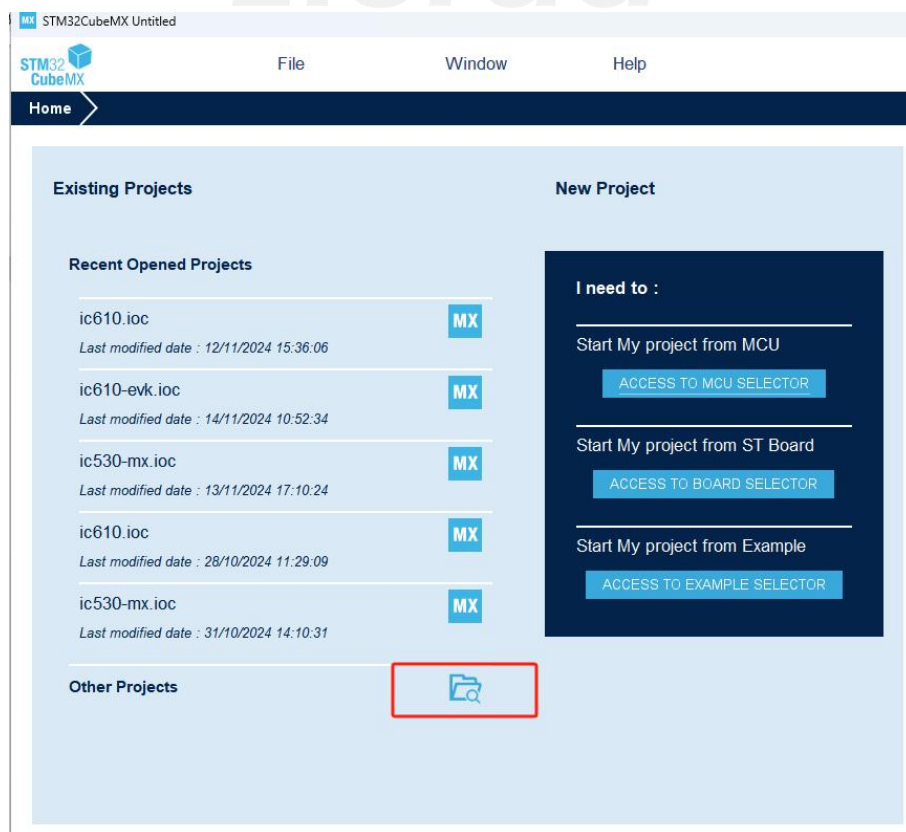
3.1 打开 cubemx 源码

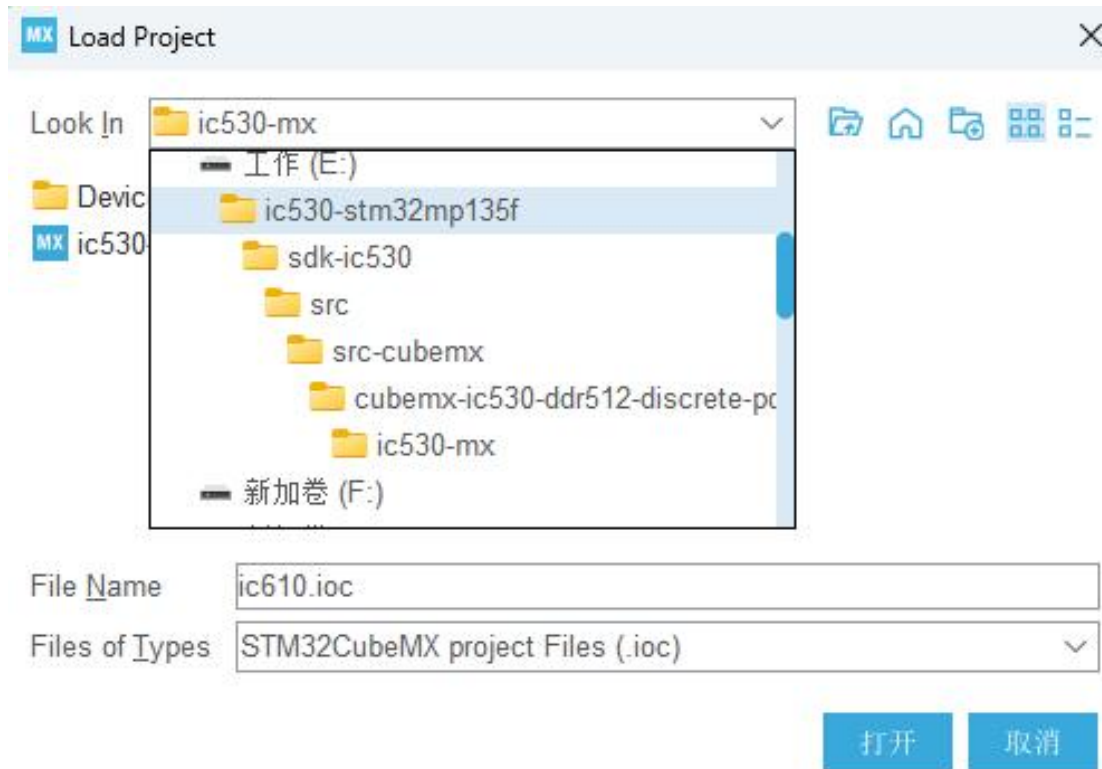
选择 sdk-ic530\src\src-cubemx\下源码如

cubemx-ic530-ddr512-discrete-power.rar，选择解压到当前文件夹。

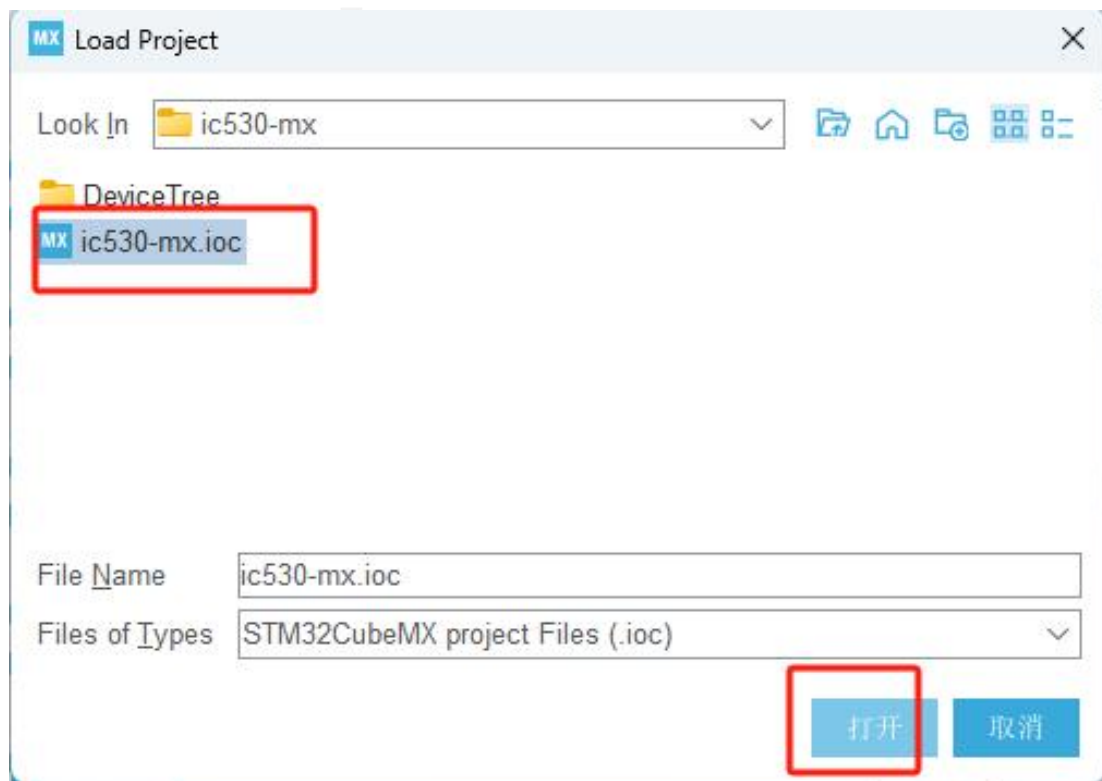


打开桌面快捷方式 cubemx-6.8.1，点击文件打开标记，打开上述源码路径。

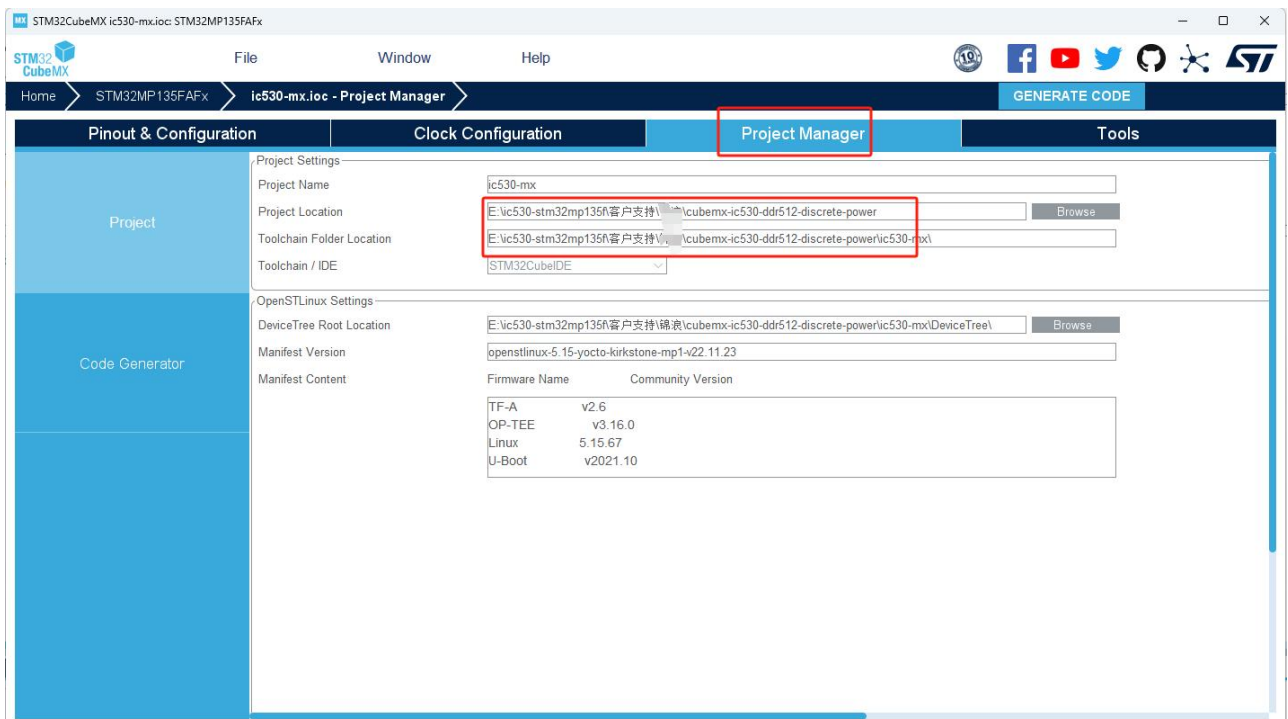
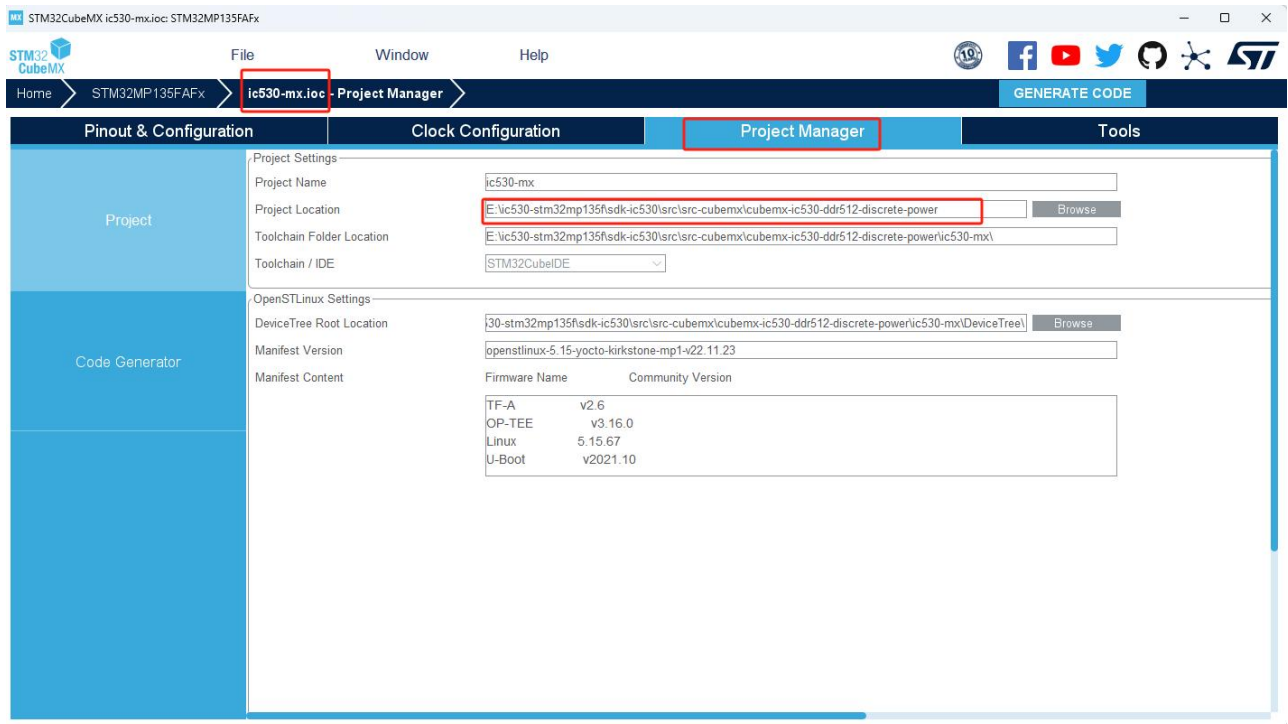




选择 ic530-mx.ioc 文件后点击打开。



打开多个 cubemx6.8.1,Cubemx 可以同时打开多个 project, Project Manager 可查看当前 project 的路径以区分。



3.2 Cubemx6.14 STM32Cube_FW_MP2 设置

IC610 版本为 6.14，该版本需要配置 Firmware，配置步骤如下：

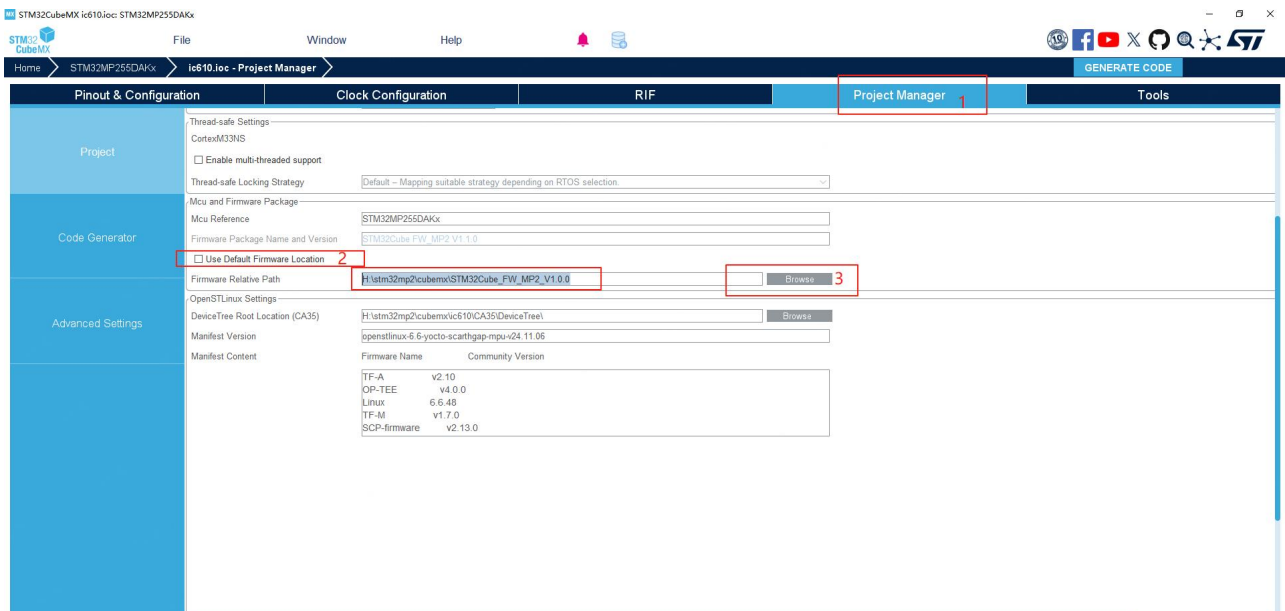
IC610 cubemx 源码：ic610-sdk\src\cubemx-dts\ic610.rar

Firmware : ic610-sdk\tools\STM32Cube_FW_MP2_V1.0.0.7z

将上述源码和 firmware 解压后，cubemx6.14 打开 ic610.ioc。

如 firmware 解压后路径为：H:\stm32mp2\cubemx\STM32Cube_FW_MP2_V1.0.0

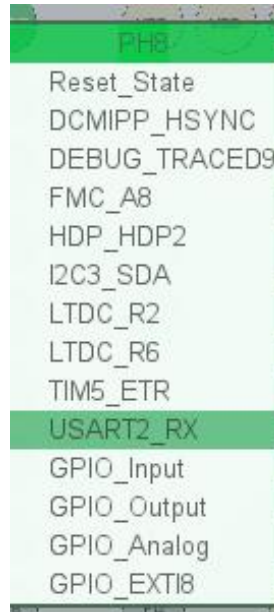
设置如下：



3.3 Cubemx 下 gpio 复用说明

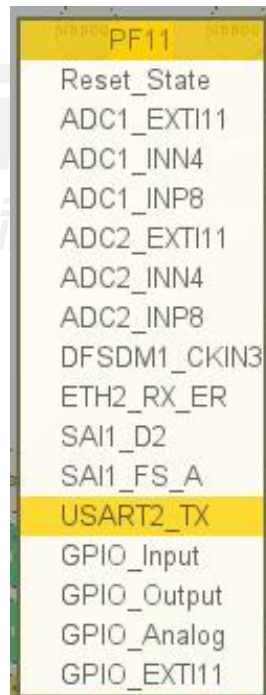
绿色：

cubemx 自动复用为某个功能或该 gpio 已经手动复用为某个功能该外设功能已经打开，该外设功能关闭，io 自动释放为灰色；



黄色：

该 gpio 手动复用为某个功能了，但该外设功能已经关闭，需要手动释放，右击黄色引脚选择“Reset_State”模式即可，否则 cubemx 无法自动使用该 gpio。



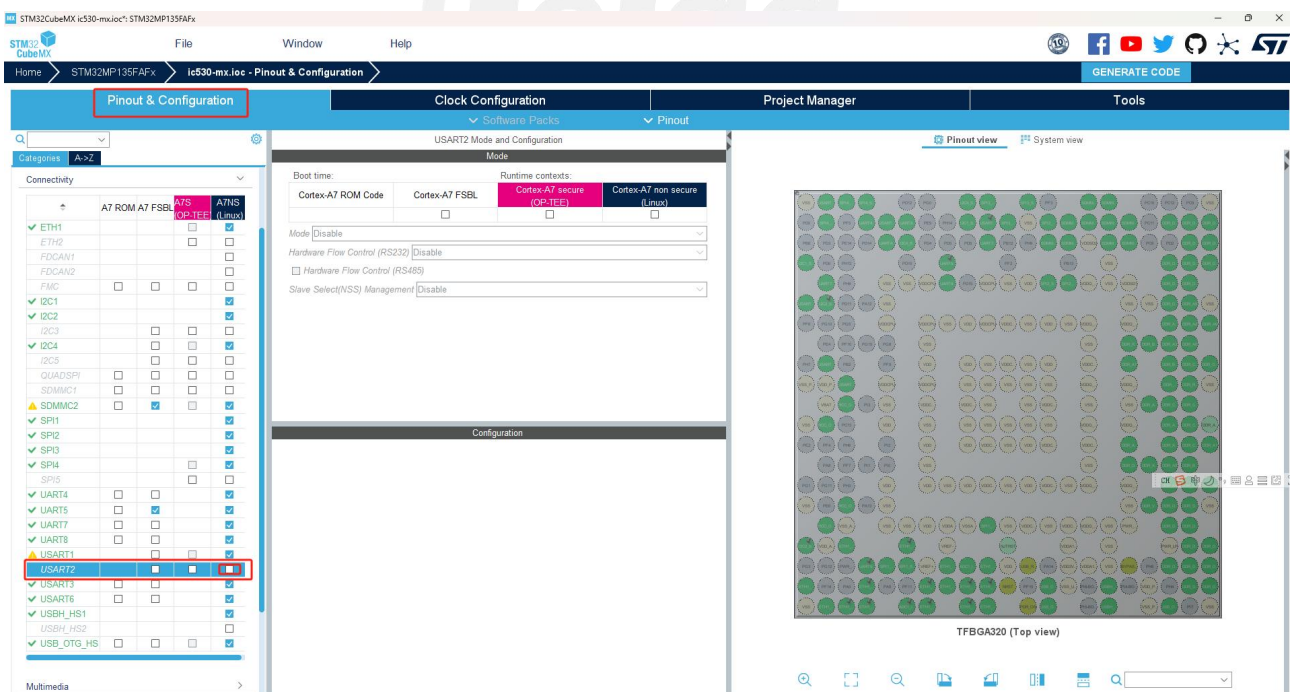
灰色：

cubemx 未使用该 gpio，闲置，配置功能时 cubemx 会根据当前闲置的 gpio，选择复用功能配置。

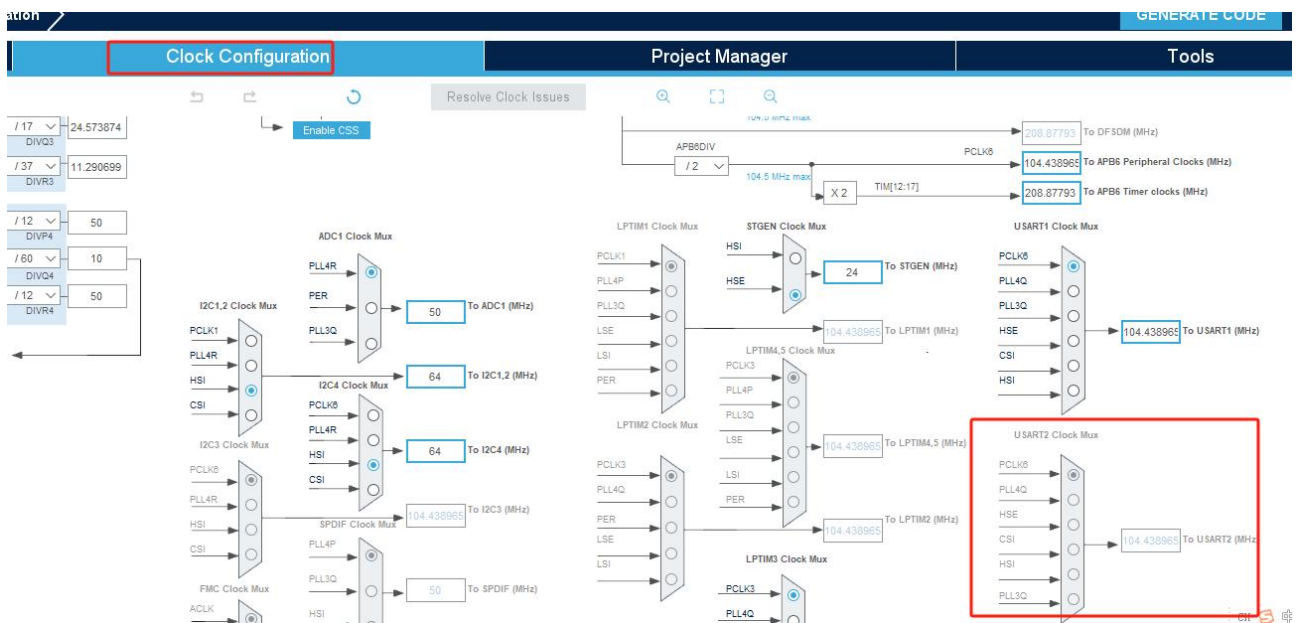


3.4 Cubemx 使能 usart2 及自动生成代码

Pinout&Configuration 为外设设置菜单, Clock Configuration 为外设时钟配置菜单。Clock 部分基础时钟已经配置好, 配置外设 Pinout&Configuration 后 clock 部分 cubemx 会自动配置, 用户也可以手动修改。

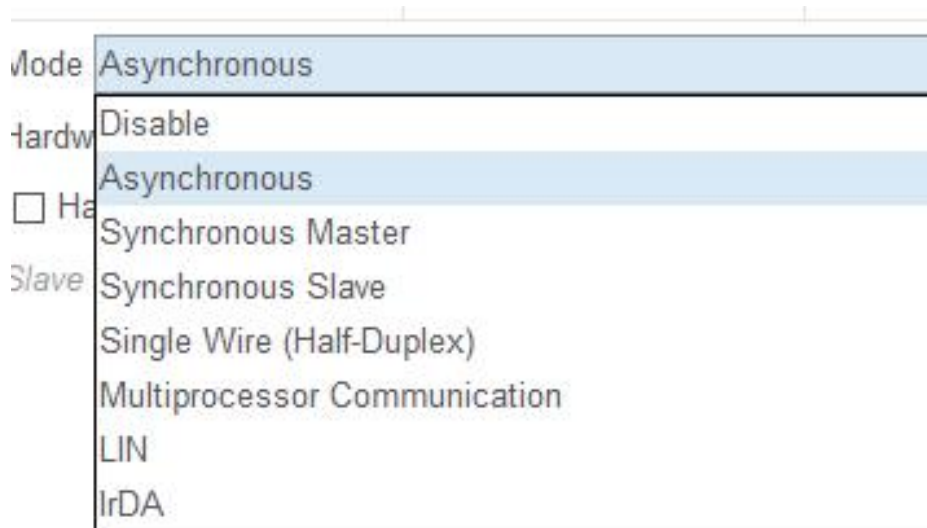


Usart2 未使能前 clock:

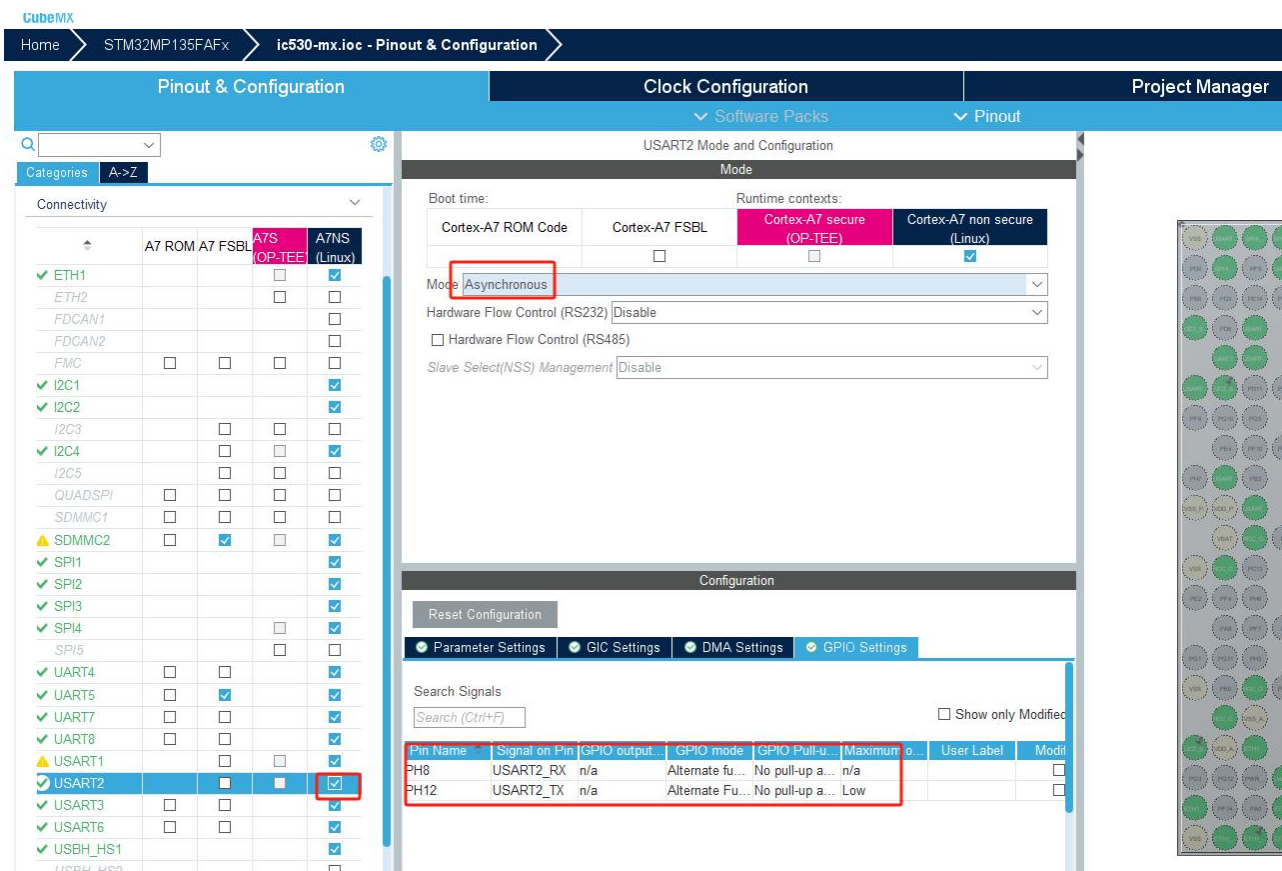


使能 usart2:

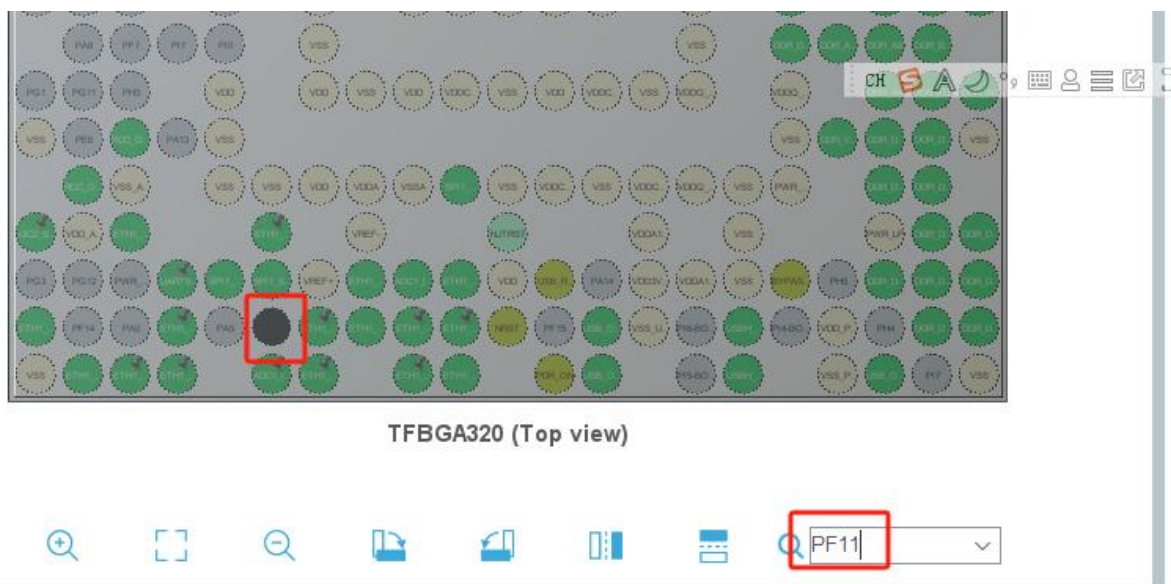
外设仅在 Linux 下使能即可，使能相关模式 cubemx 会根据当前 gpio 复用情况自动分配 gpio。



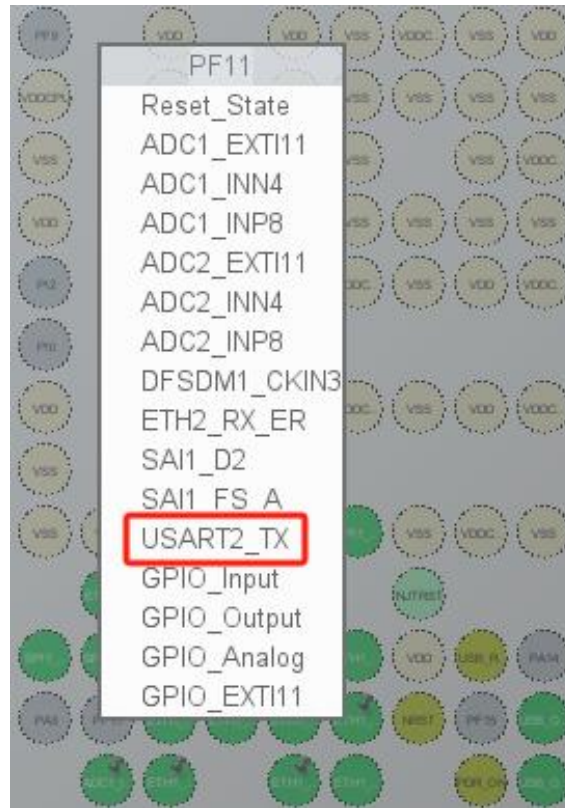
如配置为 Asynchronous 模式:



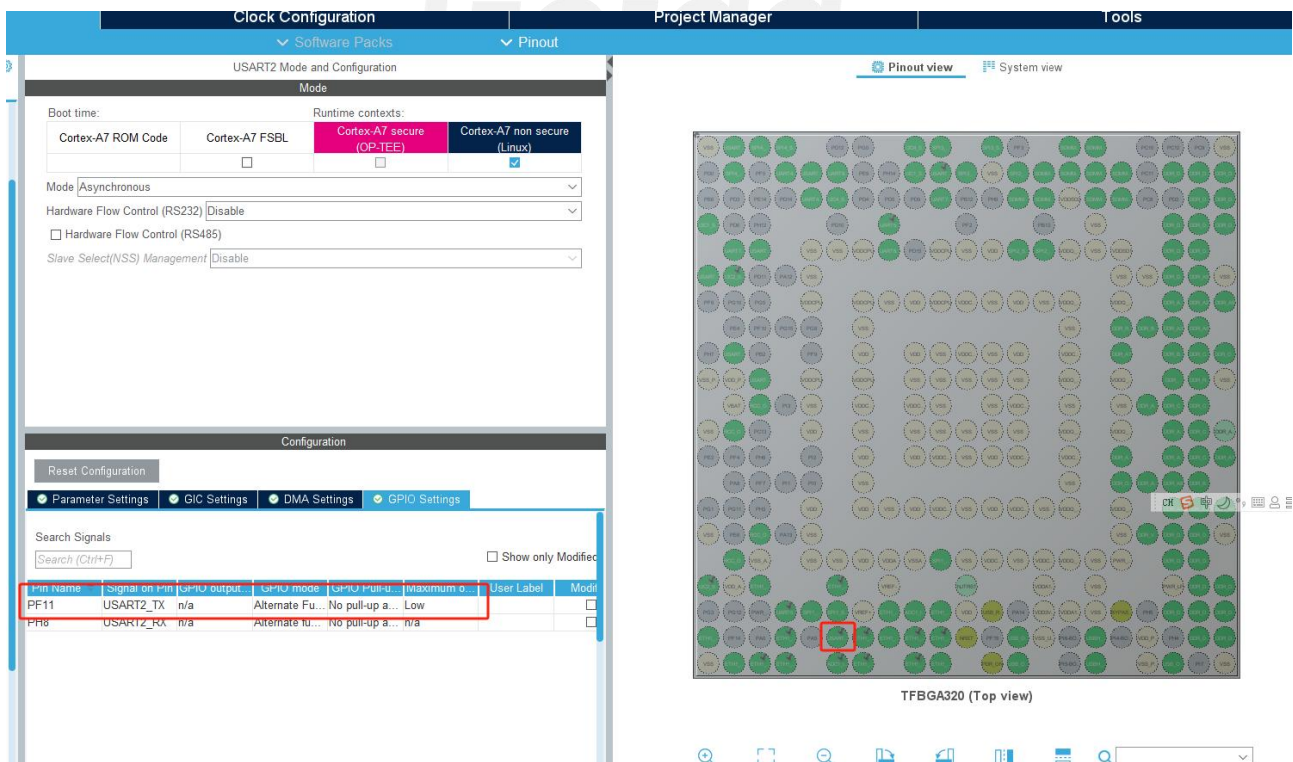
若自动配置 gpio 与硬件设计不一致, 可在右下角搜索框搜索对应 gpio 复用。如 USART2_RX 实际使用引脚为 PF11,



选择 PF11 选择复用模式即可。



如下 gpio 已经修改为 PF11,



配置完成后点击“GENERATE CODE”更新 dts。



Kernel/stm32mp135f-ic530-mx-mx.dts 自动生成。

```
usart2_pins_mx: usart2_mx-0 {
    pins1 {
        pinmux = <STM32_PINMUX('D', 15, AF1)>; /* USART2_RX */
        bias-disable;
    };
    pins2 {
        pinmux = <STM32_PINMUX('F', 11, AF1)>; /* USART2_TX */
        bias-disable;
        drive-push-pull;
        slew-rate = <0>;
    };
};

usart2_sleep_pins_mx: usart2_sleep_mx-0 {
    pins {
        pinmux = <STM32_PINMUX('D', 15, ANALOG)>; /* USART2_RX */
        pinmux = <STM32_PINMUX('F', 11, ANALOG)>; /* USART2_TX */
    };
};
```

```
/*
 *
 */

&usart2{
    pinctrl-names = "default", "sleep";
    pinctrl-0 = <&usart2_pins_mx>;
    pinctrl-1 = <&usart2_sleep_pins_mx>;
    status = "okay";

    /* USER CODE BEGIN usart2 */
    /* USER CODE END usart2 */
};
```

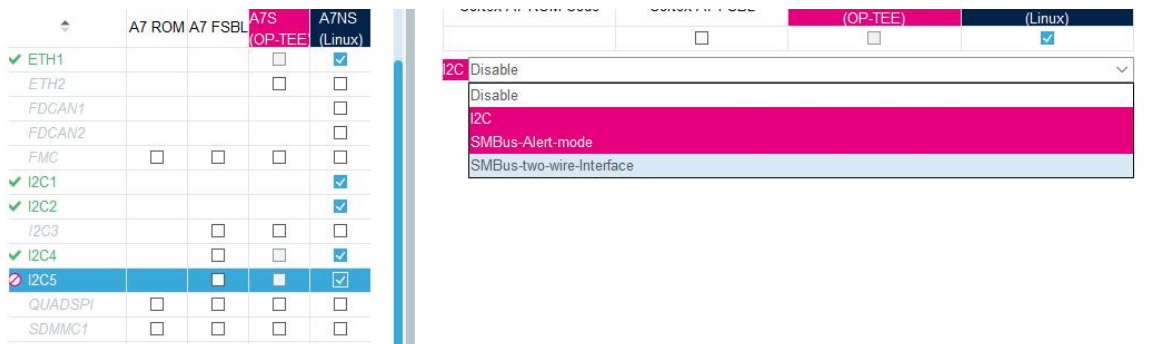
及 optee 下生成相关 clk 配置，具体内容可通过 git 查看。

在 aliases 内添加

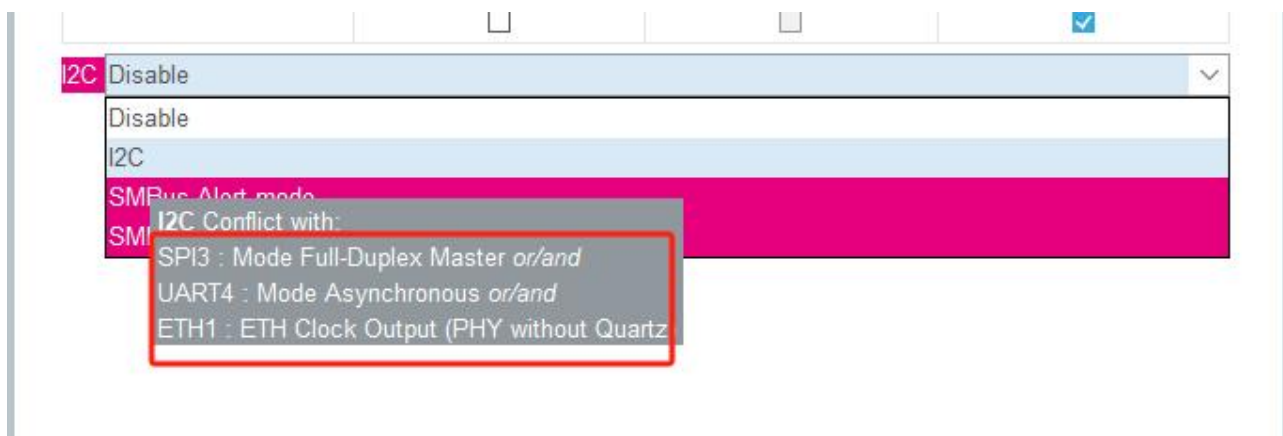
```
serial2 = &usart2;
```

若当前剩余 gpio 无法复用相关功能，则相关功能为红色表示无法设置。

如使能 i2c5 后，I2C 及 SMBus-Alert-mode 均为红色表示当前未使用的 gpio 无法配置该模式。需要根据硬件配置关闭占用的 gpio 资源。



将鼠标放在相关模式上会提醒引脚冲突及外设模块。



3.5 dts 说明

cubemx-ic530-ddr512-discrete-power\ic530-mx\DeviceTree\为各个源码下 dts 源码, dts 源码需要一一对应复制到 sdk 下。

Window 下安装 git 及 scp.sh 使用请参考《ST-A7-IC530 工业核心板_IM502_软件使用手册_Rev06.pdf》3.3.1 源码构建。

DeviceTree/ic530-mx 下源码介绍：

kernel/ optee-os/ scp.sh tf-a/ u-boot/

在 IC530 平台下，uboot 的 dts 与 kernel 下 dts 一致，cubemx 会自动拷贝 kernel 下源码到 u-boot 下，故用户只需修改 tf-a optee-os kernel 下 dts 源码即可。

3.6 dts 解析

3.6.1 includes

```
/* USER CODE BEGIN includes */
```

```
/* USER CODE END includes */
```

```
/* SPDX-License-Identifier: (GPL-2.0+ OR BSD-3-Clause) */
/*
 * Copyright (C) STMicroelectronics 2024 - All Rights Reserved
 * Author: STM32CubeMX code generation for STMicroelectronics.
 */

/* For more information on Device Tree configuration, please refer to
 * https://wiki.st.com/stm32mpu/wiki/Category:Device\_tree\_configuration
 */

/dts-v1/;
#include <dt-bindings/pinctrl/stm32-pinfunc.h>

#include "stm32mpl35.dtsi"
#include "stm32mpl3xf.dtsi"

/* USER CODE BEGIN includes */
#include <dt-bindings/gpio/gpio.h>
#include <dt-bindings/input/input.h>
#include <dt-bindings/leds/common.h>
#include <dt-bindings/rtc/rtc-stm32.h>
#define ic530_4g_ddr //if define use 4g ddr, other 2g ddr
/* USER CODE END includes */
```

中间代码为用户手动修改区域，cubemx 不会对此区域内代码做任何修改，其他区域代码为 cubemx 自动生成无需更改，故用户手动添加头文件或者宏定义等需在/* USER CODE BEGIN includes */ /* USER CODE END includes */之内添加。

依次类推，用户手动修改代码区域为/* USER CODE BEGIN * */ /* USER CODE END **** */**

3.6.2 memory

内存分区区域代码，该部分代码已经根据核心板型号配置完善，用户无需改动。

```

/ {
    model = "STMicroelectronics custom STM32CubeMX board - openstlinux-5.15-yocto-kirkstone-mp1-v22.11.23";
    compatible = "st,stm32mp135f-ic530-mx-mx", "st,stm32mp135";

    memory@c0000000 {
        device_type = "memory";
        reg = <0xc0000000 0x20000000>;

        /* USER CODE BEGIN memory */
        /* USER CODE END memory */
    };

    reserved-memory {
        #address-cells = <1>;
        #size-cells = <1>;
        ranges;

        /* USER CODE BEGIN reserved-memory */
        #ifdef ic530_4g_ddr
        optee_framebuffer@dd000000 {
            reg = <0xdd000000 0x1000000>;
            no-map;
        };

        optee@de000000 {
            reg = <0xde000000 0x2000000>;
            no-map;
        };
        #else
        optee_framebuffer@cd000000 {
            reg = <0xcd000000 0x1000000>;
            no-map;
        };

        optee@ce000000 {
            reg = <0xce000000 0x2000000>;
            no-map;
        };
        #endif
        /* USER CODE END reserved-memory */
    };
};

```

3.6.3 USER CODE BEGIN root

```

/* USER CODE BEGIN root */

aliases{

    ethernet0 = &eth1;

    serial0 = &uart5;

    serial1 = &usart1;

    serial2 = &usart2;

    serial3 = &usart3;

    serial4 = &usart6;

    serial5 = &uart5;

    serial6 = &uart7;

    serial7 = &uart8;

```

```
};

:

:

vddcore: vddcore {
    compatible = "regulator-fixed";
    regulator-name = "vddcore";
    regulator-min-microvolt = <1250000>;
    regulator-max-microvolt = <1250000>;
    regulator-off-in-suspend;
    regulator-always-on;
};

watchdog: watchdog {
    compatible = "linux,wdt-gpio";
    //根据硬件选择相应的 gpio 并使能 status
    //en-gpios = <&gpioh 5 GPIO_ACTIVE_HIGH>; //GPIO_ACTIVE_
LOW disabled dog ,GPIO_ACTIVE_HIGH enable dog
    //wdi-gpios = <&gpiop 15 GPIO_ACTIVE_HIGH>; //
    hw_algo = "toggle";
    hw_margin_ms = <100>;
    always-running;

    status = "disabled";
};

/* USER CODE END root */
```

该区域内，可添加背光、看门狗、led、按键、regulator 等相关代码。

新增以太网和串口节点,需要添加 aliases,如 cubemx 端使能配置 usart2 后,需在 aliases 添加 serial2 = &usart2;否则 usart2 无法加载,usart2 节点在 cubemx 端使能及配置 gpio 后, cubemx 可自动生产代码,具体参考 2.2。

```
aliases{
    ethernet0 = &eth1;
    serial0 = &uart5;
    serial1 = &usart1;
    serial2 = &usart2;
    serial3 = &usart3;
    serial4 = &usart6;
    serial5 = &uart5;
    serial6 = &uart7;
    serial7 = &uart8;

};
```

serialN 对应系统驱动后节点/dev/ttySTMN。

3.6.4 pinctrl

该节点下为 gpio 复用配置, cubemx 自动生成,若有需要配置个别 gpio 如下:

```
/* USER CODE BEGIN pinctrl */
goodix_pins_a: goodix-0 {
    pins {
        pinmux = <STM32_PINMUX('C', 2, GPIO)>;
        bias-pull-down;
    };
};

/* USER CODE END pinctrl */
```

3.6.5 外设 usercode

每个外设节点内都有 usercode

```
/* USER CODE BEGIN i2c1 */
```

```
/* USER CODE END i2c1 */
```

```
/* USER CODE BEGIN spi1 */
```

```
/* USER CODE END spi1 */
```

```
/* USER CODE BEGIN eth1 */
```

```
/* USER CODE END eth1 */
```

使用方法请参考 demo dts。

Lierda
利 尔 达

3.7 Debug 串口修改

当前源码及 dts 中默认使用 uart5 作为 debug 口，若修改其他串口为 debug 节点需要修改如下文件，如修改 uart4 为 debug。

使能 uart4 的 FSBL，并点击“GENERATE CODE”更新 dts。

	A7 ROM	Cortex-A7 FSBL	A7S (OP-TEE)	A7NS (Linux)
✓ ETH1			<input type="checkbox"/>	<input checked="" type="checkbox"/>
ETH2			<input type="checkbox"/>	<input type="checkbox"/>
FDCAN1				<input type="checkbox"/>
FDCAN2				<input type="checkbox"/>
FMC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ I2C1				<input checked="" type="checkbox"/>
✓ I2C2				<input checked="" type="checkbox"/>
I2C3		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ I2C4		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
⊗ I2C5		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
QUADSPI	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
SDMMC1	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
✓ SPI1				<input checked="" type="checkbox"/>
✓ SPI2				<input checked="" type="checkbox"/>
✓ SPI3				<input checked="" type="checkbox"/>
✓ SPI4			<input type="checkbox"/>	<input checked="" type="checkbox"/>
SPI5			<input type="checkbox"/>	<input type="checkbox"/>
✓ UART4	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>
✓ UART7	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>
✓ UART8	<input type="checkbox"/>	<input type="checkbox"/>		<input checked="" type="checkbox"/>
⚠ USART1		<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>

(1) 将 tf-a optee-os kernel 下 dts 内的 serial0 = &uart5;修改为 serial0 = &uart4;

(2) 复制 kernel dts pinctl 下对 uart4 的 gpio 复用配置 optee-os 内 dts 的关于 uart5 的配置

```
&pinctl {
    /* USER CODE BEGIN pinctl */
    uart5_pins_mx: uart5_mx-0 {
        pins1 {
            pinmux = <STM32_PINMUX('E', 7, AF8)>; /* UART5_TX */
            bias-disable;
        }
    }
}
```

```

drive-push-pull;
slew-rate = <0>;
};

pins2 {
    pinmux = <STM32_PINMUX('F', 13, AF8)>; /* UART5_RX */
    bias-disable;
};
};

/* USER CODE END pinctrl */
};

```

及负责 kernel dts 下节点uart5 复制到 optee-os 下的

```

/* USER CODE BEGIN addons */
/* USER CODE END addons */

```

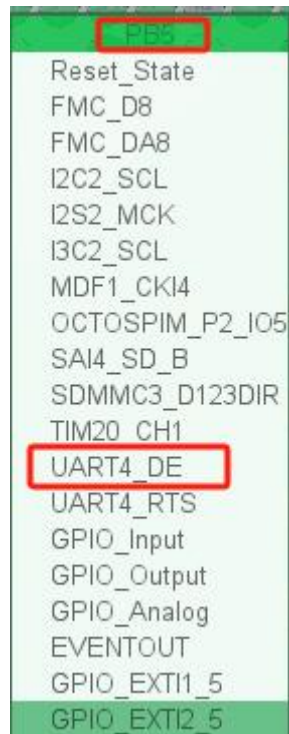
(3) optee 源码下 core/arch/arm/plat-stm32mp1/conf.mk

修改 CFG_STM32_EARLY_CONSOLE_UART ?= 5 为 CFG_STM32_EARLY_CONSOLE_UART ?= 4

3.8 串口复用为 485

在 st 平台下，串口的 485 DE 引脚为 gpio 的复用功能之一，故不是所有的串口都可复用为 485 模式，485 的 DE 引脚需选择特定 gpio。

485 的 de 引脚为 gpio 复用模式，在 cubemx 中查看引脚复用模式类似如下，如图表明 PB5 可以复用为 UART4_DE，及该引脚是 uart4 的 485 DE 引脚。



在 ic610 和 ic530 下针对 485 功能总结如下：

ic530:

可复用为 485 的串口	485DE 可选择 gpio
UART4	PA6、PA15、PE6
UART7	PB12、PE4、PF10
UART8	PE12、PE14
USART1	PA12、PC2
USART2	PA1、PD4
USART3	PD13、PE3、PG8
USART6	PF10、PE2、PG12

ic610:

可复用为 485 的串口	485DE 可选择 gpio
UART4	PB5、PD9
UART5	PC11 、 PG8
UART7	PD1、PH7
UART8	PB15、PD12、PF9
UART9	PF11 、 PI11
USART1	PA1、 PB9 、 PE12、 PI2
USART2	PA5 、 PC2
USART3	PC11、 PE5 、 PF14
USART6	PC3 、 PG5
LPUART1	PZ2 、 PZ5

Lierda
利 尔 达