

EEL7030 - Microprocessadores



LCS

Laboratório de
Comunicações
e Sistemas
Embarcados

Prof. Raimes Moraes

EEL - UFSC

Exemplo 4 - Configurar **TIMER** como contador

☐ **Objetivo:**

- Configurar TIMER no STM32CubeMX como contador
- Gerar código e utilizar funções HAL
- Acender/apagar led a cada contagem de 5 pulsos

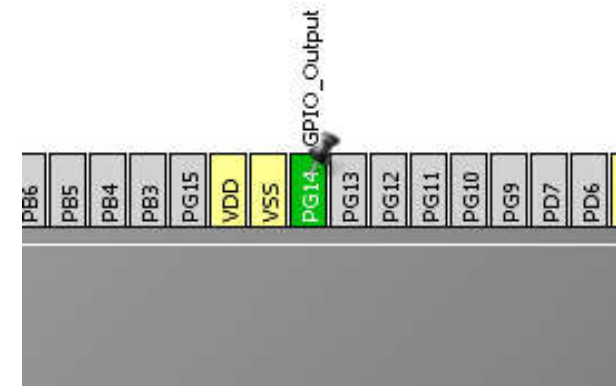
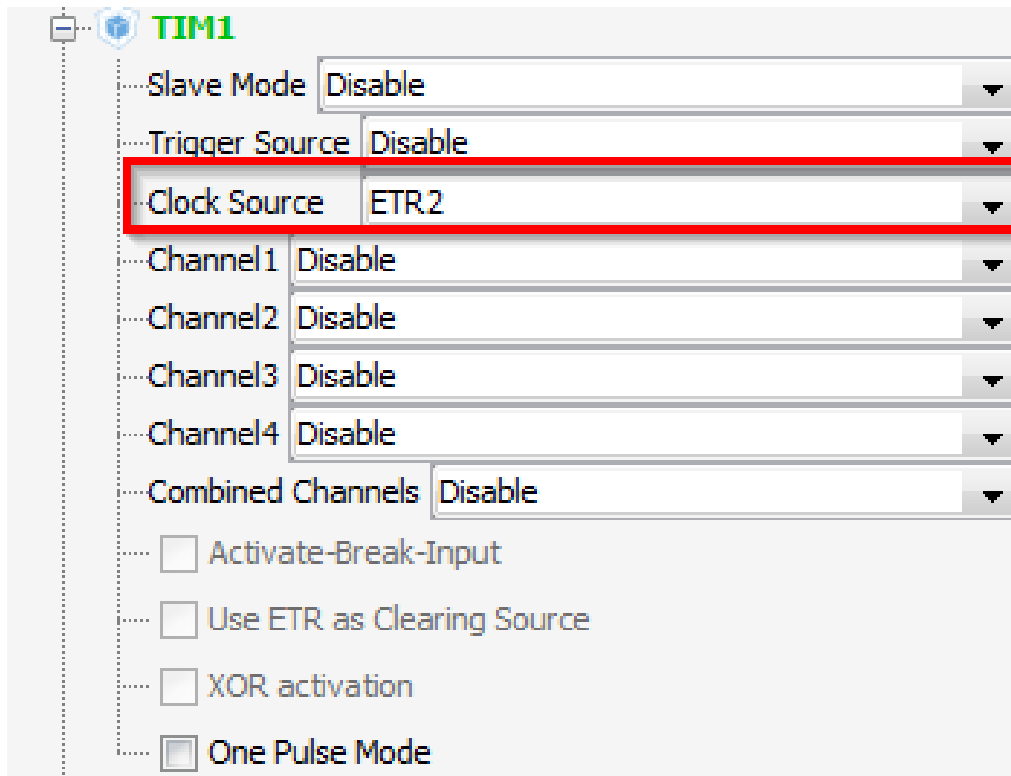
☐ Crie projeto no STMCubeMX:

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**

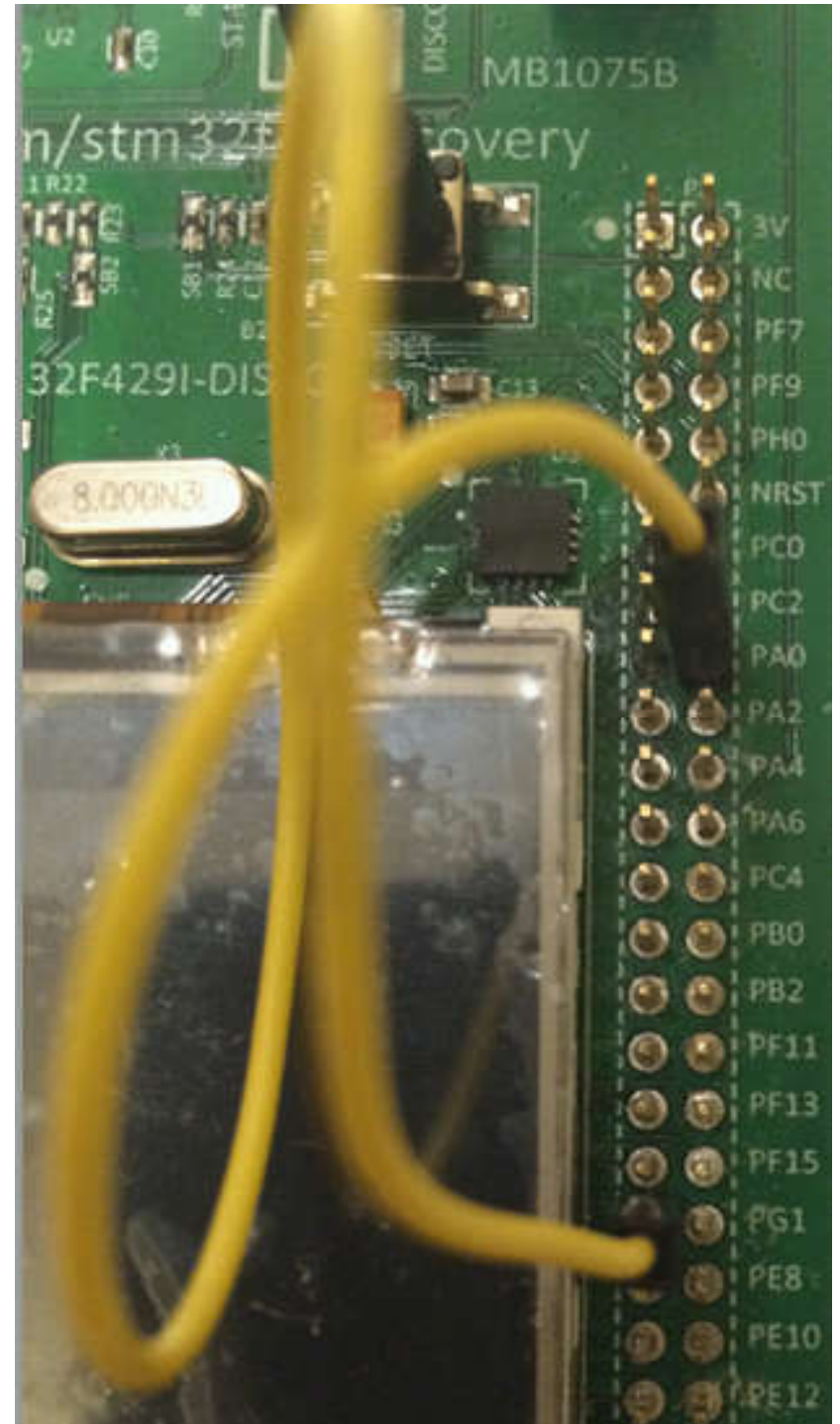
Exemplo 4 - Configurar **TIMER** como contador

3. Selecione TIM1 e especifique fonte de clock: ETR2
4. Clique no pino PG14 na figura do chip e selecione GPIO_Output (LED)



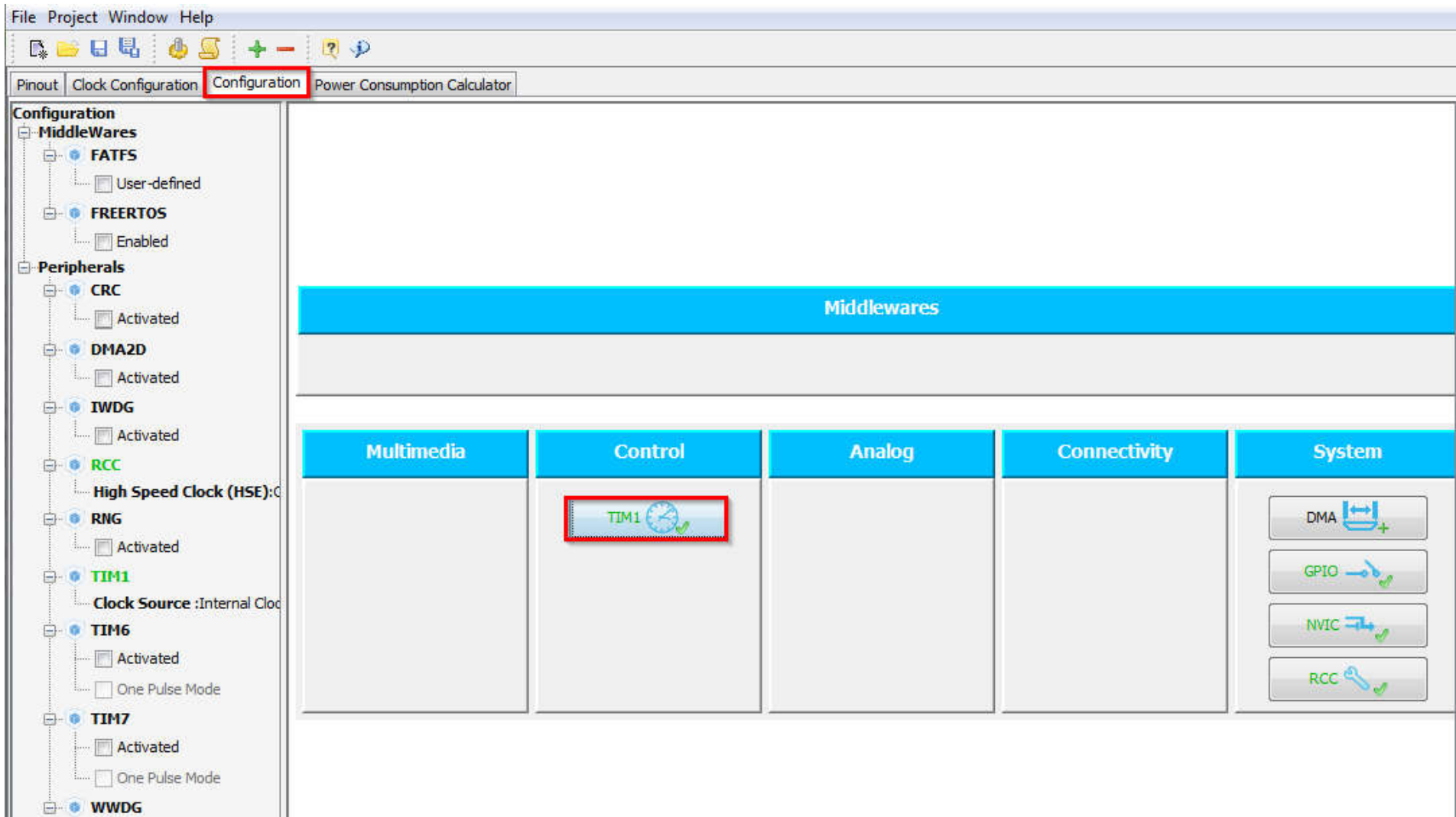
Exemplo 4 - Configurar **TIMER** como contador

5. Conecte pino PA0 (*Push Button*) ao pino PE7 (ETR)



Exemplo 4 - Configurar **TIMER** como contador

6. Configure o TIMER1 => Tab->Configuration->Control->TIM1



Exemplo 4 - Configurar **TIMER** como contador

7. Configurar registradores para contagem de 5 clocks:

- Tab->*Parameter Settings*
- *Counter period: 5*
- Programe filtragem adequada para rejeição de *bouncing* no campo *clock*

TIM1 Configuration

Parameter Settings NVIC Settings DMA Settings GPIO Settings

Configure the below parameters :

Counter Settings

Prescaler (PSC - 16 bits value)	0
Counter Mode	Up
Counter Period (AutoReload Register - 16 bits value)	5
Internal Clock Division (CKD)	No Division
Repetition Counter (RCR - 8 bits value)	0

Trigger Output (TRGO) Parameters

Master/Slave Mode	Disable (no sync between this TIM (Master) and its Slaves)
Trigger Event Selection	Reset (UG bit from TIMx_EGR)

Clock

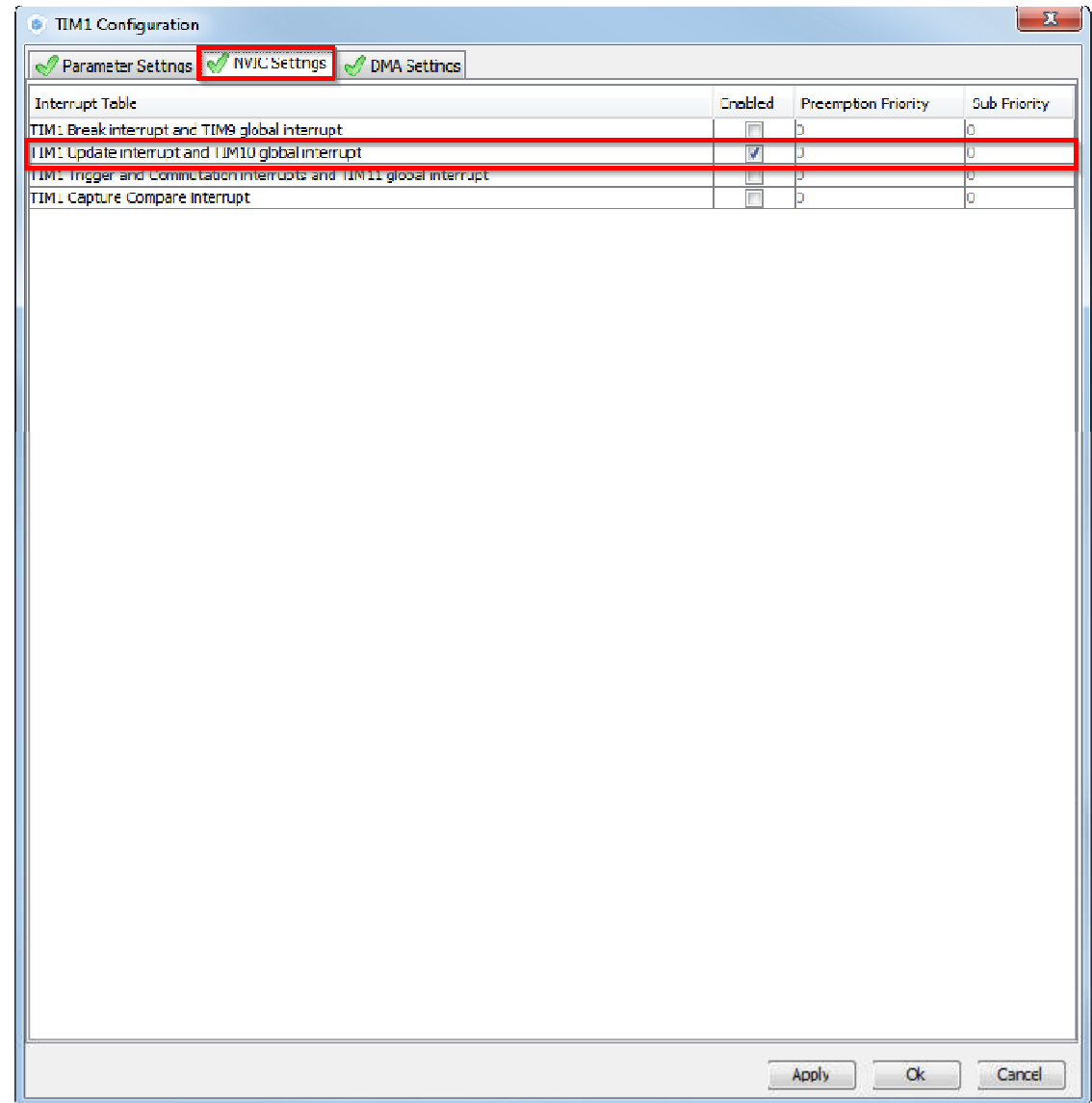
Clock Filter (4 bits value)	0
Clock Polarity	non inverted
Clock Prescaler	Prescaler not used

Apply Ok Cancel

Exemplo 4 - Configurar **TIMER** como contador

8. Configurar registradores para gerar interrupção:

- Tab-> *NVIC Settings*
- Enable TIM1 Update interrupt
- Clique em Ok



Exemplo 4 - Configurar **TIMER** como contador

9. Forneça dados para geração do código:

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

10. Gere template para código:

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

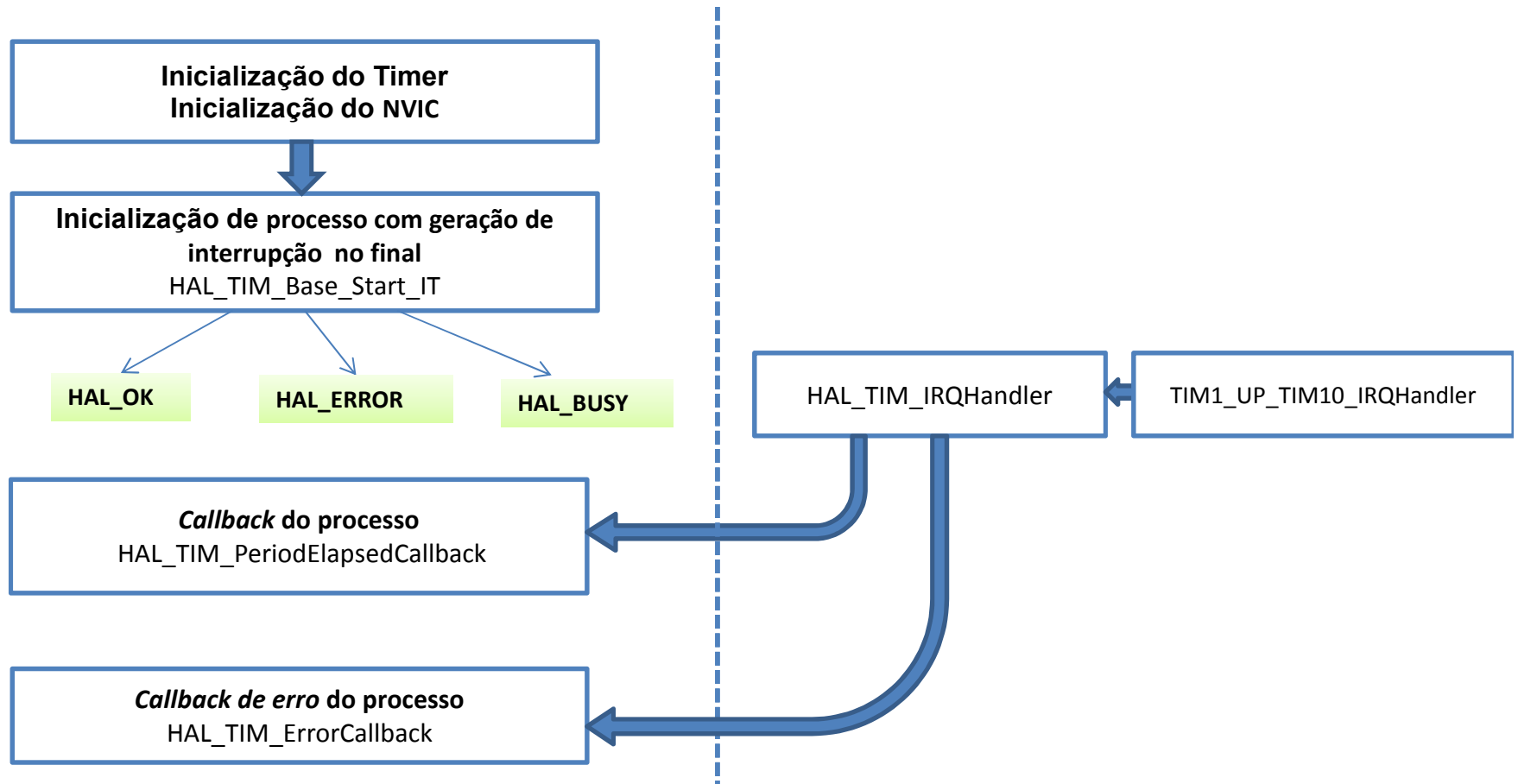
The screenshot shows the 'Project Settings' dialog box with the 'Project' tab active. The settings are as follows:

Field	Value
Project Name	First
Project Location	C:\Users\Raines\STM\Exemplos\
Toolchain Folder Location	C:\Users\Raines\STM\Exemplos\First\
Toolchain / IDE	MDK-ARM V5
Mcu Reference	STM32F429ZITx
Firmware Package Name and Version	STM32Cube FW_F4 V1.9.0

Buttons: 'Browse' (next to Project Location), 'Ok', and 'Cancel'.

Exemplo 4 - Configurar **TIMER** como contador

HAL Library TIM com interrupção



Exemplo 4 - Configurar **TIMER** como contador

11. Abra o projeto no Keil

12. Deve-se inserir funções em **main.c** (em **Application User**) entre os rótulos
/ USER CODE BEGIN 2 */* e */* USER CODE END 2 */*

- Para a inicialização do Timer, utilize a função:

```
HAL_TIM_Base_Start_IT(&htim1)
```

- Para atender a interrupção, utilize a função:

```
HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
```

- Para piscar o led, utilize a função:

```
HAL_GPIO_TogglePin(GPIO_TypeDef* GPIOx, uint16_t GPIO_Pin)
```

Exemplo 4 - Configurar **TIMER** como contador

❑ Solução

- Inicialização do Timer:

```
/* USER CODE BEGIN 2 */  
HAL_TIM_Base_Start_IT(&htim1);  
/* USER CODE END 2 */
```

- Atendimento de interrupção:

```
/* USER CODE BEGIN 4 */  
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)  
{  
    HAL_GPIO_TogglePin(GPIOD, GPIO_PIN_14);  
}  
/* USER CODE END 4 */
```