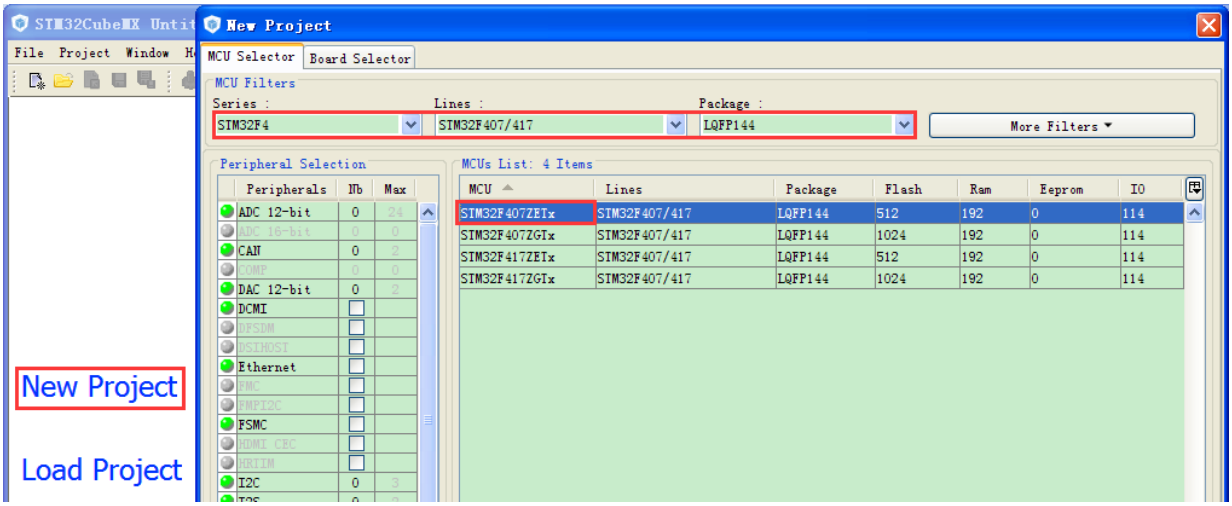
STM32Cube 学习之十二：RTC

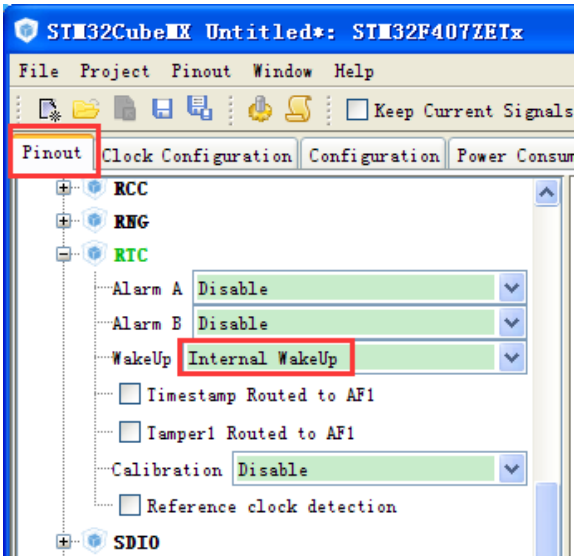
假设已经安装好 STM32CubeMX 和 STM32CubeF4 支持包。

Step1.打开 STM32CubeMX，点击“New Project”，选择芯片型号，STM32F407ZETx。

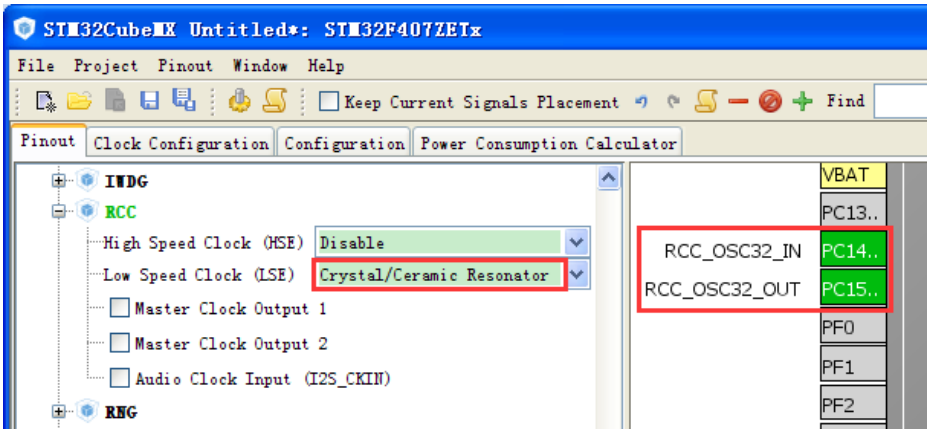


Step2. 在 Pinout 界面下配置引脚功能。

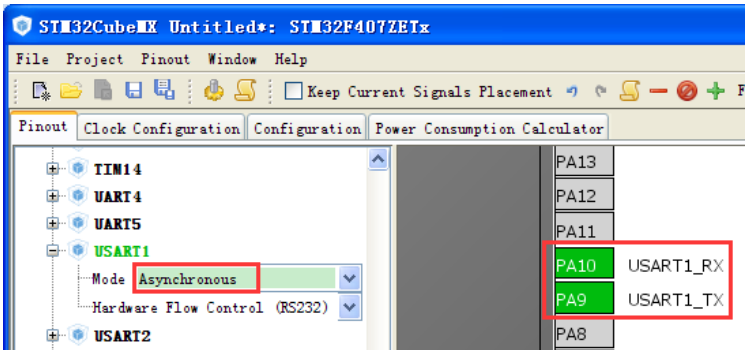
RTC：使用其内部唤醒功能即可。



配置外部 32768Hz 晶振输入，作为 RTC 时钟。

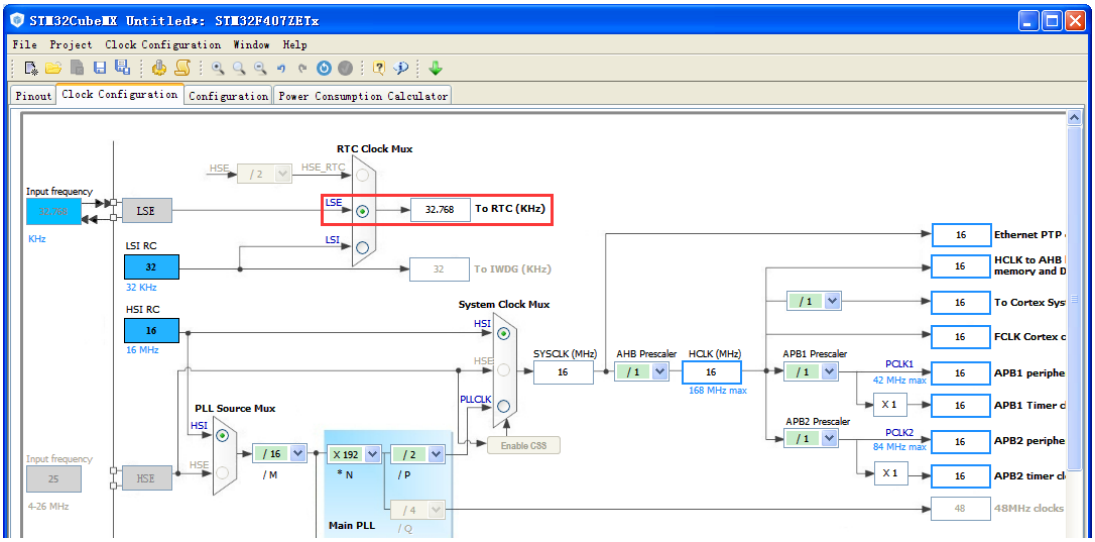


USART：配置串口，作为信息输出接口。



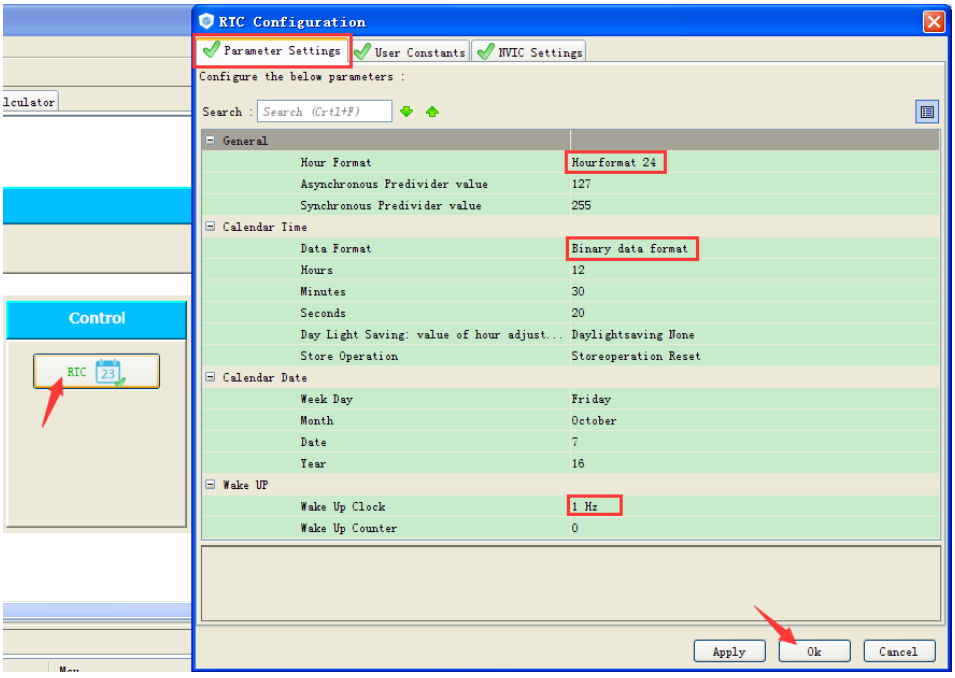
Step3.在 Clock Configuration 界面配置时钟源。

配置时钟树，RTC 使用 32768Hz 外部时钟，其它使用默认值，内部 16MHz。

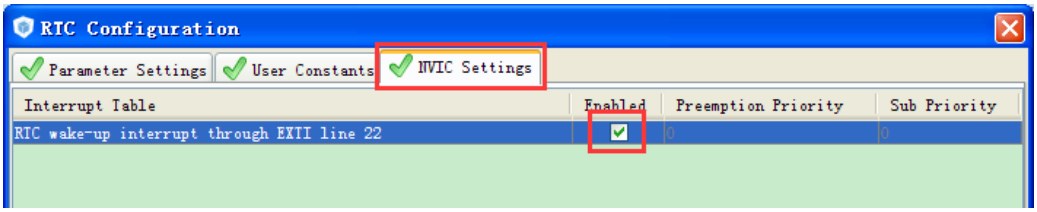


Step4.配置外设参数。

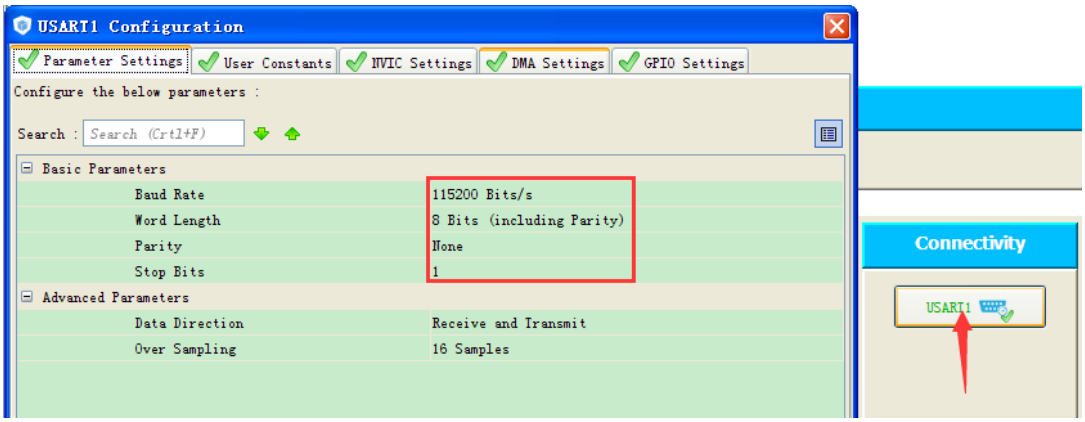
RTC：时间 24 小时格式，数据格式使用二进制，日期 xx16 年 10 月 7 日星期五，时间 12:30:20，唤醒时钟频率 1Hz，其它参数默认。



使能 RTC 唤醒中断。

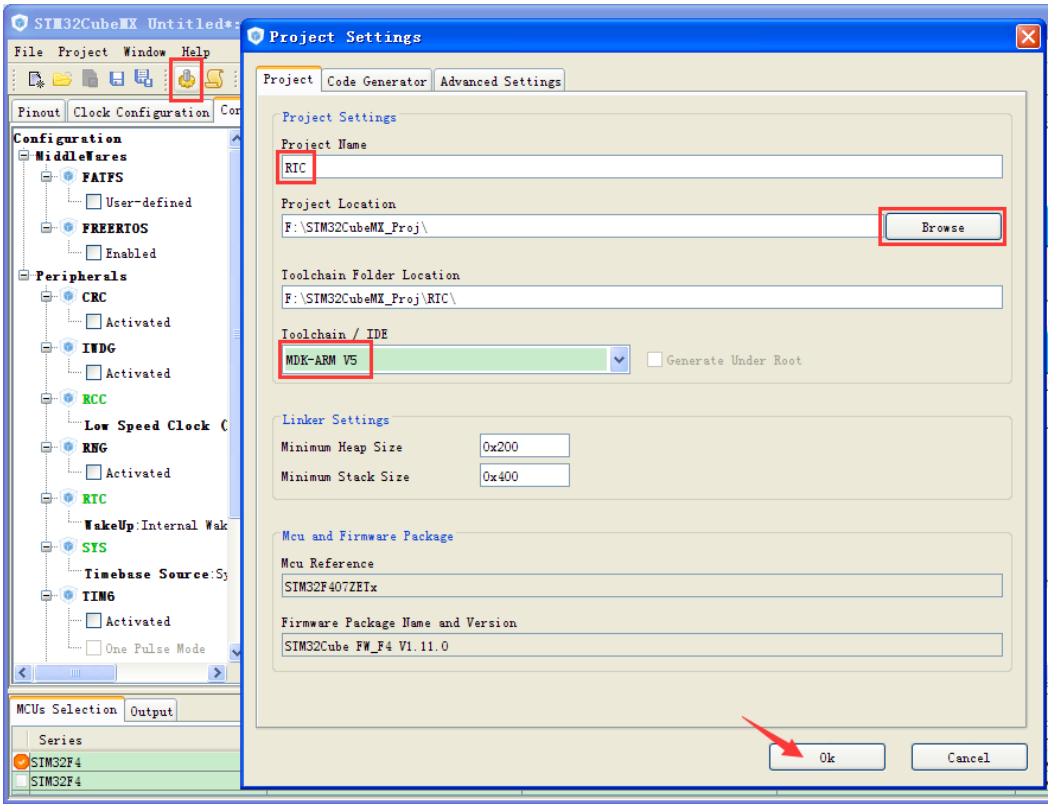


USART：串口参数默认即可。

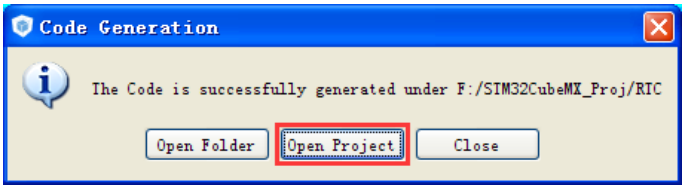


Step5.生成源代码。

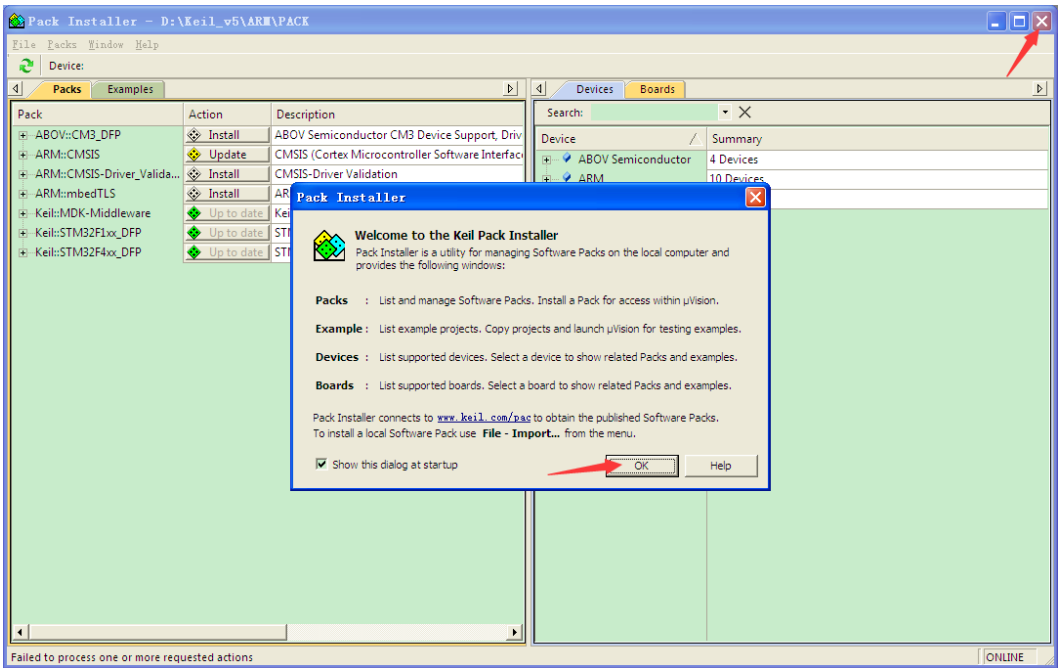
点击生成源代码按钮。在设置界面中输入工程名，保存路径，工程 IDE 类型，点 OK 即可。



生成代码完成后可直接打开工程。

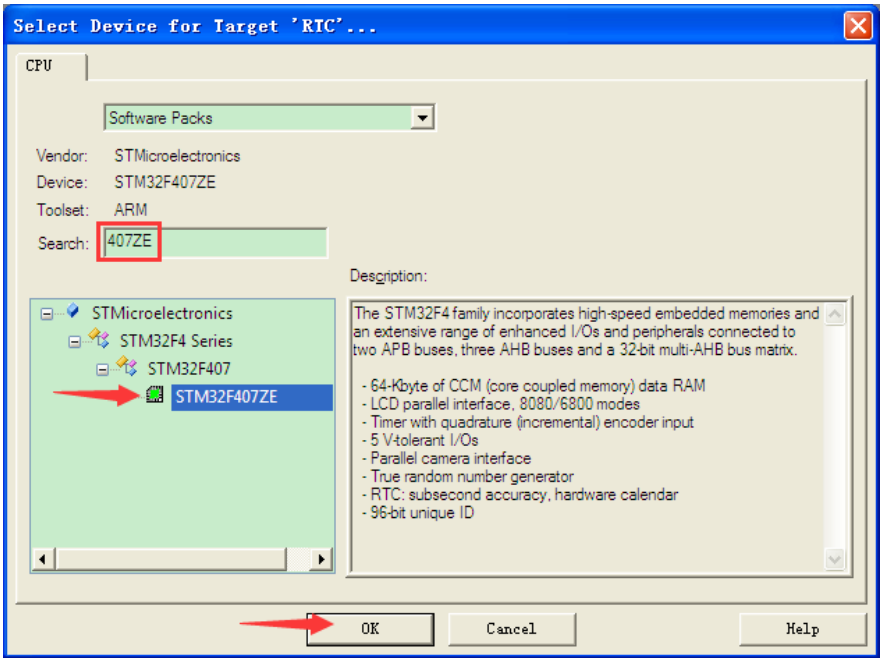


弹出如下对话框时，如果已经安装了 F4 的支持包，则点击 OK 关闭。如果没有安装，则点击界面中的 www.keil.com/...链接，找到芯片的支持包，然后安装。



关闭后面的界面。

点击“是”，然后选择芯片型号。可以在搜索框中输入关键字，加快选择速度。



Step6.添加功能代码。

先在 main.c 文件用户代码区输入包含标准输入输出头文件。

```
33  /* Includes ------------------------------
34  #include "stm32f4xx_hal.h"
35
36  /* USER CODE BEGIN Includes */
37  #include <stdio.h>
38  /* USER CODE END Includes */
39
```

在用户代码区 4 实现标准输出 printf()的底层驱动函数 fputc()，功能是在 UART1 输出一个字符。

```
212  /* USER CODE BEGIN 4 */
213  int fputc(int ch, FILE *f)
214  {
215      HAL_UART_Transmit(&huart1, (uint8_t*)&ch, 1, 10);
216      return ch;
217  }
```

在主函数的用户代码区 1，定义几个变量。

```
68   /* USER CODE BEGIN 1 */
69     RTC_DateTypeDef sDate;
70     RTC_TimeTypeDef sTime;
71     uint8_t second_tmp = 0;
72   /* USER CODE END 1 */
```

在 while(1)中的用户代码区 3，读取时间和日期，并从串口输出。

```
93   while (1)
94   {
95   /* USER CODE END WHILE */
96
97   /* USER CODE BEGIN 3 */
98     HAL_Delay(100);
99     HAL_RTC_GetTime(&hrtc, &sTime, RTC_FORMAT_BIN);// 先读取时间
100    HAL_RTC_GetDate(&hrtc, &sDate, RTC_FORMAT_BIN);// 后读取日期
101    if (second_tmp != sTime.Seconds) { // 判断"秒"是否发生变化
102        second_tmp = sTime.Seconds; // 保存"秒"
103        printf("20%d%d-%d%d-%d%d\r\n",
104                sDate.Year /10%10, sDate.Year %10,
105                sDate.Month/10%10, sDate.Month%10,
106                sDate.Date /10%10, sDate.Date %10);
107        printf(" %d%d:%d%d:%d%d\r\n\r\n",
108                sTime.Hours  /10%10, sTime.Hours  %10,
109                sTime.Minutes/10%10, sTime.Minutes%10,
110                sTime.Seconds/10%10, sTime.Seconds%10);
111    }
112  }
113  /* USER CODE END 3 */
```

至此，工程完成。每秒钟输出一次日期和时间，如下图。

本例程的问题：

本例程只演示了 RTC 的配置，读取过程。例程中，每次上电运行都初始化 RTC 并设置日期和时间，这是不符合实际应用要求的。改善该问题，可参看官方日历例程 RTC_Calendar，可以用 RTC 备份寄存器判断 RTC 是否断过电，然后根据情况确定是否重新初始化 RTC。

本实验过程中遇到的一个问题：

Step4 中已经使能了 RTC 唤醒中断，配置唤醒频率为 1Hz，也添加了回调函数的实现：

```
234  void HAL_RTCEx_WakeUpTimerEventCallback(RTC_HandleTypeDef *hrtc)
235  {
236      RTC_DateTypeDef sDate;
237      RTC_TimeTypeDef sTime;
238
239      HAL_RTC_GetTime(hrtc, &sTime, RTC_FORMAT_BIN);
240      HAL_RTC_GetDate(hrtc, &sDate, RTC_FORMAT_BIN);
241
242      printf("20%2d-%2d-%2d\r\n", sDate.Year, sDate.Month, sDate.Date);
243      printf(" %2d:%2d:%2d\r\n", sTime.Hours, sTime.Minutes, sTime.Seconds);
244  }
245  /* USER CODE END 4 */
```

第一次下载时，正常产生了该中断，并输出了 RTC 信息。但是，之后重新下载程序就不产生该中断了。因为我所用的开发板上带有 RTC 备用电池，所以推测原因可能是 RTC 多次初始化会出问题。实验中，把 RTC 备用电池取下后，复位也不能进该中断，必须保证 RTC 断电后才能正常进入该中断。

要解决该问题，可参看官方日历例程 RTC_Calendar。

官方例程请参考 stm32cubef4.zip 解压后 STM32Cube_FW_F4_V1.9.0\Projects\STM324xG_EVAL\Examples\RTC\RTC_Calendar 目录下的工程。

S.D.Lu 于 深圳
2016 年 10 月