

175480_09

EA871

Patrick de C. T. R. Ferreira

Interação com o código:

Botões:

S1: Faz o toggle entre start e pause. Se o programa não está contando e este é clicado, o cronômetro é disparado. Se o cronômetro está fazendo a contagem e este botão é clicado, o cronômetro pausa;

S2: faz o reset do cronômetro. A qualquer momento em que este é clicado, a contagem de tempo é reiniciada do zero;

S3: Registra o valor atual do tempo em uma LAP. A lap atual é exibida no display LCD e todas as LAPs registradas são exibidas no terminal. Caso seja requisitada salvar mais uma LAP após as três vagas estarem preenchidas, o programa vai sobrescrevendo-as uma a uma em ordem.

Teclas:

s: Dispara a contagem do cronômetro;

p: Pausa a contagem do cronômetro;

r: Reseta o cronômetro;

l: Grava uma LAP do cronômetro.

Atividade 1:

1. Quatro níveis de prioridade, onde cada índice determina qual interrupção pode interromper cada rotina, seguindo a tabela 3-7 Interrupt vector assignments do manual do KL25.
2. É uma interrupção que não pode ser desabilitada.
3. Indica qual a interrupção implementada em cada posição do vetor de interrupções, com seu respectivo IRQ e sua descrição.

4. Core vectors são posições do vetor com interrupções diretamente para o processador ARM do sistema. Os non core vectors são as posições que atuam em outros componentes do sistema, como no DMA.
5. É o vetor SysTick_Handler, de acordo com o arquivo kinetis_sysinit.c .
6. O vetor de interrupções do arquivo kinetis_sysinit.c faz a associação de SysTick_Handler a **void SysTick_Handler(void)**.
7. Pois esta função já está definida no arquivo kinetis_inti.c. Contudo, sua definição permite que, caso **void SysTick_Handler(void)** seja definida novamente, a nova definição se sobreponha à do arquivo kinetis_sysinit.c.
8. É o índice 15 segundo a tabela 3-7 Interrupt vector assignments do manual do KL25.
9. Esta vetor direciona a execução do programa para as rotinas de interrupção adequadas e contém também a declaração das mesmas. No arquivo também está contido o vetor de interrupções com as ordens de prioridade para tratamento das mesmas, segundo o manual do KL25.
10. void SysTick_Handler() __attribute__((weak, alias("Default_Handler"))); é responsável por declarar void SysTick_Handler() caso o programador não o faça. Com isso, não haverá erro de não declaração de função caso o programador não declare esta função, ela será declarada devido a sua atribuição como Default_Handler. Porém, caso o programador a declare, a nova declaração se sobrepõe a esta.