

# **EEL7030 - Microprocessadores**



**LCS**

Laboratório de  
Comunicações  
e Sistemas  
Embarcados

**Prof. Raimes Moraes**

**EEL - UFSC**

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

### ❑ Objetivo:

- Configurar RTC no STM32CubeMX
- Gerar código e utilizar funções HAL
- Inserir biblioteca do LCD no projeto (ver material adicional)
- Mostrar horário no LCD

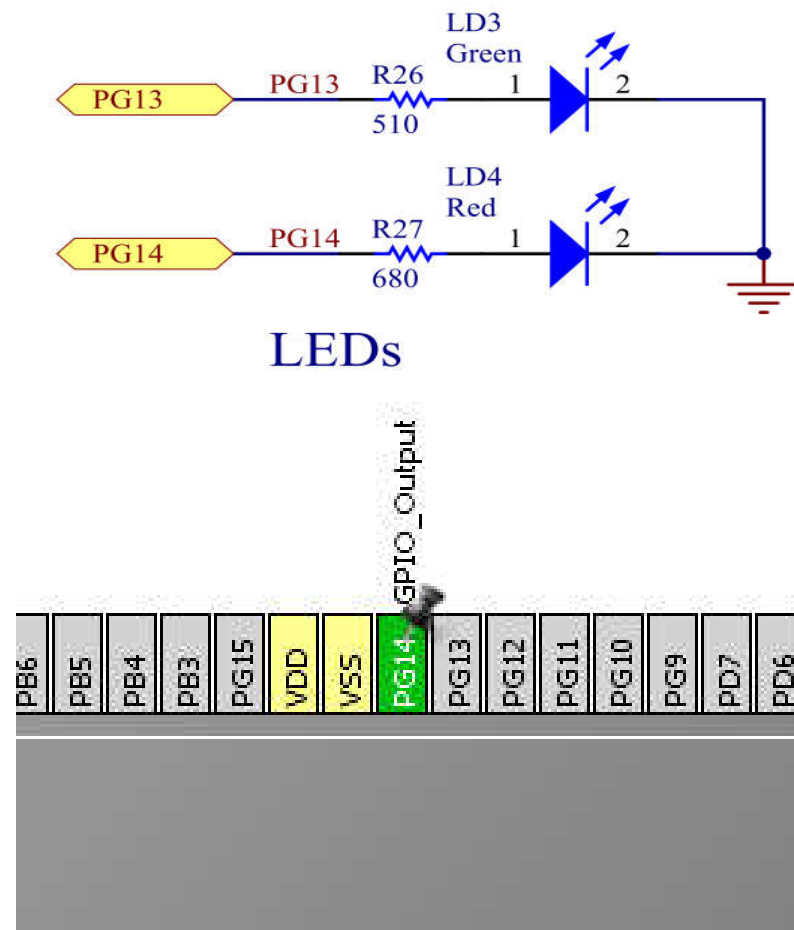
### ❑ Crie projeto no STMCubeMX:

#### 1. *New Project*

2. No MCU Selector: STM32F4 (series); STM32F429/439 (lines ); LQFP144 (package); selecione: STM32F429ZITx - **OBS: Selecionar crystal em RCC e configurar clocks; Selecionar SWD para Debug em SYS**

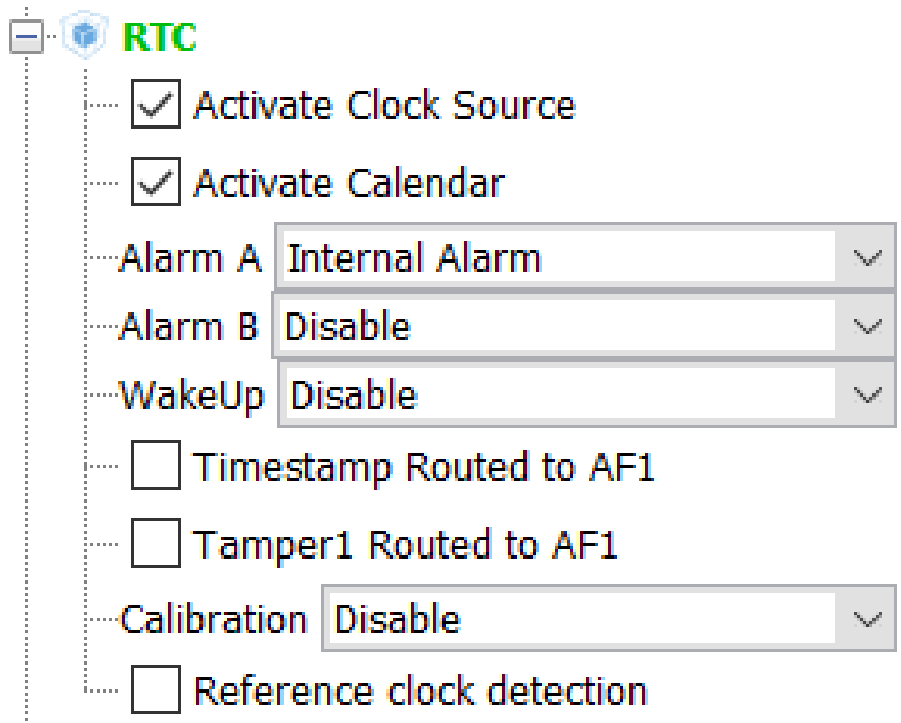
## Exemplo 7 – Utilização do RTC (Real-Time Clock)

3. Clique no pino PG14 e selecione GPIO\_Output para acender led



## Exemplo 7 – Utilização do RTC (Real-Time Clock)

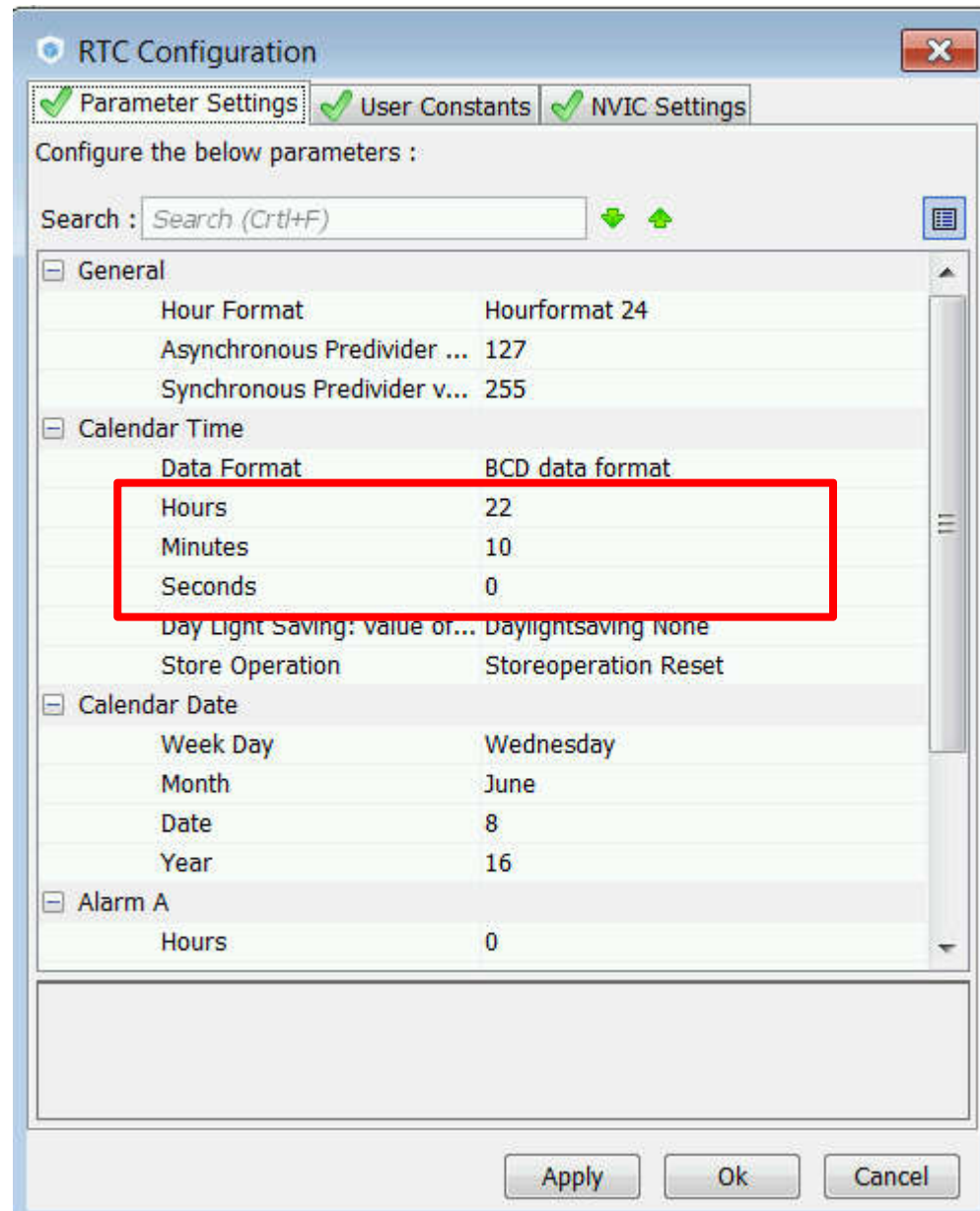
4. Selecione RTC e selecione *Internal Alarm (A)* (figura à esquerda)
5. Configure o RTC. Para tal, selecione *Tab Configuration->Control-> RTC* (figura à direita)



## Exemplo 7 – Utilização do RTC (Real-Time Clock)

### 6. Configure horários no RTC:

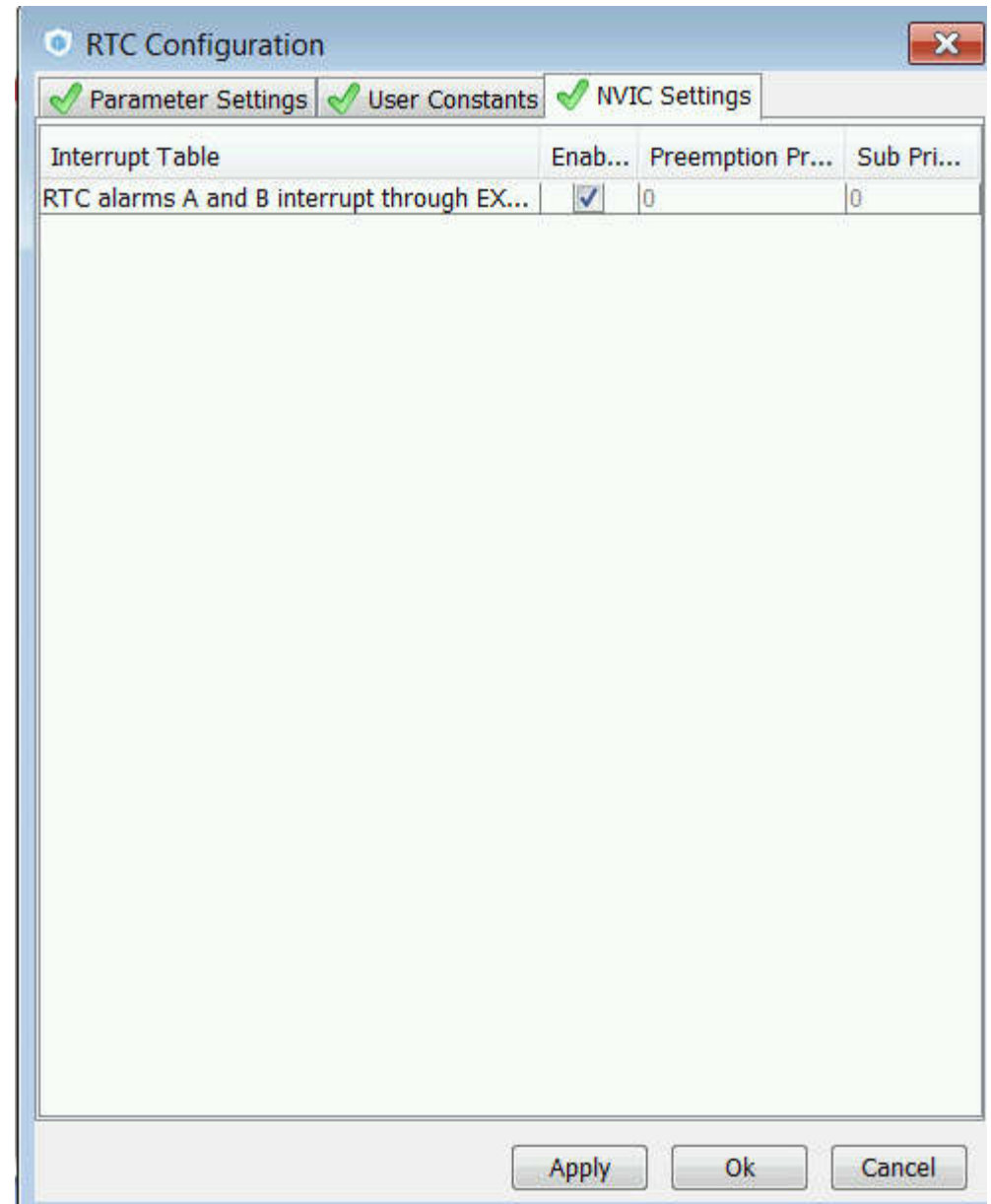
- *Calendar Time*
- *Hours*
- *Minutes*
- *Seconds*



## Exemplo 7 – Utilização do RTC (Real-Time Clock)

### 7. Habilitar interrupção:

- Tab-> *NVIC Settings*
- Habilitar *Alarm Interrupt*
- Clique em Ok



## Exemplo 7 – Utilização do RTC (Real-Time Clock)

8. Forneça dados para geração do código:

- Menu -> Project -> Project Settings
- Digite *Project name*
- Digite *Project location*
- Escolha *toolchain* (MDK-ARM V5)
- Clique em Ok

9. Gere template para código:

- Menu -> Project -> Generate Code
- *Open Project*

The screenshot shows the 'Project Settings' dialog box with the 'Code Generator' tab selected. The 'Project Settings' section contains the following fields:

- Project Name:** First
- Project Location:** C:\Users\Raines\STM\Exemplos\ (with a 'Browse' button)
- Toolchain Folder Location:** C:\Users\Raines\STM\Exemplos\First\
- Toolchain / IDE:** MDK-ARM V5 (dropdown menu)

The 'Mcu and Firmware Package' section contains the following fields:

- Mcu Reference:** STM32F429ZITx
- Firmware Package Name and Version:** STM32Cube FW\_F4 V1.9.0

At the bottom right, there are 'Ok' and 'Cancel' buttons.

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

### PARTE I

10. Insira bibliotecas necessárias para a escrita no LCD  
(**utilizar Tutorial LCD disponível no Moodle**)



## Exemplo 7 – Utilização do RTC (Real-Time Clock)

❑ Após energização, registradores do RTC são protegidos contra modificação indesejada.

❑ Para permitir sua alteração, inserir antes de MX\_RTC\_Init():

11.        **HAL\_PWR\_EnableBkUpAccess()**

❑ Habilitar RCC:

12.        **\_\_HAL\_RCC\_RTC\_ENABLE()**

❑ Crie variáveis para leitura do horário e criação de vetor com horas em ascii para escrita em LCD:

```
13.        RTC_TimeTypeDef myTime;  
           RTC_DateTypeDef myDate;  
           uint8_t horacerta[16];  
           uint8_t state=1;                // usada na parte II
```

❑ Estes e os demais passos para configurar o LCD e proceder a escrita de mensagens,, bem como horário, encontra-se detalhados nos próximos slides

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

### ❑ Solução (parte 1):

- Insira nos includes, a declaração ao lado :

```
/* USER CODE BEGIN Includes */  
#include "stm32f429i_discovery_lcd.h"  
/* USER CODE END Includes */
```

- Antes da chamada da função MX\_GPIO\_Init(); inserir as funções do quadro abaixo:

```
/* Initialize all configured peripherals */  
HAL_PWR_EnableBkUpAccess();//enable PWR backup domain access (RTC,BKReg)  
__HAL_RCC_RTC_ENABLE();//Enable RTC. not created by cube because the RTC can run.
```

- Crie variáveis para leitura do horário e criação de vetor com horas em ascii para escrita em LCD:

```
/* USER CODE BEGIN 0 */
```

```
RTC_TimeTypeDef myTime;  
RTC_DateTypeDef myDate;  
uint8_t horacerta[16];  
uint8_t state=1;
```

```
/* USER CODE END 0 */
```

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

❑ **Solução (parte 2):** Insira funções abaixo para configurar o LCD

```
/* USER CODE BEGIN 2 */
```

```
BSP_LCD_Init();                //init LCD
//set the layer buffer address into SDRAM
BSP_LCD_LayerDefaultInit(1, SDRAM_DEVICE_ADDR);
BSP_LCD_SelectLayer(1);        //select on which layer we write
BSP_LCD_DisplayOn();           //turn on LCD
BSP_LCD_Clear(LCD_COLOR_BLUE); //clear the LCD on blue color
BSP_LCD_SetBackColor(LCD_COLOR_BLUE); //set text background color
BSP_LCD_SetTextColor(LCD_COLOR_WHITE); //set text color
```

```
//write text
BSP_LCD_DisplayStringAtLine(2," Cube STM32 ");
BSP_LCD_DisplayStringAtLine(3," Floripa ");
BSP_LCD_DisplayStringAtLine(5," EEL7030 ");
```

```
/* USER CODE END 2 */
```

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

- ❑ **Solução (parte 3):** Insira o código do **if** abaixo dentro do while da função main

```
/* Infinite loop para frequente leitura e escrita do horário lido no LCD
/* USER CODE BEGIN WHILE */
while (1)
{
    if (state) {
        HAL_RTC_GetTime(&hrtc, &myTime, FORMAT_BIN);
        HAL_RTC_GetDate(&hrtc, &myDate, FORMAT_BIN);
        sprintf((char *) horacerta, "Hora: %2d:%2d:%2d\n", myTime.Hours,
myTime.Minutes, myTime.Seconds);
        BSP_LCD_DisplayStringAtLine(4,horacerta);
        HAL_Delay(500);
        //state=0;
    } //end of if
/* USER CODE END WHILE */
} // end of while
```

- ❑ **Compile o programa e grave-o no kit, reset o kit (botão preto) e verifique o funcionamento**

## **Exemplo 7 – Utilização do RTC (Real-Time Clock)**

### **PARTE II**

**Utilizar RTC com interrupção**

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

### ❑ Solução (parte 1):

- Dentro de while da função main, comente e remova comentário das 2 linhas de código, como abaixo:

```
//HAL_Delay(500);  
state=0;
```

- Insira a função abaixo para atender ao alarme:

```
/* USER CODE BEGIN 4 */  
void HAL_RTC_AlarmAEventCallback(RTC_HandleTypeDef *hrtc){  
  
    RTC_AlarmTypeDef sAlarm;  
    HAL_RTC_GetAlarm(hrtc,&sAlarm,RTC_ALARM_A,FORMAT_BIN);  
    if(sAlarm.AlarmTime.Seconds>58){  
        sAlarm.AlarmTime.Seconds=0;  
    }else{  
  
        // Ajuste do tempo do alarme em 1s a mais do horário inicial:  
        sAlarm.AlarmTime.Seconds=sAlarm.AlarmTime.Seconds+1;  
    }  
    while(HAL_RTC_SetAlarm_IT(hrtc, &sAlarm, FORMAT_BIN)!=HAL_OK){}  
    HAL_GPIO_TogglePin(GPIOD,GPIO_PIN_14);  
    state=1;  
  
}
```

## Exemplo 7 – Utilização do RTC (Real-Time Clock)

- ❑ **Solução (parte 2):** Dentro da função `MX_RTC_Init`, modifique as 2 linhas de código como abaixo:

```
sAlarm.AlarmTime.Seconds = 1; // Habilita comparação de Segundos
```

```
sAlarm.AlarmMask =  
RTC_ALARMMASK_DATEWEEKDAY|RTC_ALARMMASK_HOURS|RTC_ALARMMASK_MINUTES;
```

- ❑ **Deve ficar como abaixo:**

```
/* Em MX_RTC_Init, habilitar o Alarm A - observar linhas em negrito */  
/**Enable the Alarm A      */
```

```
sAlarm.AlarmTime.Hours = 0;  
sAlarm.AlarmTime.Minutes = 0;  
sAlarm.AlarmTime.Seconds = 1;           // Habilita comparação de Segundos  
sAlarm.AlarmTime.SubSeconds = 0;  
sAlarm.AlarmTime.TimeFormat = RTC_HOURFORMAT12_AM;  
sAlarm.AlarmTime.DayLightSaving = RTC_DAYLIGHTSAVING_NONE;  
sAlarm.AlarmTime.StoreOperation = RTC_STOREOPERATION_RESET;  
sAlarm.AlarmMask = RTC_ALARMMASK_DATEWEEKDAY|RTC_ALARMMASK_HOURS|RTC_ALARMMASK_MINUTES;  
sAlarm.AlarmSubSecondMask = RTC_ALARMSUBSECONDMASK_ALL;  
sAlarm.AlarmDateWeekDaySel = RTC_ALARMDATEWEEKDAYSEL_DATE;  
sAlarm.AlarmDateWeekDay = 1;  
sAlarm.Alarm = RTC_ALARM_A;  
HAL_RTC_SetAlarm_IT(&hrtc, &sAlarm, FORMAT_BCD);
```

## **Exemplo 7 – Utilização do RTC (Real-Time Clock)**

- ☐ **Compile o programa e grave-o no kit, reset o kit (botão preto) e verifique o funcionamento**