EA871 – Laboratório de Programação Básica de Sistemas Digitais

Atividade 07 – 2º semestre de 2023

1. Objetivos

- Contrastar o funcionamento de soluções para contagem de tempo baseadas em espera ativa e em interrupções de temporizador.

2. Atividade da Aula (faremos em aula, juntos)

Nosso objetivo é piscar o LED incorporado à placa, conectado no pino 13, usando o temporizador 0 para controlar os intervalos de tempo que o LED permanece aceso e apagado.

a) Determine os valores a serem atribuídos aos registradores TCCR0A, TCCR0B e TIMSK0 para que o temporizador opere no modo normal, ou seja, contagem até o valor máximo (255), *prescaler* = 8 e interrupção por *overflow* ativa. Nessa configuração, qual é o intervalo de tempo entre as interrupções?

b) Utilize o temporizador 0, configurado conforme especificado no item anterior, para piscar o LED do pino 13. O LED deve permanecer 256 ms em cada estado. Construa o programa e valide-o no Tinkercad, no MPLAB X e na placa.

3. Resumo da Atividade (individual, para entrega)

O desafio proposto nesta atividade é desenvolver um programa que faça com que dois LEDs pisquem em diferentes frequências, ao mesmo tempo em que uma mensagem de texto é continuamente transmitida pela UART. Mais especificamente, desejamos que o LED incorporado (pino 13) pisque com uma frequência de 1 Hz (0,5 s aceso, 0,5 s apagado), e que um LED externo, conectado ao pino 12 da placa Arduino, pisque de forma a ficar 0,78 s aceso e 0,78 s apagado. Utilize o circuito disponível em:
https://www.tinkercad.com/things/1h5eJtCssdU

A solução para a atividade deve necessariamente explorar a interrupção do *timer* 0 associada ao modo *Clear Timer on Compare Match* (CTC). As especificações do temporizador (e.g., *prescaler* e valor máximo da contagem) devem ser projetadas por cada aluno e justificadas nos comentários do código-fonte. Notem que há várias maneiras de se configurar o temporizador para obter o comportamento desejado do sistema. Não é permitido utilizar a função _delay_ms() ou qualquer outro tipo de espera ativa para temporizar os LEDs.

Em relação à UART, vamos utilizar a mesma configuração da atividade 6 (interrupções). Além disso, assim como feito anteriormente, após a mensagem de texto ser enviada uma vez, deve ser introduzido um atraso de 5 segundos utilizando a função _delay_ms() antes de se iniciar a nova transmissão. Substituam o vetor de caracteres a ser transmitido pela seguinte mensagem:

```
char msg[] = "Atividade 7 - Interrupcoes temporizadas ajudam a tratar concorrencia entre tarefas! \
n\n";
```

Observação: uma abordagem baseada somente em espera ativa (i.e., que usa rotinas de atraso para gastar tempo) encontraria algumas dificuldades para manter os vários dispositivos (LEDs e UART) simultaneamente em operação, dada a concorrência existente entre as tarefas.

Especificações da USART:

- Velocidade de transmissão normal (i.e., modo *double-speed* desativado);
- Modo de transmissão multi-processador desabilitado;
- Número de bits de dados por *frame* igual a 8;
- Modo assíncrono de funcionamento da USART;
- Sem bits de paridade;
- Uso de um bit de parada;
- *Baud rate* igual a 19.200 bps.

IMPORTANTE: Algumas vezes, a ordem com que os registradores do temporizador são configurados pode travar a simulação no Tinkercad. Trata-se de um problema do simulador que não tem base no manual do microcontrolador. Portanto, sugerimos a seguinte ordem na sequência de configuração: OCR0X, TIMSK0, TCCR0B e TCCR0A.

Instruções para a submissão do trabalho

1) Nos comentários do código-fonte, explique o funcionamento geral do programa e justifique as operações e os valores carregados em todos os registradores. **Em especial, como há diferentes estratégias de uso do temporizador para se atingir o intervalo de tempo desejado, é fundamental que a opção escolhida pelo aluno seja explicada nos comentários logo no início do código-fonte, de modo a justificar as configurações do temporizador.** Lembre-se que seus comentários fazem parte do relatório. Programas sem comentários terão nota máxima 5. Justificativas de configuração incorretas, mesmo que a configuração esteja certa, serão penalizadas fortemente, principalmente se tiverem sido discutidas em exemplos disponibilizados em vídeo.
2) Ao final da atividade, salve o código-fonte com nome "seu_ra.txt" (Exemplo: 025304.txt).
3) Faça o **upload** da sua solução da atividade 7 no Google Classroom.