

EA871 – Laboratório de Programação Básica de Sistemas Digitais

Atividade 09 – 2s2023

1. Objetivos

- Familiarização com a programação em linguagem simbólica (*Assembly*).
- Familiarização com o processo de conversão de instruções em linguagem simbólica para o código de máquina que a representa.
- Verificação do funcionamento do mecanismo de interrupção do processador.

2. Exercícios em sala de aula

- 1) Determine o que fazem as instruções a seguir e obtenha seus correspondentes códigos de máquina (consulte o manual do microcontrolador):
 - `jmp 0x0034`
 - `sbi 0x07,0`
 - `sts 0x002A,r16`

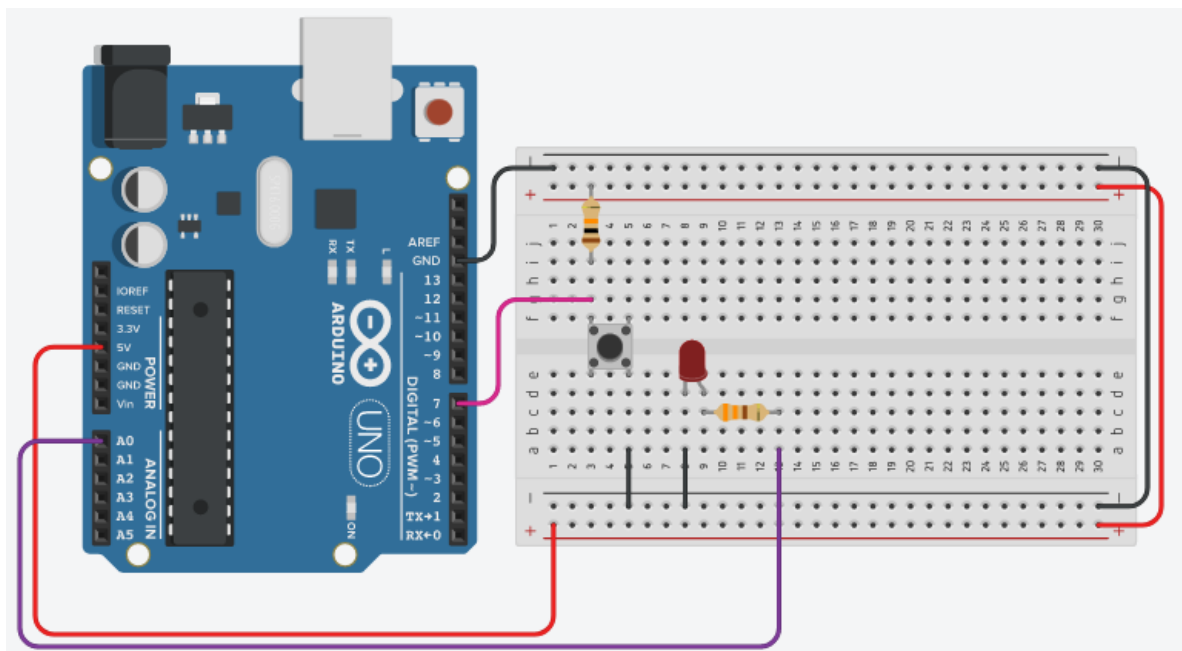


Figura 1: Circuito com botão e LED

- 2) Considere o circuito mostrado na Figura 1 em que há um botão conectado ao pino 7 do Arduino Uno (PD7) e um LED conectado ao terminal A0 (PC0). O programa em linguagem simbólica a seguir faz com que o LED acenda enquanto o botão está pressionado e apague caso contrário.

```
.org 0x0000    ; o programa começa no endereço 0x0000 (de palavra) da memória de programa, que corresponde ao
               endereço carregado no PC (Program Counter) após um reset do microcontrolador
jmp main      ; salto para evitar a região da memória onde ficam armazenados os vetores de interrupção

.org 0x0068    ; primeiro endereço (de byte) livre fora da faixa reservada para os vetores de interrupção
main:

sbi 0x07,0    ; configura o pino A0 como saída
cbi 0x0A,7    ; configura o pino 7 como entrada
cbi 0x08,0    ; apaga o LED inicialmente

loop:

in r16,0x09   ; lê o estado do botão
andi r16,0x80 ; aplica máscara para isolar o bit de interesse
brbs 1,liga_led ; desvia de acordo com o estado do botão
cbi 0x08,0    ; apaga o LED
rjmp loop
```

```
liga_led:
```

```
sbi 0x08,0 ; acende o LED
```

```
rjmp loop
```

- Qual a função da diretiva “.org”?
- O que simbolizam os rótulos “main”, “loop” e “liga_led”?
- Valide o funcionamento do programa no simulador do MPLAB X através da verificação do conteúdo dos registradores envolvidos no monitoramento do botão e no acionamento do LED.
- Verifique o conteúdo da memória de programa nos endereços (de palavra) 0x0000 e 0x0034. Os valores correspondem ao que você esperava. Justifique.

- 3) Repita o exercício anterior usando a interrupção *Pin Change* para detectar o estado do botão e acionar o LED. Observe o mecanismo de interrupção através da execução do programa passo a passo no simulador. Monitore o contador de programa (PC) e as mudanças na pilha ao longo do processo.

3. Atividade para entrega

Faça um programa que pisca o LED presente na placa (terminal 13). O LED deve acender por 100ms e depois permanecer apagado por 100ms, repetindo este ciclo indefinidamente, ou seja, deve piscar ininterruptamente com uma frequência de 5Hz. Utilize interrupções do temporizador 0 para medir os tempos necessários para produzir o comportamento especificado do LED. **Se optar pelo modo CTC, não use a interrupção associada ao evento de *overflow*.**

Construa o programa e valide-o utilizando o simulador do MPLAB X. Use a ferramenta “*Stopwatch*” (**Window -> Debugging -> Stopwatch**) do simulador, que permite que o intervalo de tempo entre dois *breakpoints* seja medido com precisão. Lembre-se que o sinal de relógio do Arduino UNO é de 16 MHz e é necessário configurar o simulador do MPLAB X com esta frequência para que as medidas de tempo observadas durante a simulação estejam corretas. Após validar o funcionamento usando o simulador, teste o programa na placa.

Instruções para a submissão do trabalho

- 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 configurações do temporizador que permitem resolver o desafio, é fundamental que seja justificado como os valores dos registradores foram obtidos.** Lembre-se que seus comentários fazem parte do relatório então procure demonstrar neles que entendeu o que fez. Programas sem comentários terão nota máxima 4. Justificativas de configuração incorretas, mesmo que a configuração esteja certa, serão penalizadas fortemente, principalmente se tiverem sido discutidas em exemplos disponibilizados em vídeo.
- Ao final da atividade, salve o código-fonte com nome “**seu_ra.txt**” (Exemplo: 025304.txt).
- Faça o upload da sua solução da atividade 9 no Google Classroom.