

1 Eventos de Tempo

Na disciplina estudamos um fragmento de código que usa a Interrupção de tempo para periodicamente disparar um tratador e executar determinada funcionalidade (como um pisca-pisca). Deseja-se melhorar o tratador de interrupção para criar um agendador de eventos de tempo.

A API para esse agendador deve ser algo como:

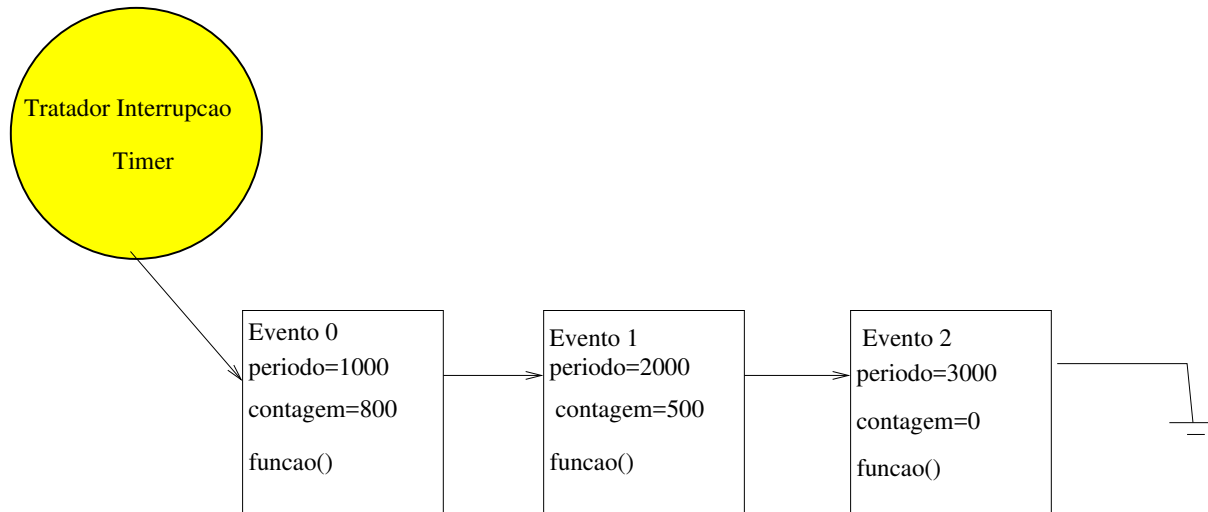
```
id = evento_periodico(tempo_ms, funcao_rodar);
```

onde o tempo_ms é um valor numérico que indica de quanto em quanto tempo é necessário disparar o evento. A "função_rodar" é o endereço de uma função que deve ser executada. Ex:

```
id = cria_evento_periodico(1000, funcao_pisca_led);  
id2 = cria_evento_periodico(2000, funcao_pisca_led2);  
id3 = cria_evento_periodico(3000, funcao_pisca_led3);  
.  
.  
.  
remove_evento_periodico(id1);
```

O agendador deve permitir que vários eventos estejam ativos.

Para isso no estilo da figura abaixo precisa ser criado:



Você talvez necessita controlar o acesso a uma região crítica, para isso pode desabilitar interrupções.

```
void habilita_interrupcao_timer (void)  
{  
    NVIC_EnableIRQ(RIT_IRQn);  
}
```

```
void desabilita_interrupcao_timer (void)  
{  
    NVIC_DisableIRQ(RIT_IRQn);  
}
```

Pense em como alterar o agendador de eventos periódicos para prover uma função delay_ms(tempo_ms) para o usuário.

Data limite: 8/10/2018