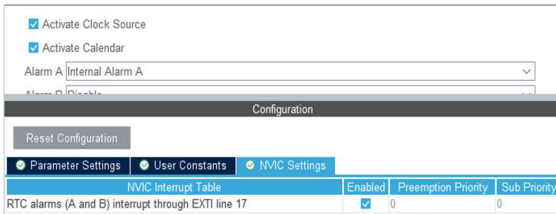
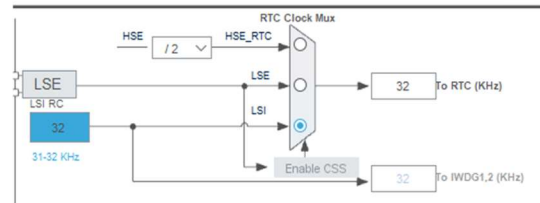


Setting up RTC (Real Time Clock) & One Second Alarm using STM32hxxx and CubeMX

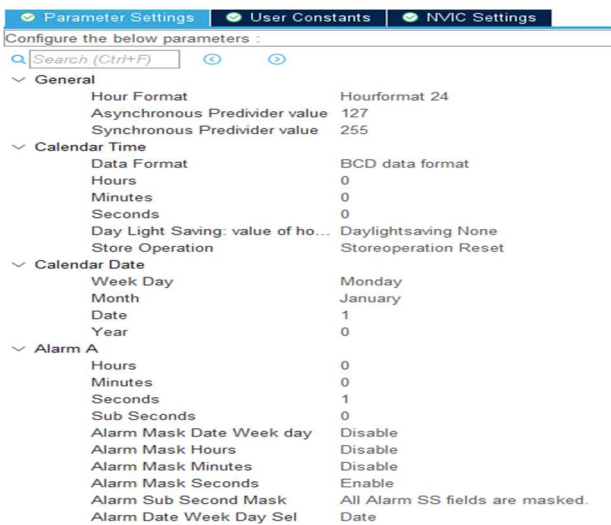
NVIC Setting



Clock Source



Parameter Settings



```
#include "stm32f4xx_hal.h"
#include "stdio.h"
```

```
RTC_HandleTypeDef hrtc;
```

```
int main(void) {
    HAL_Init();
    SystemClock_Config();
    MX_GPIO_Init();
    RTC_Init();
    Set_Date_Time();
    Set_AlarmA();
    while (1) {
        Get_Date_time();
        HAL_Delay(1000);
    }
}
```

```
void Set_Date_Time(void) {
    RTC_TimeTypeDef sTime = {0};
    RTC_DateTypeDef sDate = {0};

    sTime.Hours = 0x0;
    sTime.Minutes = 0x0;
    sTime.Seconds = 0x0;
    sTime.DayLightSaving = RTC_DAYLIGHTSAVING_NONE;
    sTime.StoreOperation = RTC_STOREOPERATION_RESET;
    if (HAL_RTC_SetTime(&hrtc, &sTime, RTC_FORMAT_BIN) != HAL_OK) {
        Error_Handler();
    }
    sDate.WeekDay = RTC_WEEKDAY_MONDAY;
    sDate.Month = RTC_MONTH_JANUARY;
    sDate.Date = 0x1;
    sDate.Year = 0x0;
    if (HAL_RTC_SetDate(&hrtc, &sDate, RTC_FORMAT_BIN) != HAL_OK) {
        Error_Handler();
    }
}
```

```
void Get_Date_Time(void) {
    RTC_TimeTypeDef rtc_time;
    RTC_DateTypeDef rtc_date;
    HAL_RTC_GetTime(&hrtc, &rtc_time, RTC_FORMAT_BIN);
    HAL_RTC_GetDate(&hrtc, &rtc_date, RTC_FORMAT_BIN);
}
```

```
void Set_AlarmA(void) {
    RTC_AlarmTypeDef sAlarm = {0};

    sAlarm.AlarmTime.Hours = 0x0;
    sAlarm.AlarmTime.Minutes = 0x0;
    sAlarm.AlarmTime.Seconds = 0x1;
    sAlarm.AlarmTime.SubSeconds = 0x0;
    sAlarm.AlarmTime.DayLightSaving = RTC_DAYLIGHTSAVING_NONE;
    sAlarm.AlarmTime.StoreOperation = RTC_STOREOPERATION_RESET;
    sAlarm.AlarmMask = RTC_ALARMMASK_ALL;
    sAlarm.AlarmSubSecondMask = RTC_ALARMSUBSECONDMASK_ALL;
    sAlarm.AlarmDateWeekDaySel = RTC_ALARMDATEWEEKDAYSEL_DATE;
    sAlarm.AlarmDateWeekDay = 0x1; //Put Current Date to Trigger it
    sAlarm.Alarm = RTC_ALARM_A;
    if (HAL_RTC_SetAlarm_IT(&hrtc, &sAlarm, RTC_FORMAT_BIN) != HAL_OK) {
        Error_Handler();
    }
}
```

```
void RTC_Init(void) {
    hrtc.Instance = RTC;
    hrtc.Init.HourFormat = RTC_HOURFORMAT_24;
    hrtc.Init.AsynchPrediv = 127;
    hrtc.Init.SynchPrediv = 255;
    hrtc.Init.OutPut = RTC_OUTPUT_DISABLE;
    hrtc.Init.OutPutPolarity = RTC_OUTPUT_POLARITY_HIGH;
    hrtc.Init.OutPutType = RTC_OUTPUT_TYPE_OPENDRAIN;
    hrtc.Init.OutPutRemap = RTC_OUTPUT_REMAP_NONE;
    if (HAL_RTC_Init(&hrtc) != HAL_OK) {
        Error_Handler();
    }
}
```