UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
SISTEMAS MICROPROCESSADOS

Atividade Prática 03 - Timers e PWM

Professor(a): Ricardo Jardel Nunes Silveira

Introdução

Nesta atividade prática iremos abordar o periférico de temporização ou timers. Eles serão utilizados em dois cenários: controle do buzzer e no controle de brilho de um LED.

No primeiro caso, um sinal de GPIO convencional foi ligado ao transistor que aciona o buzzer (página 5 do esquemático). Porém, sabe-se que os buzzers produzem sons diferentes quando se aplicam formas de ondas quadradas de diferentes frequências. Desta forma, vamos utilizar o TIMER13 para nos fornecer a temporização correta para acionar e desacionar o GPIO do buzzer de forma que ele produza uma forma de onda quadrada.

Já no segundo caso, utilizaremos um dos pinos (ou canais) associados com o TIMER1 para gerar um sinal de PWM de frequência alta o suficiente para que o acionamento do LED não seja percebido a olho nu. Sinais de PWM de *duty cycle* variados irão produzir brilhos variados no LED selecionado.

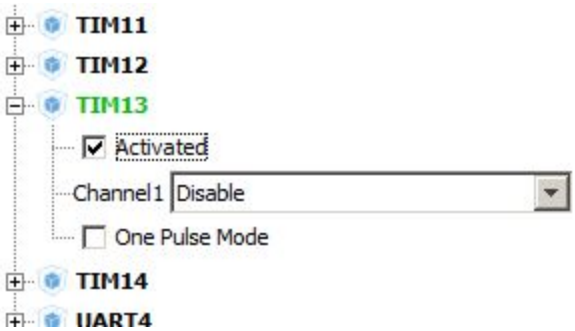
Para cada uma das atividades citadas criaremos um projeto diferente. O primeiro deverá ser chamado de `soft_pwm` e o segundo de `pwm`, apenas.

Parte 1 - Acionamento do Buzzer Utilizando o TIMER13 (*soft_pwm*)

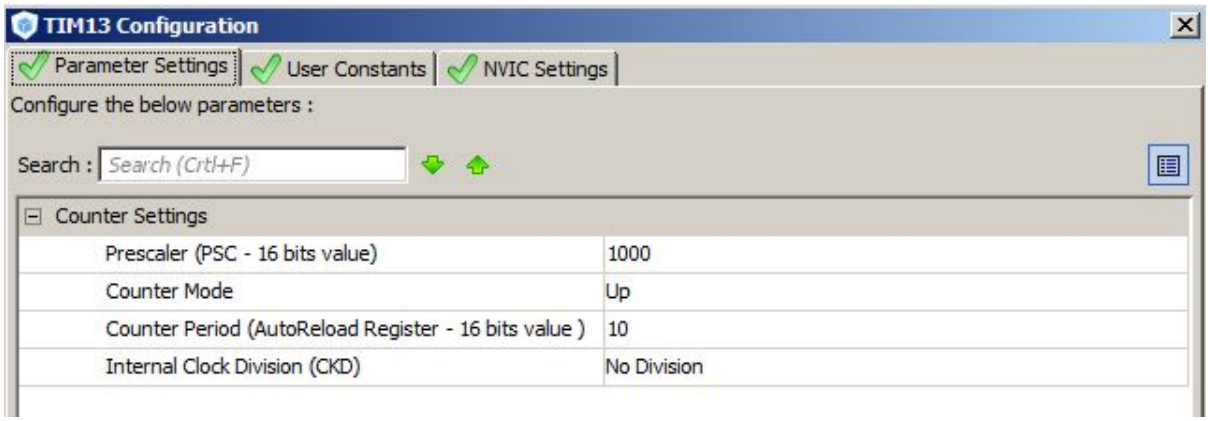
1. Crie um projeto chamado `soft_pwm` na ferramenta STM32CubeMX. Neste ponto verifique na placa o part number do microcontrolador para que o projeto seja criado para o alvo correto.

2. Habilite a utilização do clock externo através do uso do cristal conforme realizado em práticas anteriores.

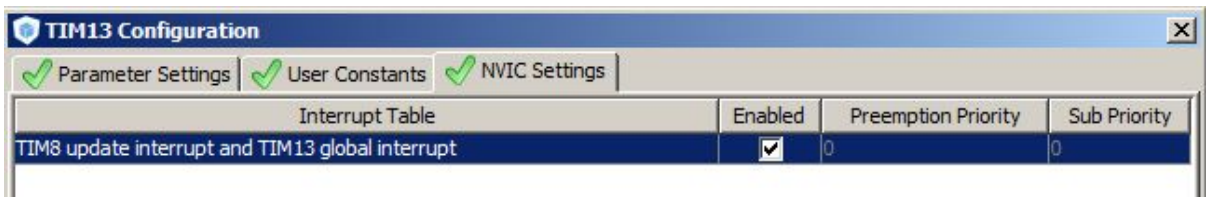
3. Habilite o TIMER13 no painel de configuração localizado no lado esquerdo da interface do STMCubeMX conforme a figura abaixo:



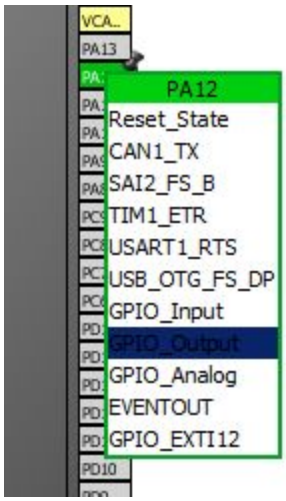
4. Vá para a aba *Configuration* e clique no botão TIM13 para configurar o TIMER13. Na caixa de diálogo que se abre preencha a aba *Parameter Settings* com o valores da caixa de diálogo da figura abaixo:



5. Na mesma caixa de diálogo visite a aba *NVIC Settings* e habilite a interrupção do TIMER13. A figura abaixo ilustra o procedimento:



6. Configure o pino PA12 como *GPIO_Output* para que este funcione como um pino de saída convencional.



7. Gere o código utilizando os mesmos procedimentos das práticas anteriores. Não esqueça de selecionar a IDE TrueStudio como o ambiente de desenvolvimento alvo.

8. Vamos agora iniciar a codificação do nosso projeto chamando a função que inicia a contagem do TIMER13. Insira o seguinte código entre os comentários `/* USER CODE BEGIN 2 */` e `/* USER CODE END 2 */`:

```
HAL_TIM_Base_Start_IT(&htim13);
```

9. Vamos implementar também a função que tratará a interrupção de estouro do timer. Insira o seguinte código entre os comentários `/* USER CODE BEGIN 4 */` e `/* USER CODE END 4 */`:

```
void HAL_TIM_PeriodElapsedCallback(TIM_HandleTypeDef *htim)
{
	if (htim == &htim13)
	{
		HAL_GPIO_TogglePin(GPIOA, GPIO_PIN_12);
	}
}
```

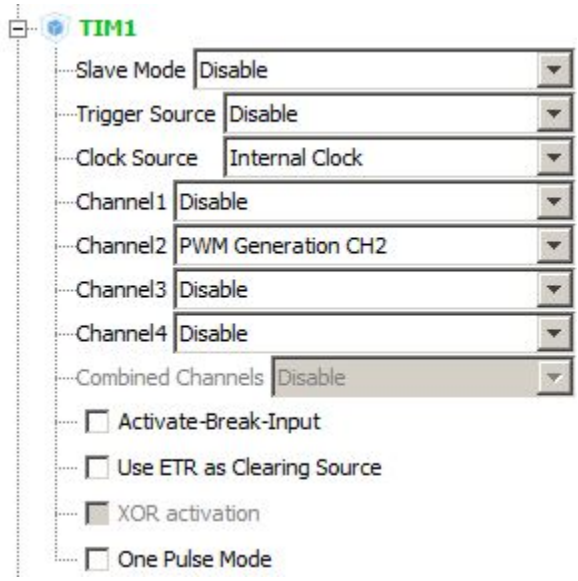
10. Compile o projeto, converta o arquivo *.elf* para um arquivo *.hex* e utilize a USB para gravar seu código. Neste ponto você deverá ouvir o buzzer emitindo um sinal sonoro.

Parte 2 - Brilho Variável do LED Utilizando o TIMER1 (*pwm*)

1. Crie um projeto chamado pwm na ferramenta STM32CubeMX. Neste ponto verifique na placa o part number do microcontrolador para que o projeto seja criado para o alvo correto.

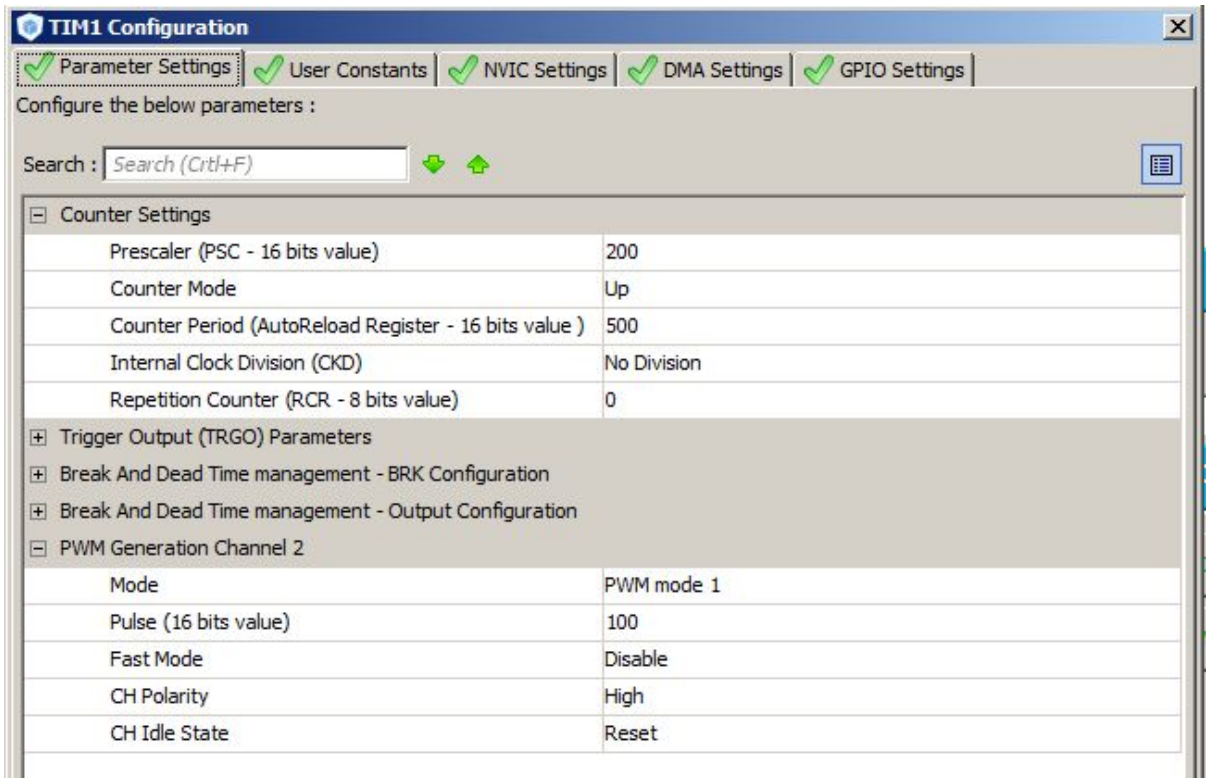
2. Habilite a utilização do clock externo através do uso do cristal conforme realizado em práticas anteriores.

3. Configure o TIMER1 no painel de configuração localizado no lado esquerdo da interface do STMCubeMX conforme a figura abaixo:



Neste ponto a ferramenta deverá ter pintado de verde o pino PE11 e o marcado com uma etiqueta TIM1_CH2. Esta marcação indica que iremos utilizar este pino como um canal de PWM controlado pelo TIMER1. Você pode verificar no esquemático da placa que este pino corresponde ao LED verde da placa.

4. Vá para a aba *Configuration* e clique no botão TIM1 para configurar o TIMER1. Na caixa de diálogo que se abre preencha a aba *Parameter Settings* com o valores da caixa de diálogo da figura abaixo:



5. Gere o código utilizando os mesmos procedimentos das práticas anteriores. Não esqueça de selecionar a IDE TrueStudio como o ambiente de desenvolvimento alvo.

6. Vamos agora iniciar a codificação do nosso projeto chamando a função que inicia a geração do sinal de PWM no pino PE11. Insira o seguinte código entre os comentários `/* USER CODE BEGIN 2 */` e `/* USER CODE END 2 */`:

```
HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_2);
```

7. Compile o projeto, converta o arquivo *.elf* para um arquivo *.hex* e utilize a serial para gravar seu código. Neste ponto você deverá ver o LED verde parcialmente acesso. Este brilho reduzido é devido ao sinal que o aciona pois este não possui *duty cycle* de 100%. Realize medições sobre o LED e verifique se a relação de porcentagem do tempo em ALTO da onda quadrada pelo seu período é a mesmo da relação do parâmetro *Pulse* e *Counter Period* do TIMER1.

8. Altere os valores de *Pulse* em passos de 50 e veja o brilho do LED variando conforme a alteração do *duty cycle*.

Questionário

1. Qual o papel dos parâmetros Prescale e Counter Period configurados na caixa de diálogo do TIMER13? Utilize um osciloscópio e realize medições sobre o resistor R401 de forma a entender como estes parâmetros afetam a frequência da forma de onda gerada nesse resistor.

2. Visite o manual de referência do microcontrolador e identifique as principais diferenças entre os TIMERs 13 e 1.

3. Descreva os outros modos de utilização dos canais independentes do timers: *Input Capture*, *Output Compare* e *One-pulse mode output*.