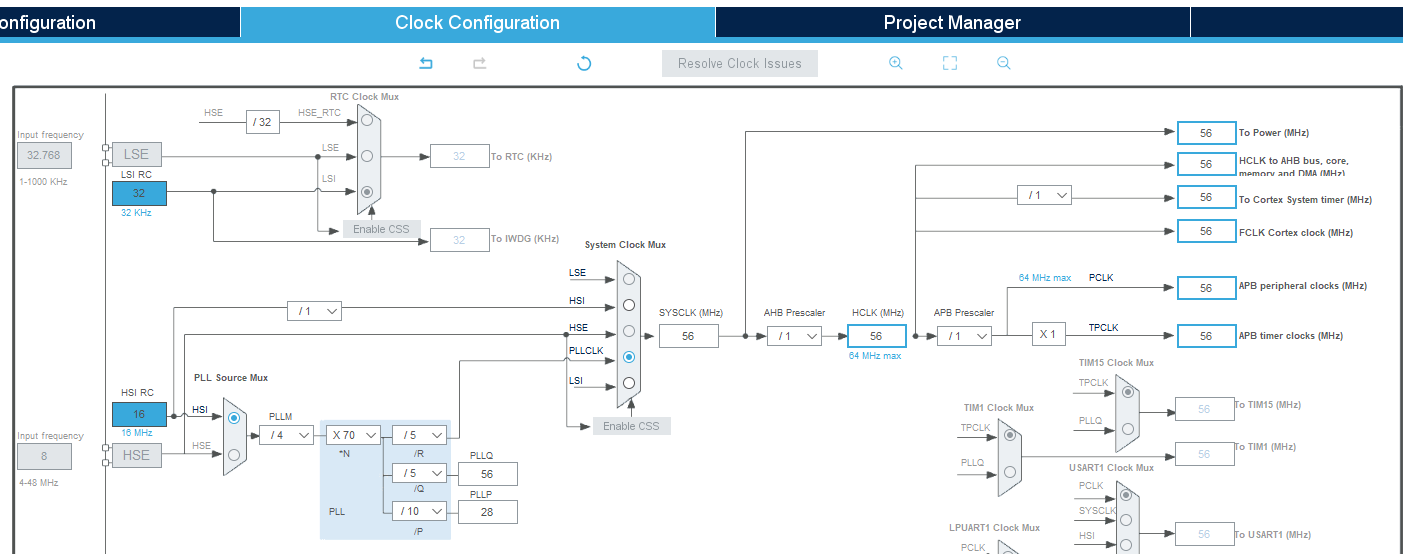
定时器闹钟与日历功能整合 RTC\_CalendarAndAlarm

# 创建工程

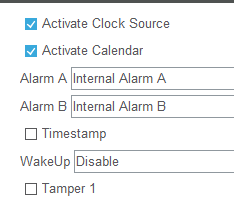
将PA5配置为GPIO输出模式驱动LED4灯

将RCC最下方Generate the peripherals clock configuration的配置，由“TRUE”改为“FALSE”

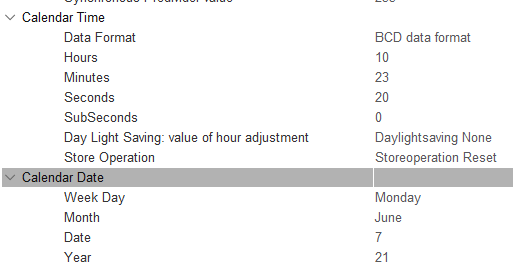
按照下图所示配置时钟



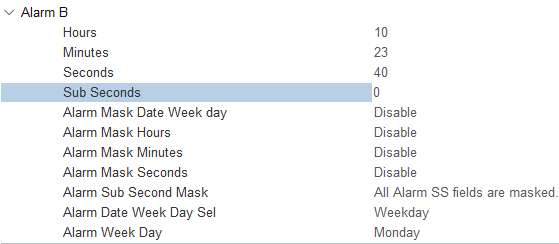
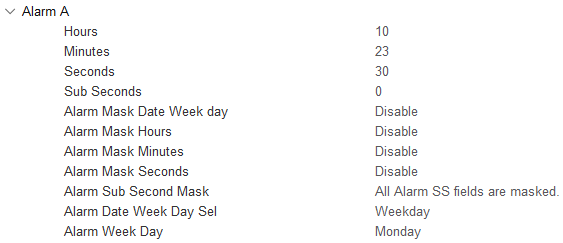
RTC闹钟、日历使能



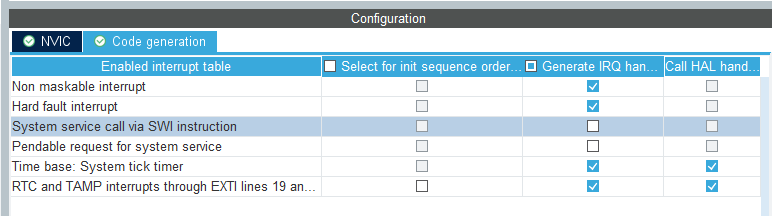
设置起始时间与日期



设置闹钟A与闹钟B



中断设置：使能19-21号RTC中断，关闭如下中断



# 在mian.c文件下添加代码

头文件、函数声明、变量定义

#include <stdio.h>

uint8\_t aShowTime[8] = "hh:ms:ss";

uint8\_t aShowDate[10] = "dd-mm-yyyy";

\_\_IO uint32\_t RTCStatus = 0;

static void RTC\_CalendarShow(uint8\_t \*showtime, uint8\_t \*showdate);

循环之前设置RTC状态RTCStatus = 1;

在循环中实时输出时间 RTC\_CalendarShow(aShowTime, aShowDate);

添加两个闹钟的中断回调函数

void HAL\_RTC\_AlarmAEventCallback(RTC\_HandleTypeDef \*hrtc)

{

/\* 点亮LED灯 \*/

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_5,1);

}

void HAL\_RTCEx\_AlarmBEventCallback(RTC\_HandleTypeDef \*hrtc)

{

/\* 关闭LED灯 \*/

HAL\_GPIO\_WritePin(GPIOA,GPIO\_PIN\_5,0);

}

添加获取时间的子函数

static void RTC\_CalendarShow(uint8\_t \*showtime, uint8\_t \*showdate)

{

RTC\_DateTypeDef sdatestructureget;

RTC\_TimeTypeDef stimestructureget;

/\* Get the RTC current Time \*/

HAL\_RTC\_GetTime(&hrtc, &stimestructureget, RTC\_FORMAT\_BIN);

/\* Get the RTC current Date \*/

HAL\_RTC\_GetDate(&hrtc, &sdatestructureget, RTC\_FORMAT\_BIN);

/\* Display time Format : hh:mm:ss \*/

sprintf((char \*)showtime, "%2d:%2d:%2d", stimestructureget.Hours, stimestructureget.Minutes, stimestructureget.Seconds);

/\* Display date Format : mm-dd-yy \*/

sprintf((char \*)showdate, "%2d-%2d-%2d", sdatestructureget.Month, sdatestructureget.Date, 2000 + sdatestructureget.Year);

}

# **更改stm32g0xx\_hal\_msp.c文件**

全文替换为下方：

#include "main.h"

static uint32\_t RtcClockSource = RCC\_RTCCLKSOURCE\_LSI;

void HAL\_MspInit(void)

{

\_\_HAL\_RCC\_SYSCFG\_CLK\_ENABLE();

\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();

HAL\_SYSCFG\_StrobeDBattpinsConfig(SYSCFG\_CFGR1\_UCPD1\_STROBE | SYSCFG\_CFGR1\_UCPD2\_STROBE);

}

void HAL\_RTC\_MspInit(RTC\_HandleTypeDef\* hrtc)

{

if(hrtc->Instance==RTC)

{

RCC\_OscInitTypeDef RCC\_OscInitStruct = {0};

RCC\_PeriphCLKInitTypeDef PeriphClkInitStruct = {0};

\_\_HAL\_RCC\_PWR\_CLK\_ENABLE();

HAL\_PWR\_EnableBkUpAccess();

HAL\_RCCEx\_GetPeriphCLKConfig(&PeriphClkInitStruct);

if (PeriphClkInitStruct.RTCClockSelection == RtcClockSource)

{ }

else

{

PeriphClkInitStruct.PeriphClockSelection = RCC\_PERIPHCLK\_RTC;

if (PeriphClkInitStruct.RTCClockSelection != RCC\_RTCCLKSOURCE\_NONE)

{

PeriphClkInitStruct.RTCClockSelection = RCC\_RTCCLKSOURCE\_NONE;

if (HAL\_RCCEx\_PeriphCLKConfig(&PeriphClkInitStruct) != HAL\_OK)

Error\_Handler();

}

RCC\_OscInitStruct.OscillatorType = RCC\_OSCILLATORTYPE\_LSI | RCC\_OSCILLATORTYPE\_LSE;

RCC\_OscInitStruct.PLL.PLLState = RCC\_PLL\_NONE;

RCC\_OscInitStruct.LSIState = RCC\_LSI\_ON;

RCC\_OscInitStruct.LSEState = RCC\_LSE\_OFF;

if (HAL\_RCC\_OscConfig(&RCC\_OscInitStruct) != HAL\_OK)

Error\_Handler();

PeriphClkInitStruct.RTCClockSelection = RtcClockSource;

if (HAL\_RCCEx\_PeriphCLKConfig(&PeriphClkInitStruct) != HAL\_OK)

Error\_Handler();

}

\_\_HAL\_RCC\_RTC\_ENABLE();

\_\_HAL\_RCC\_RTCAPB\_CLK\_ENABLE();

HAL\_NVIC\_SetPriority(RTC\_TAMP\_IRQn, 0, 0);

HAL\_NVIC\_EnableIRQ(RTC\_TAMP\_IRQn);

}

}

void HAL\_RTC\_MspDeInit(RTC\_HandleTypeDef\* hrtc)

{

if(hrtc->Instance==RTC)

{

\_\_HAL\_RCC\_RTC\_DISABLE();

\_\_HAL\_RCC\_RTCAPB\_CLK\_DISABLE();

HAL\_NVIC\_DisableIRQ(RTC\_TAMP\_IRQn);

}

}

# 调试

进入Debug可看到时间与日期信息，闹钟A触发LED灯亮，闹钟B触发LED灯灭