

EEL7030 - Microprocessadores



Prof. Raimes Moraes

EEL - UFSC

Exemplo 1- Configurar **GPIO** para piscar LED

❑ Objetivo

- Mostrar uso de pino de entrada e saída no STM32CubeMX
- Mostrar geração de código no CubeMX e uso das funções HAL

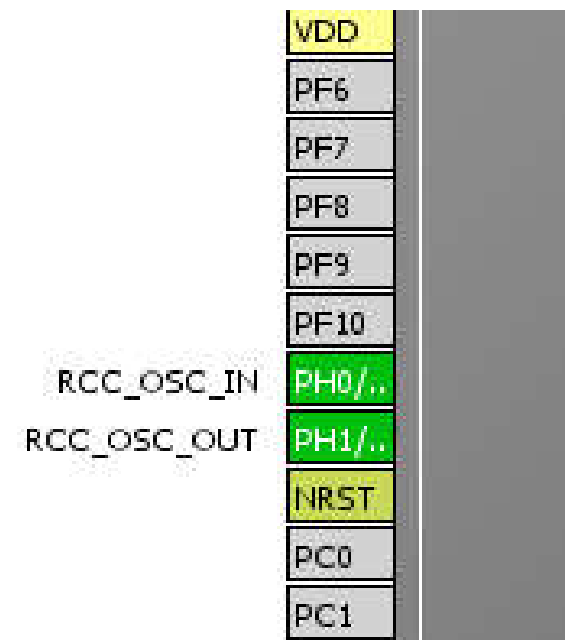
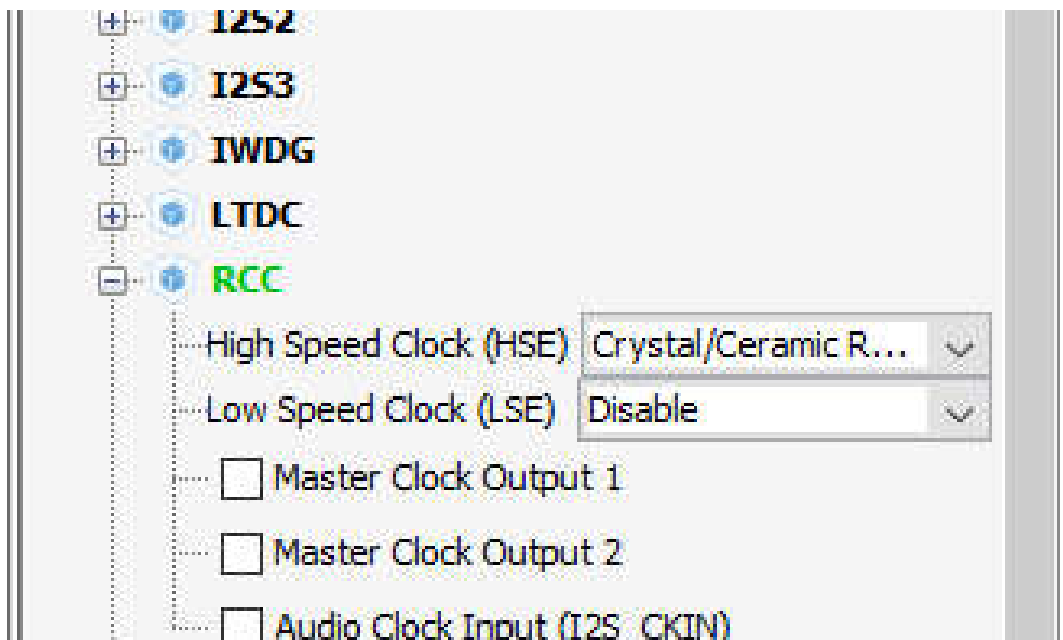
❑ Crie projeto no STMCubeMX:

1. New Project
2. No MCU Selector: STM32F4 (series); STM32F429/439 (lines); LQFP144 (package); selecione: STM32F429ZITx

Exemplo 1- Configurar GPIO para piscar LED

❑ Configuração do clock dos diferentes barramentos e periféricos:

3. Selecionar RCC (*Reset and Clock Control*) e especificar *Crystal* como fonte de clock HSE (*High Speed External oscillators*)

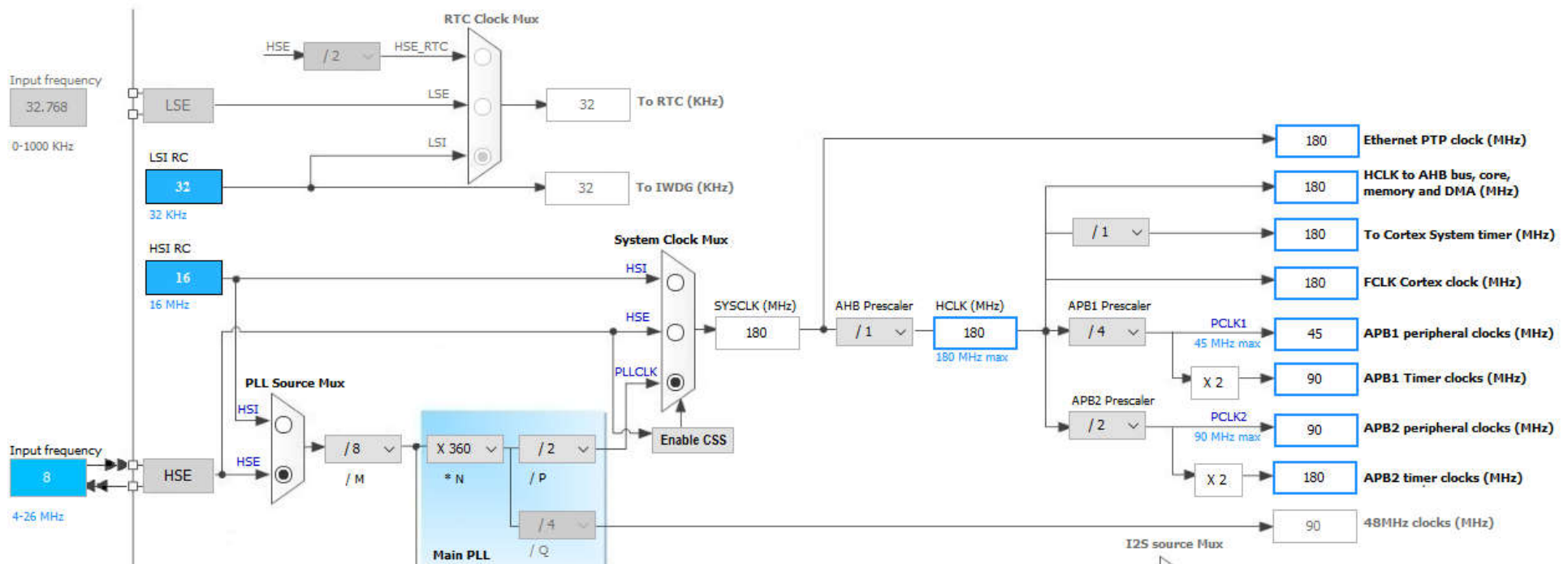


Exemplo 1- Configurar GPIO para piscar LED

❑ Configuração do clock dos diferentes barramentos e periféricos:

4. TAB -> Clock Configuration

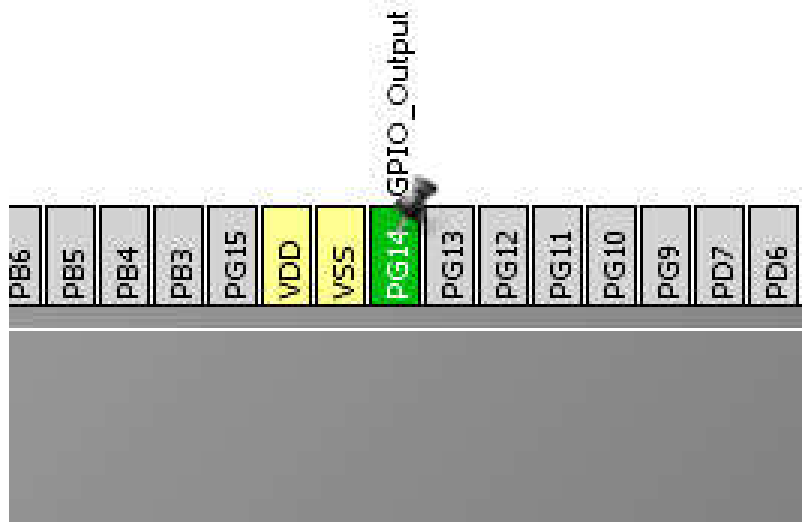
5. Substituir cristal de 25 MHz por 8 MHz e seguir esquema abaixo:



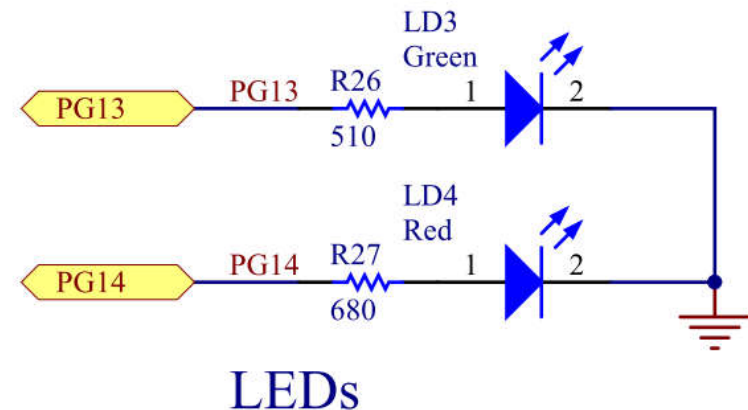
Exemplo 1- Configurar **GPIO** para piscar LED

- ❑ Configurar pino ao qual o LED está conectado (PG14) como pino de saída: **GPIO_Output**

6. Em TAB -> Pinout, clicar no pino e seleccionar a opção como abaixo:



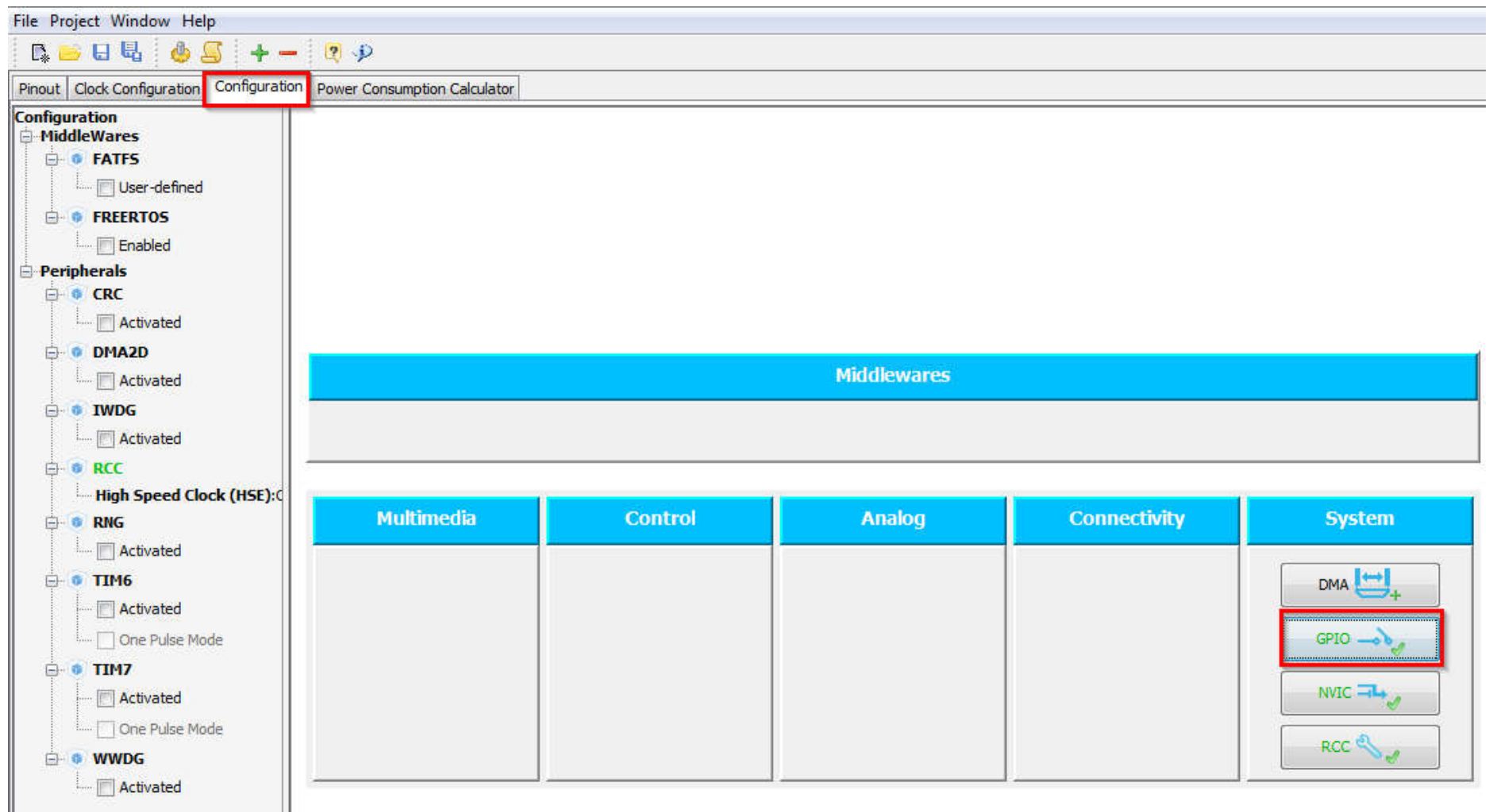
Conexão dos leds aos pinos



Exemplo 1- Configurar GPIO para piscar LED

- ❑ Configurar pino de saída:

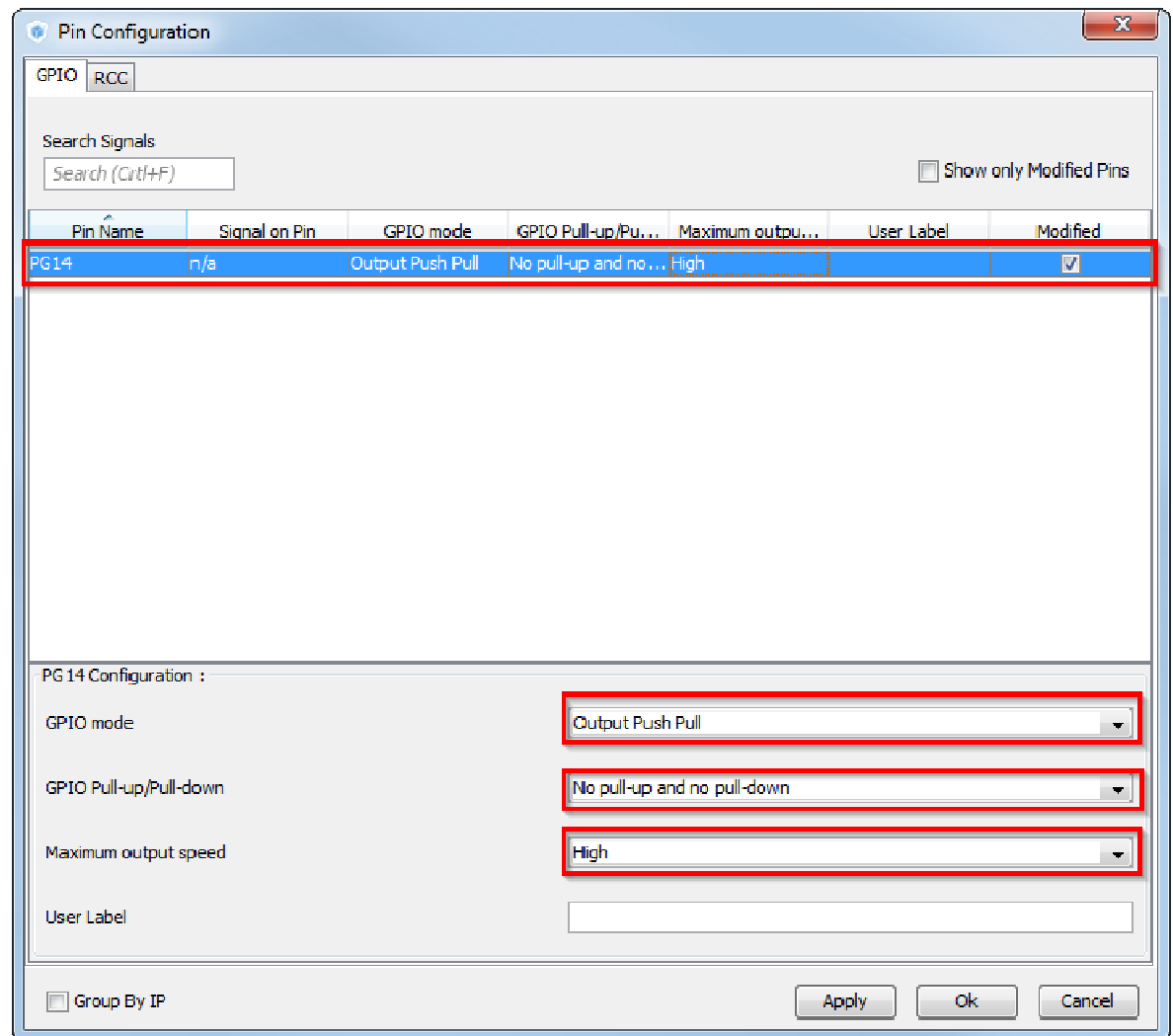
7. TAB->Configuration->System->GPIO



Exemplo 1- Configurar GPIO para piscar LED

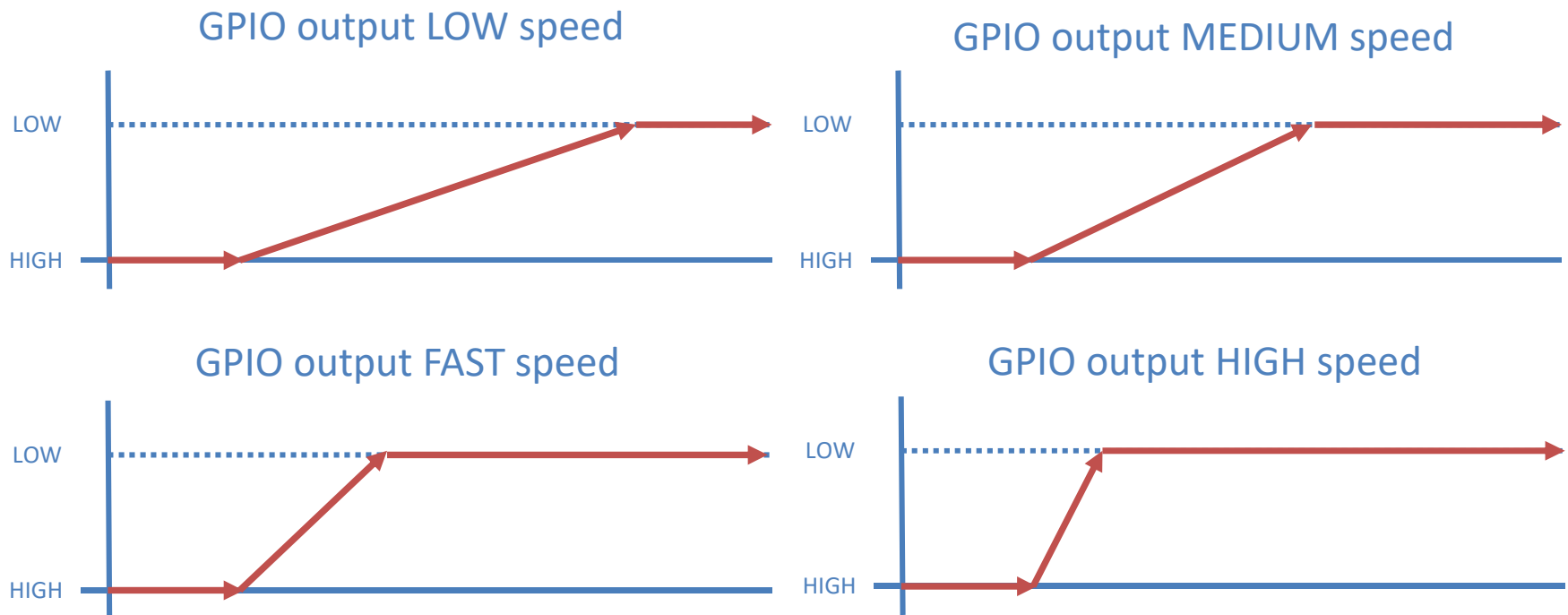
8. Configurar pino de saída:

- GPIO mode: Push Pull
- No pull-up and pull-down
- Output speed: HIGH
- Clique OK



Exemplo 1- Configurar GPIO para piscar LED

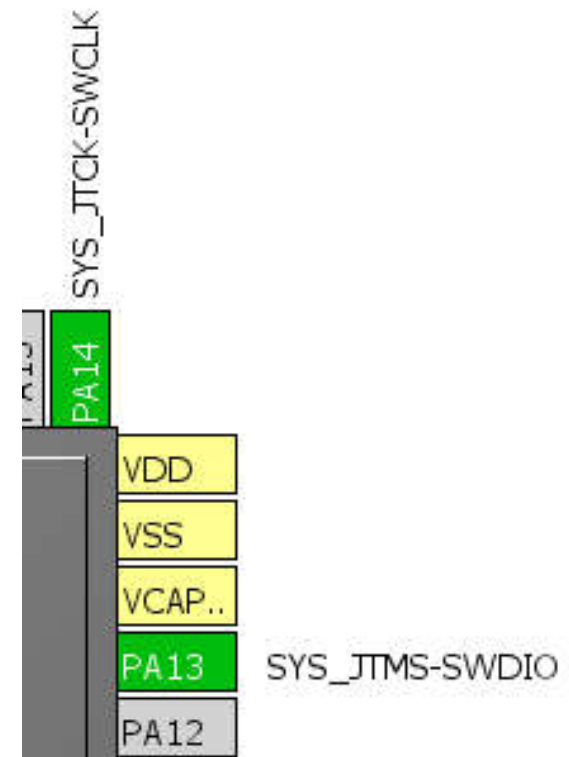
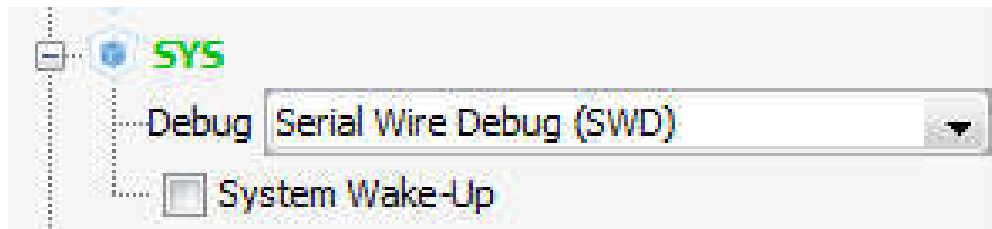
- ❑ Configuração da velocidade (*Output speed*) de alteração do estado do pino de saída (*rising e falling*)
 - **Maior velocidade => maior ruído EMI e maior consumo do STM32;**
 - Deve-se ajustar velocidade em função do periférico (para SPI a 45MHz, velocidade deve ser alta)



Exemplo 1- Configurar **GPIO** para piscar **LED**

9. Para permitir depuração, configure pinos relacionados ao SWD (única opção no kit):

- Configuração a ser realizada, usando: TAB->Pinout->SYS
- Escolher **SWD**;



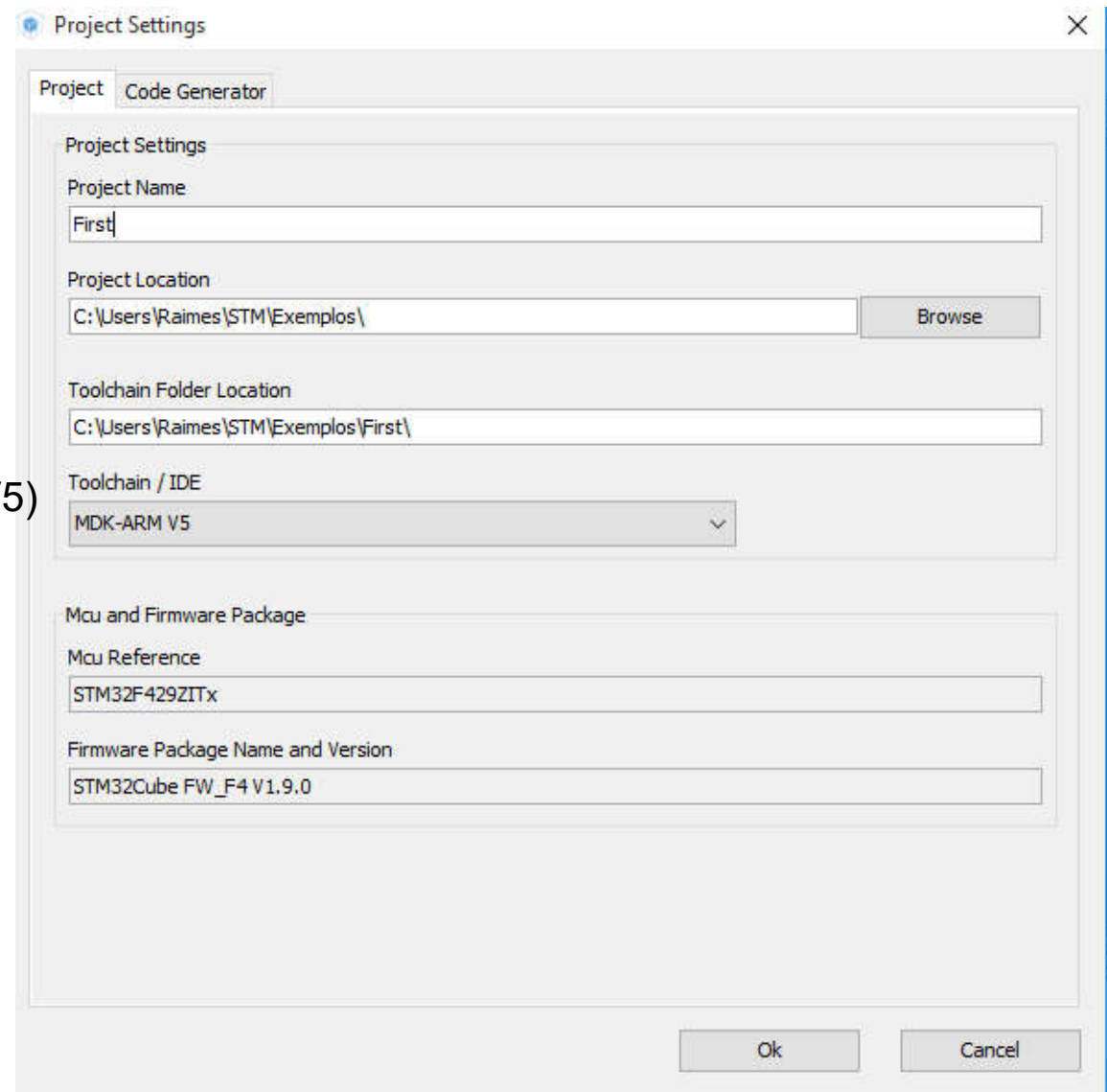
Exemplo 1- Configurar **GPIO** para piscar LED

10. Forneça dados para geração do código:

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

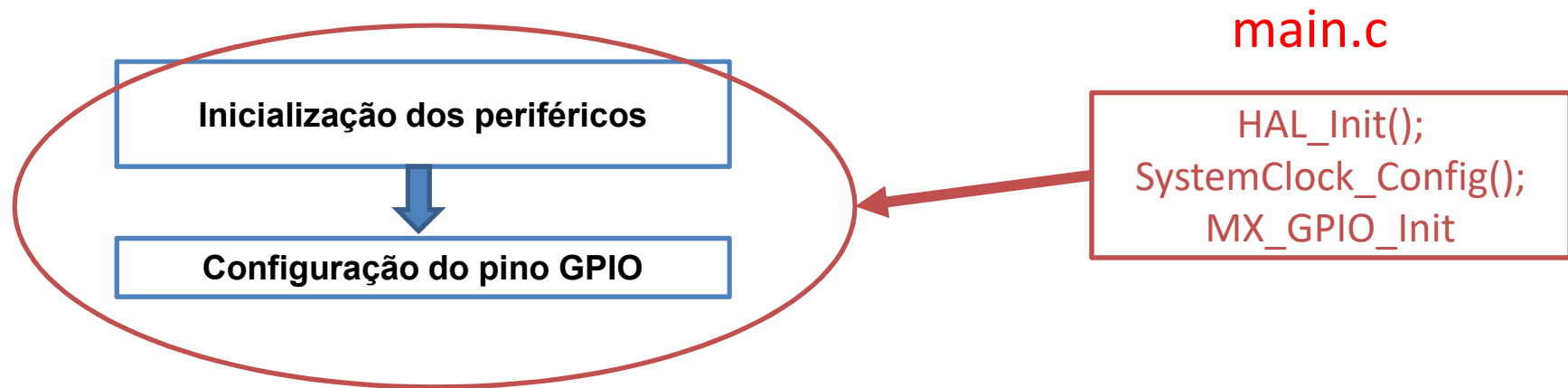
11. Gere template para código:

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



Exemplo 1- Configurar **GPIO** para piscar LED

Código Gerado



HAL_Init(): Função para inicialização da biblioteca HAL. Deve ser a primeira instrução a ser executada no programa: configura *prefetch* da Flash e memória cache para instruções e dados.

Exemplo 1- Configurar GPIO para piscar LED

12. Abra o projeto no Keil

13. Deve-se inserir funções em **main.c** (em *Application User*)

- Entre os rótulos */* USER CODE BEGIN 3 */* e */* USER CODE END 3 */*, dentro do loop infinito *while(1){ }*, inserir as funções:

HAL_Delay // para inserir atraso

HAL_GPIO_WritePin // ou *HAL_GPIO_TogglePin* para piscar o led

```
/* USER CODE BEGIN 3 */
/* Infinite loop */
while (1)
{
    HAL_GPIO_WritePin(GPIOG, GPIO_PIN_14, GPIO_PIN_SET);
    HAL_Delay(500);

    HAL_GPIO_WritePin(GPIOG, GPIO_PIN_14, GPIO_PIN_RESET);
    HAL_Delay(500);
}
/* USER CODE END 3 */
```

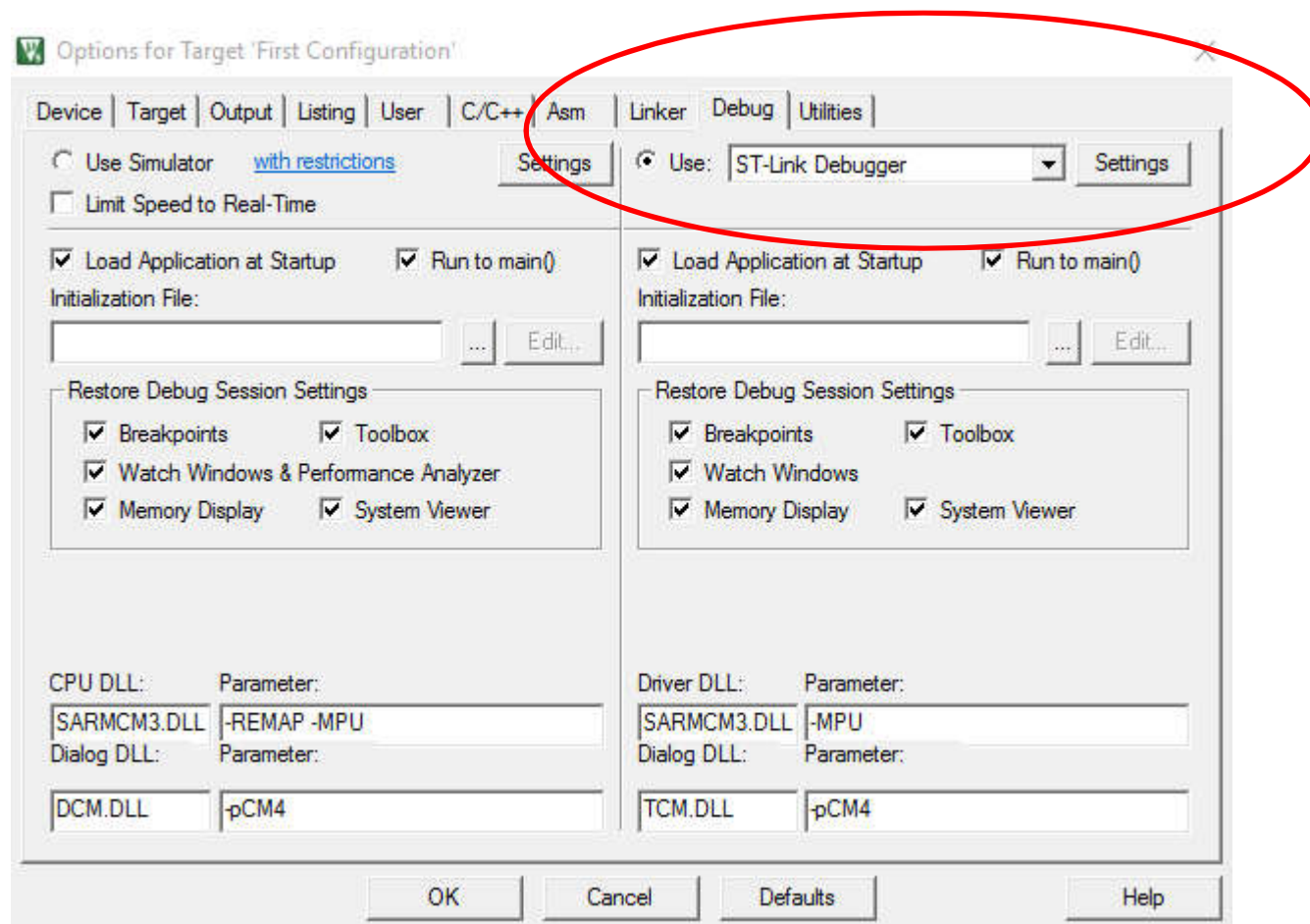
Exemplo 1- Configurar **GPIO** para piscar LED

❑ Outra opção:

```
/* USER CODE BEGIN 3 */
/* Infinite loop */
while (1)
{
    HAL_GPIO_TogglePin(GPIOG, GPIO_PIN_14);
    HAL_Delay(500);
}
/* USER CODE END 3 */
```

Exemplo 1- Configurar GPIO para piscar LED

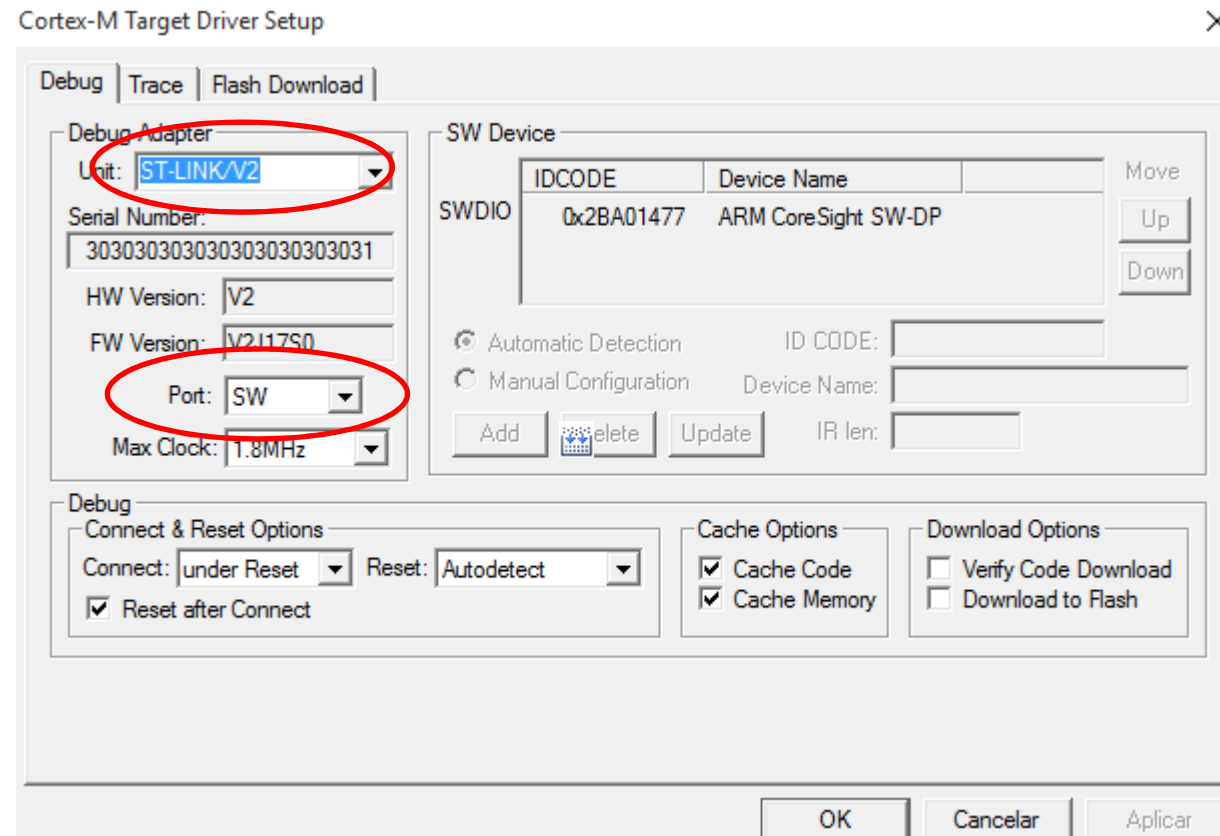
14. Para fazer o download do código no kit, pressione ALT+F7 e selecione o tab Debug:



❑ Clique em *Settings* e....

Exemplo 1- Configurar GPIO para piscar LED

15. Verifique os campos abaixo:



16. Compile



carregue no kit



e pressione o botão de reset do kit