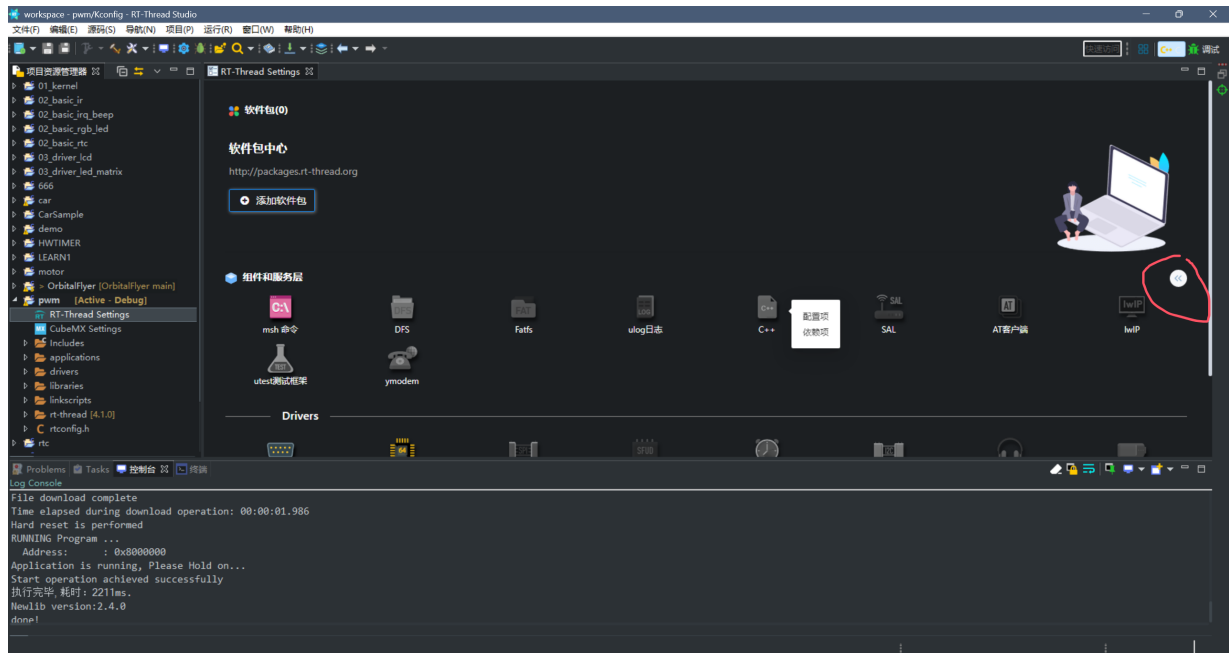


PWM设备

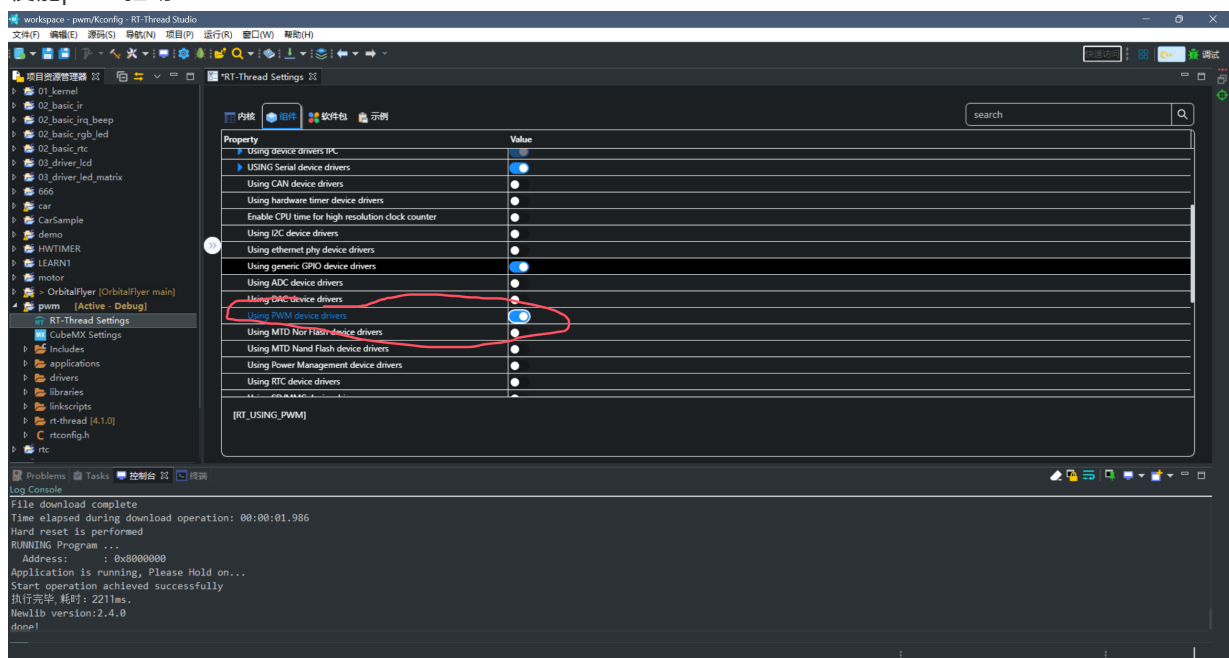
使用设备:STM32G431CBT6

操作流程:

1.新建项目,配置RT_Thread_Setting



使能pwm驱动:



ctrl+s 保存后退出

2.配置CubeMX

The image displays three sequential screenshots of the STM32CubeMX software interface, showing the configuration process for an STM32G431CBT6 microcontroller.

Screenshot 1: Pinout & Configuration

- The **Pinout & Configuration** tab is active.
- The **SYS Mode and Configuration** section is expanded.
- The **Mode** dropdown is set to **Debug (Serial Wire)**.
- The **Configuration** section shows a warning: "Warning: This peripheral has no parameters to be configured."

Screenshot 2: RCC Mode and Configuration

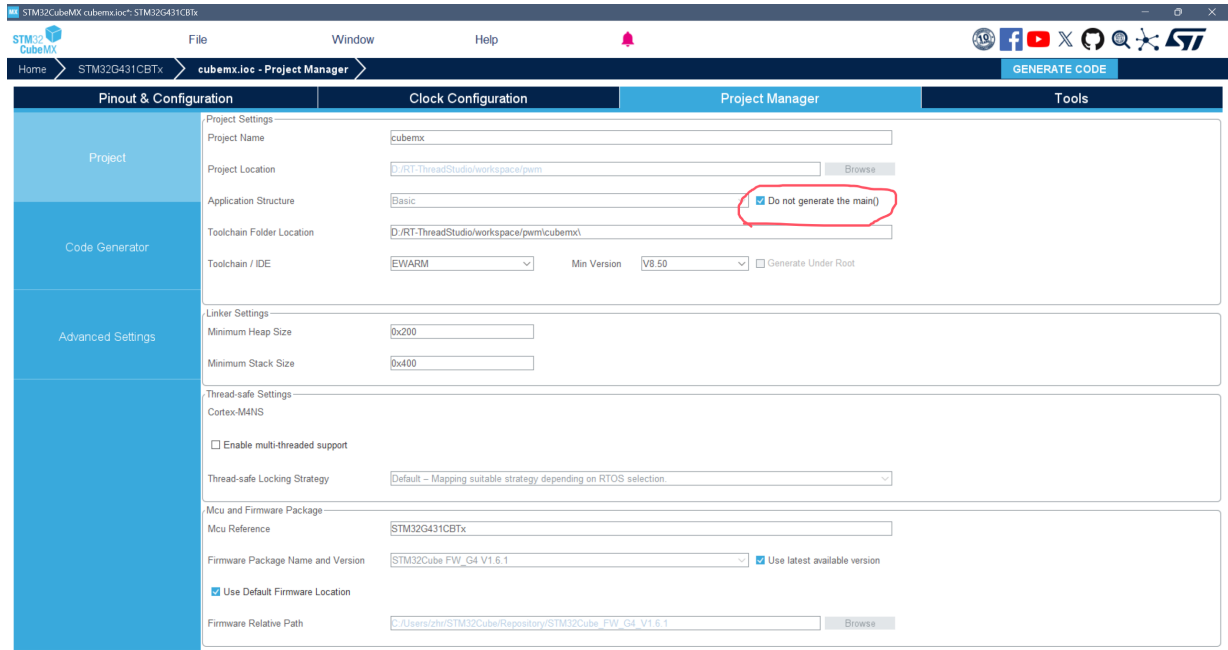
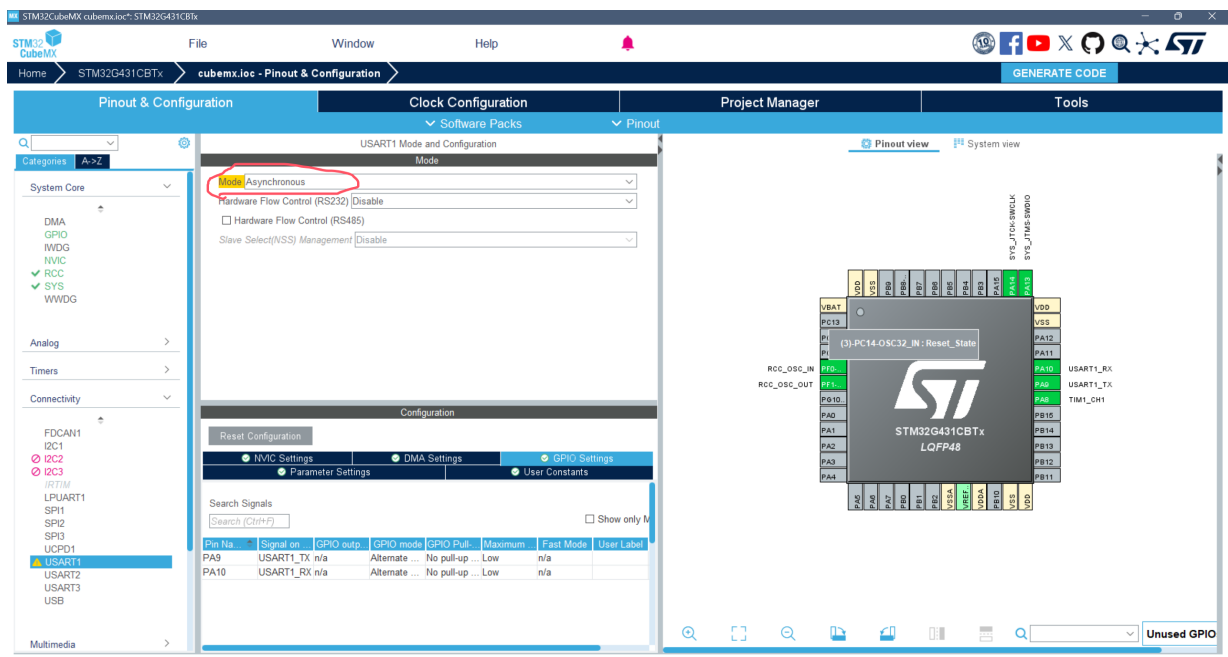
- The **RCC Mode and Configuration** section is expanded.
- The **Mode** dropdown is set to **High-Speed Clock (HSE)**.
- The **Configuration** section shows the **Parameter Settings** tab selected.
- The **System Parameters** section is expanded, showing the following settings:

Parameter	Value
VDD voltage (V)	3.3 V
Instruction Cache	Enabled
Prefetch Buffer	Disabled
Data Cache	Enabled
Flash Latency (WS)	4 WS (5 CPU cycle)

Screenshot 3: TIM1 Mode and Configuration

- The **TIM1 Mode and Configuration** section is expanded.
- The **Mode** dropdown is set to **Slave Mode**.
- The **Configuration** section shows the **Parameter Settings** tab selected.
- The **Counter Settings** section is expanded, showing the following settings:

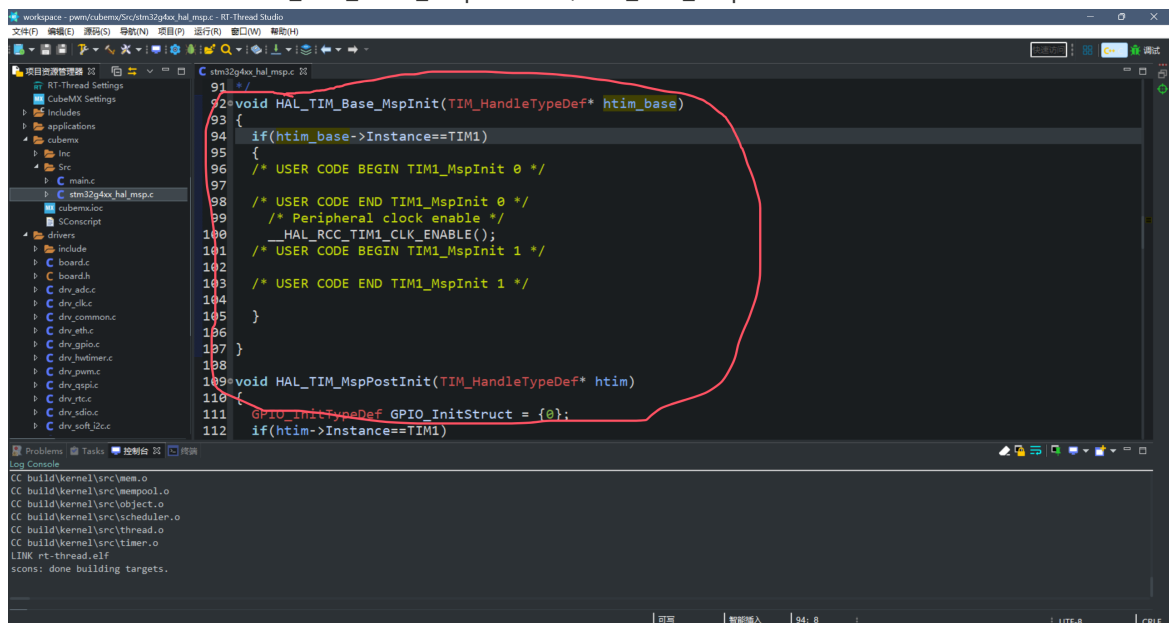
Parameter	Value
Prescaler (PSC - 16 bits value)	0
Counter Mode	Up
Dithering	Disable
Counter Period (AutoReload Register - 1... 65535)	



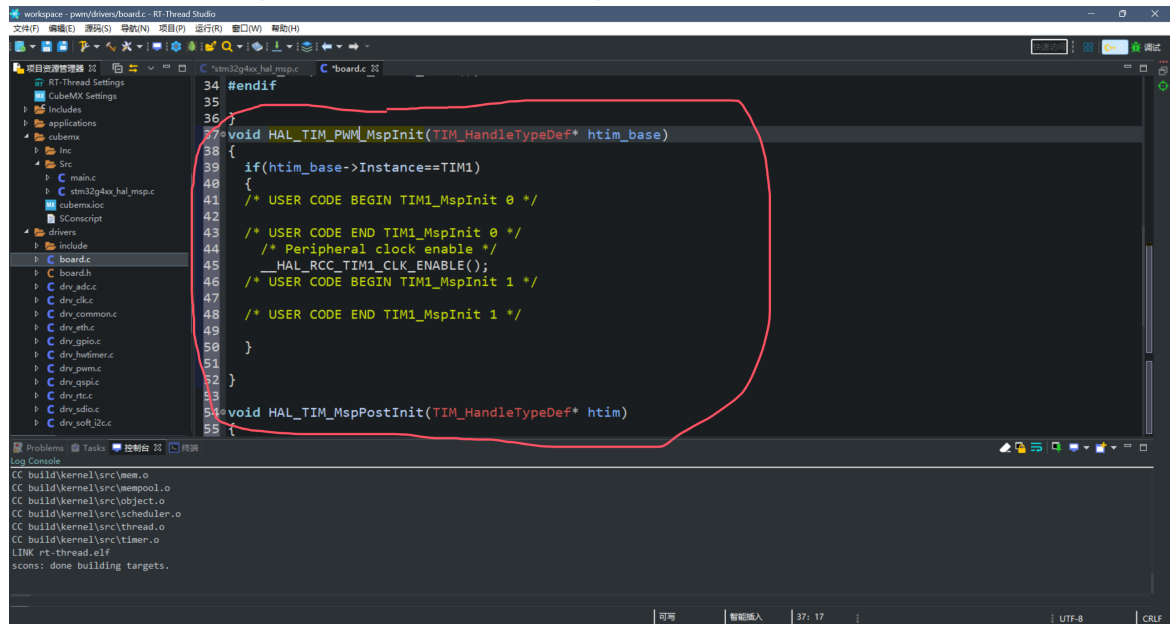
配置完后点击右上角生成,生成完毕后一定要叉掉CubeMX!!!

3.配置rtt的环境

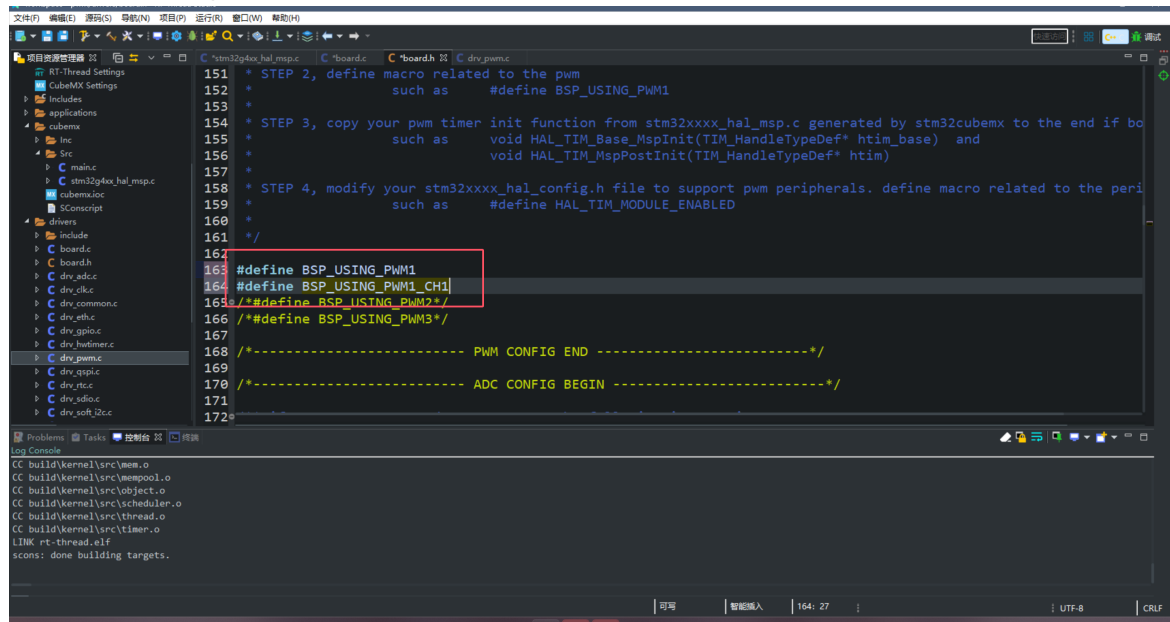
1. 将CubeMX中生成的HAL_TIM_Base_MspInit函数,HAL_TIM_MspPostInit函数剪切在board.c的末尾



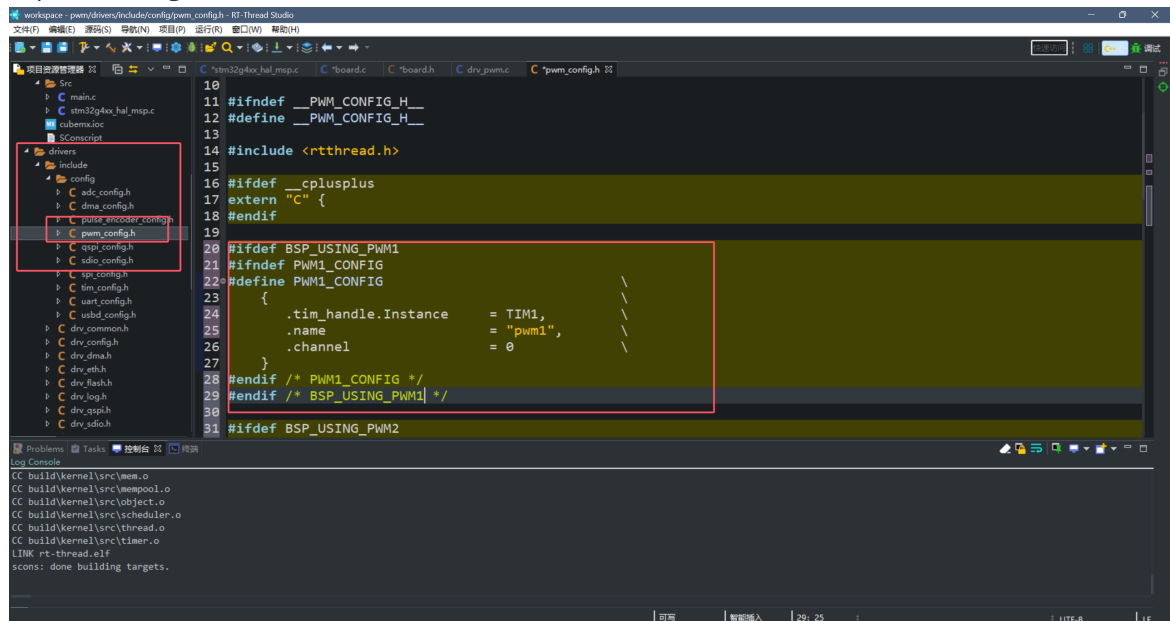
2. 将HAL_TIM_Base_MspInit改为HAL_TIM_PWM_MspInit



3. 在board.h中增加pwm的宏定义

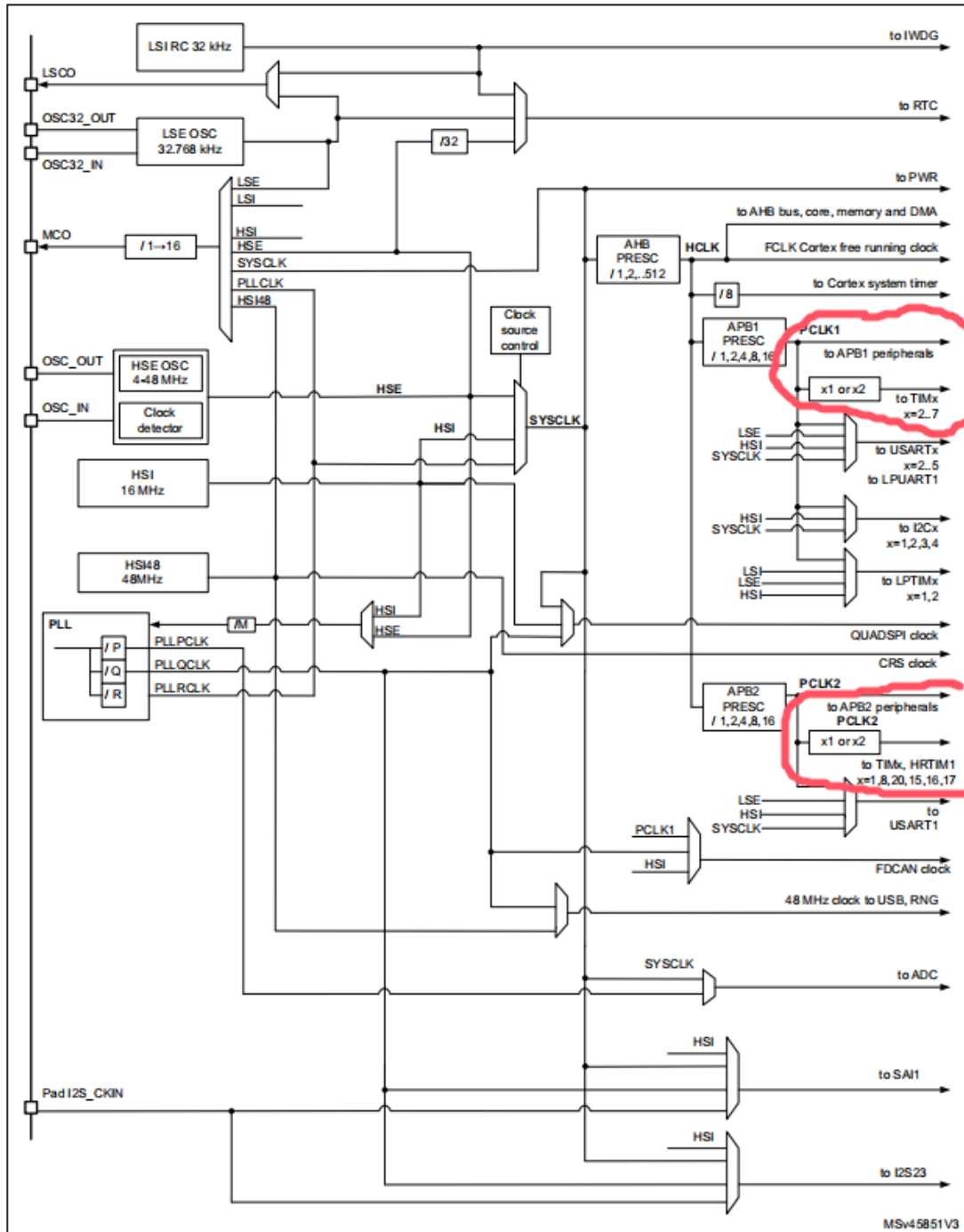


4. 在pwm_config.h中增加定义



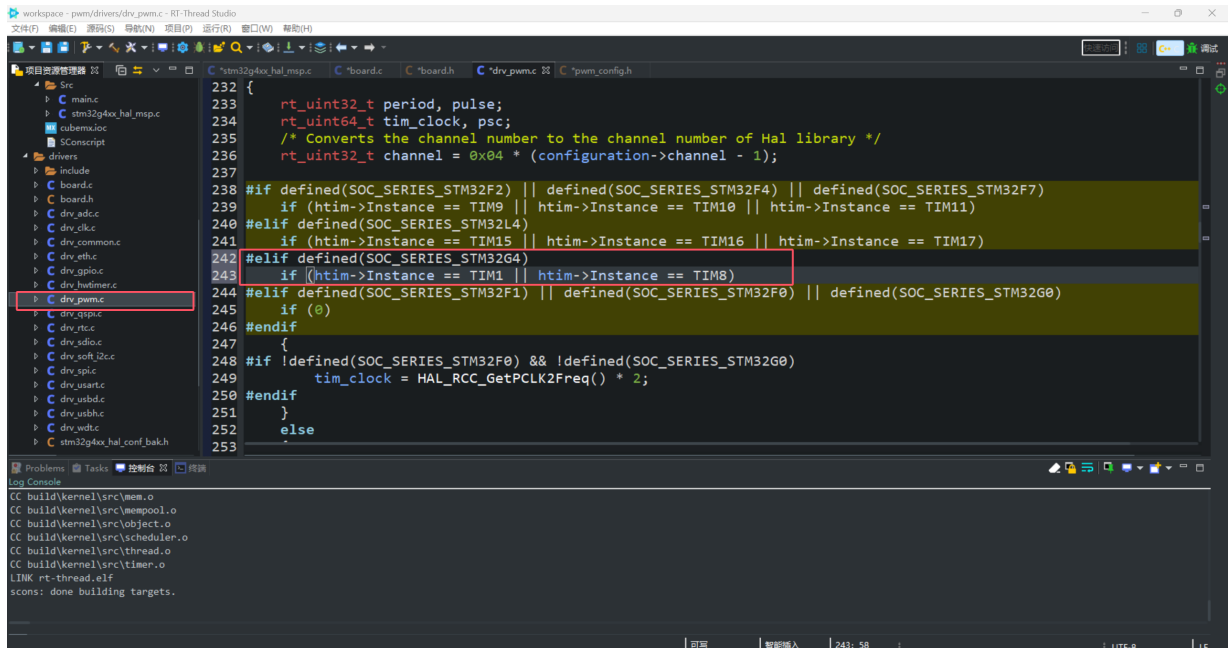
5. 注意到drv_pwm.c中的底层驱动没有针对G4系列的定义,所以需要手动加

Figure 14. Clock tree

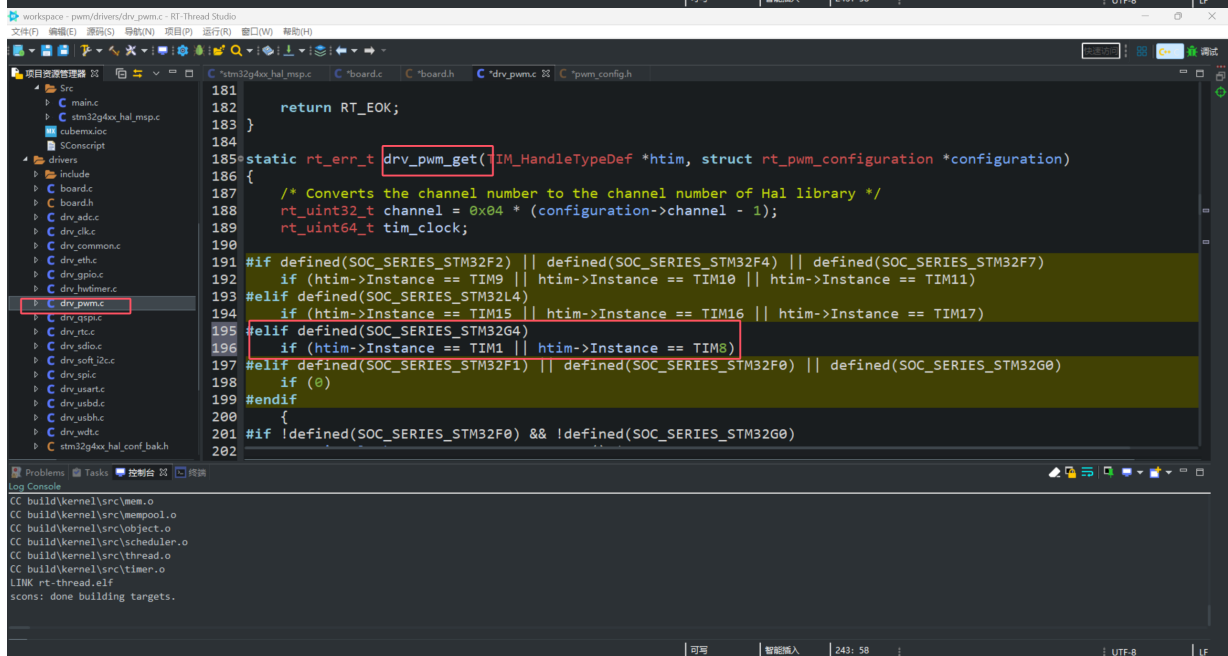


1. For full details about the internal and external clock source characteristics, please refer to the "Electrical

由参考时钟的时钟树可以得出,定时器1,8使用PCLK2,而其他使用PCLK1,所以在drv_pwm_set函数,drv_pwm_get函数中作如下添加:

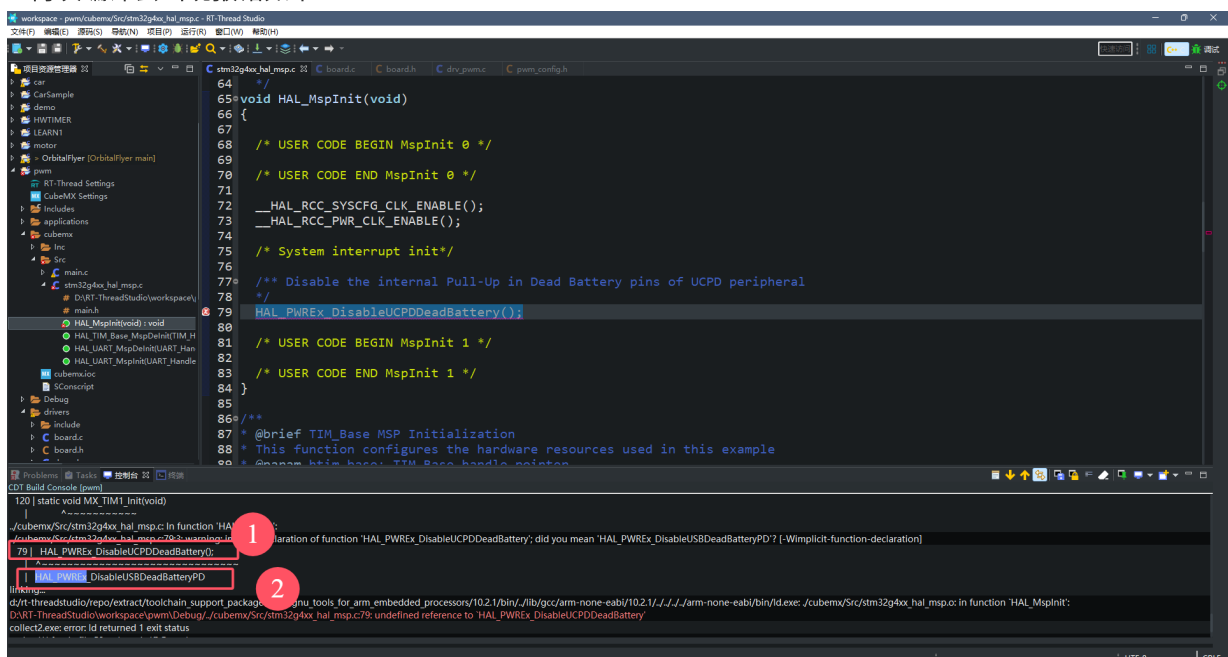


```
232 {
233     rt_uint32_t period, pulse;
234     rt_uint64_t tim_clock, psc;
235     /* Converts the channel number to the channel number of Hal library */
236     rt_uint32_t channel = 0x04 * (configuration->channel - 1);
237
238     #if defined(SOC_SERIES_STM32F2) || defined(SOC_SERIES_STM32F4) || defined(SOC_SERIES_STM32F7)
239     if (htim->Instance == TIM9 || htim->Instance == TIM10 || htim->Instance == TIM11)
240     #elif defined(SOC_SERIES_STM32L4)
241     if (htim->Instance == TIM15 || htim->Instance == TIM16 || htim->Instance == TIM17)
242     #elif defined(SOC_SERIES_STM32G4)
243     if (htim->Instance == TIM1 || htim->Instance == TIM8)
244     #elif defined(SOC_SERIES_STM32F1) || defined(SOC_SERIES_STM32F0) || defined(SOC_SERIES_STM32G0)
245     if (0)
246     #endif
247     {
248     #if !defined(SOC_SERIES_STM32F0) && !defined(SOC_SERIES_STM32G0)
249         tim_clock = HAL_RCC_GetPCLK2Freq() * 2;
250     #endif
251     }
252     else
253     {
```



```
181
182     return RT_EOK;
183 }
184
185 static rt_err_t drv_pwm_get(TIM_HandleTypeDef *htim, struct rt_pwm_configuration *configuration)
186 {
187     /* Converts the channel number to the channel number of Hal library */
188     rt_uint32_t channel = 0x04 * (configuration->channel - 1);
189     rt_uint64_t tim_clock;
190
191     #if defined(SOC_SERIES_STM32F2) || defined(SOC_SERIES_STM32F4) || defined(SOC_SERIES_STM32F7)
192     if (htim->Instance == TIM9 || htim->Instance == TIM10 || htim->Instance == TIM11)
193     #elif defined(SOC_SERIES_STM32L4)
194     if (htim->Instance == TIM15 || htim->Instance == TIM16 || htim->Instance == TIM17)
195     #elif defined(SOC_SERIES_STM32G4)
196     if (htim->Instance == TIM1 || htim->Instance == TIM8)
197     #elif defined(SOC_SERIES_STM32F1) || defined(SOC_SERIES_STM32F0) || defined(SOC_SERIES_STM32G0)
198     if (0)
199     #endif
200     {
201     #if !defined(SOC_SERIES_STM32F0) && !defined(SOC_SERIES_STM32G0)
202     {
```

6.再次编译会出现报错如下:



```
79 | HAL_PWREx_DisableUCPDDeadBattery();
80 |
81 |
82 |
83 |
84 |
85 |
86 |
87 |
88 |
89 |
```

找到错误出处,将圈一函数,替换为圈二函数

```

void HAL_MspInit(void)
{
    /* USER CODE BEGIN MspInit 0 */

    /* USER CODE END MspInit 0 */

    __HAL_RCC_SYSCFG_CLK_ENABLE();
    __HAL_RCC_PWR_CLK_ENABLE();

    /* System interrupt init*/

    /** Disable the internal Pull-Up in Dead Battery pin
    */
    HAL_PWREx_DisableUSBDeadBatteryPD();

    /* USER CODE BEGIN MspInit 1 */

    /* USER CODE END MspInit 1 */
}

```

此时再编译,编译成功.

7.书写main函数如下:

```

1  #include <rtthread.h>
2
3  #define DBG_TAG "main"
4  #define DBG_LVL DBG_LOG
5  #include <rtdbg.h>
6
7  #include <rtdevice.h>
8  #include <board.h>
9
10 #define PWM_DEV_NAME      "pwm1"
11 #define PWM_DEV_CHANNEL  1
12 static struct rt_device_pwm* pwm_dev;
13
14 rt_uint32_t period = 55560;
15 rt_uint32_t pulse = 0;
16
17 int main(void)
18 {
19     pwm_dev = (struct rt_device_pwm*)rt_device_find(PWM_DEV_NAME);
20     if(pwm_dev == RT_NULL){
21         rt_kprintf("PWM device not found\n");
22         return -1;
23     }
24     rt_err_t temp = rt_pwm_set(pwm_dev, PWM_DEV_CHANNEL, period, pulse);
25     if(temp != RT_EOK)
26     {
27         rt_kprintf("Failed to set PWM parameters for channel %d\n", pulse);

```

```

28     }
29     else
30     {
31         rt_kprintf("set PWM parameters for channel %d\n",pulse);
32     }
33     rt_err_t temp1 = rt_pwm_enable(pwm_dev,PWM_DEV_CHANNEL);
34     if(temp1!=RT_EOK)
35     {
36         rt_kprintf("Failed to enable PWM parameters for channel %d\n",pulse);
37     }
38     else
39     {
40         rt_kprintf("enable PWM parameters for channel %d\n",pulse);
41     }
42     int count = 1;
43
44     while (count++)
45     {
46
47         rt_thread_mdelay(1000);
48     }
49
50     return RT_EOK;
51 }

```

8.编译下载后,查看控制台.

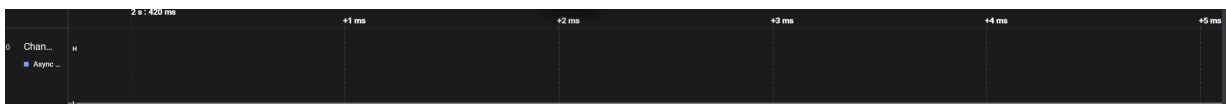
出现下图则成功:

```

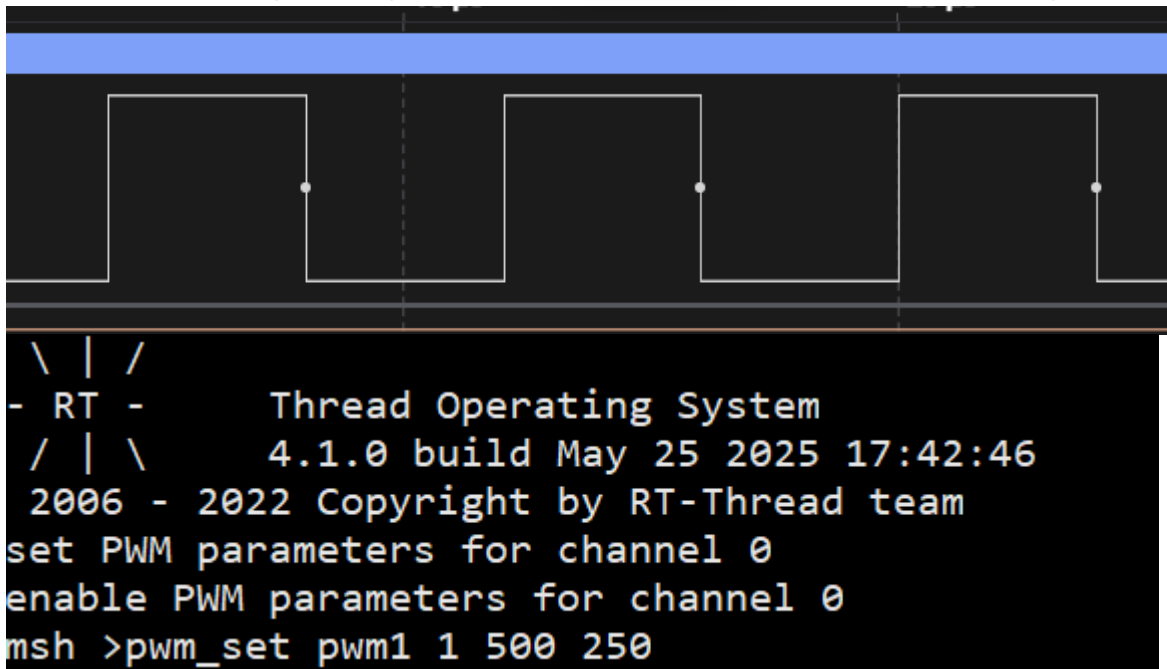
 \ | /
- RT -   Thread Operating System
 / | \   4.1.0 build May 25 2025 17:42:46
2006 - 2022 Copyright by RT-Thread team
set PWM parameters for channel 0
enable PWM parameters for channel 0
msh >

```

9.可以使用Finsh命令控制:



- 目前是全部低电平,输入pwm_set pwm1 1 500 250可以生成周期为500hz,脉冲宽度为250的pwm波,



- 使用pwm_get pwm1 1可以读取设备当前状态:

```
msh >pwm_set pwm1 1 500 250  
msh >pwm_get pwm1 1  
Get info of device: [pwm1]:  
period      : 500  
pulse       : 250  
Duty cycle  : 50%  
msh >
```