

# Atividade Prática

## Animação *PushLEDs* em C

### 1 Itens abordados

Para completar esta atividade prática, você precisa conhecer os seguintes tópicos:

- Periféricos:
  - *Ports* de I/O.
- Linguagem C:
  - Operações bit a bit.

### 2 Critérios Avaliativos

Esta atividade deve ser postada em um arquivo compactado **em formato ZIP** no SIGAA impreterivelmente até às \_\_\_\_:\_\_\_\_:\_\_\_\_ do dia \_\_\_\_/\_\_\_\_/\_\_\_\_. Recomenda-se a postagem da atividade o quanto antes, de modo a evitar problemas associados à indisponibilidade de acesso à Internet.

O arquivo zip deve conter:

- a) O arquivo *workspace* e o diretório *.vscode* do Visual Studio Code.
- b) Todos os códigos-fonte (.c, .cpp, .h, .hpp, .s, .inc, .asm) necessários para a compilação com sucesso do código.
- c) O arquivo de simulação do Proteus, compatível com a versão 8.6 SP2 build 23525.

A nota do atividade prática é baseada no funcionamento e na qualidade da solução apresentada. São critérios indispensáveis:

- Funcionamento da solução;
  - Cumprimento dos critérios estabelecidos.
- Organização do código;
  - Documentação (comentários necessários no código-fonte);
  - Endentação coerente;
  - Formatação adequada.
- Otimização;
  - Uso apropriado dos periféricos;

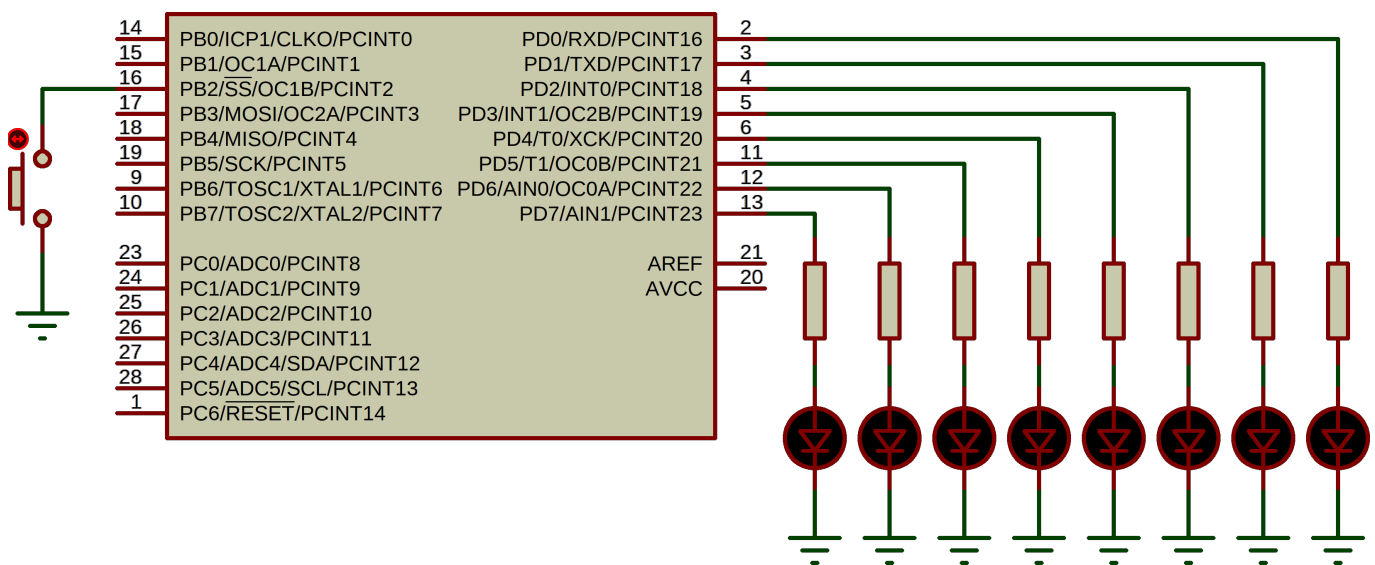
- Uso racional de memórias RAM, EEPROM e Flash.

### 3 Objetivo

Criar uma animação estilo *pushLEDs* em linguagem C, utilizando as operações de manipulação bit a bit.

### 4 Hardware

A seguir, é apresentado o diagrama esquemático do arquivo de simulação e a lista de componentes e suas configurações:



- Microcontrolador;
  - Device: ATMEGA328P;
  - Library: AVR2;
  - Part Reference: oculto;
  - Part Value: oculto;
  - Encapsulamento: SPDIL28;
  - Fuse CLKDIV8: (1) Unprogrammed;
  - Fuse CLKSEL: (1111) External crystal 8.0–MHz;
  - Clock Frequency: 16 MHz;
- Resistores;
  - Device: RES;
  - Library: DEVICE;
  - Part Reference: oculto;
  - Part Value: oculto;
  - Encapsulamento: RES40;
  - Model Type: DIGITAL;
  - Resistance: 1k5;

- LED;
  - Device: LED-GREEN<sup>1</sup>
  - Library: ACTIVE;
  - Part Reference: oculto;
  - Part Value: oculto;
  - LISA Model File: Digital;
  - Full drive current: 1mA;
- Button;
  - Device: BUTTON;
  - Library: ACTIVE;
  - Part Reference: vazio;
  - Part Value: vazio;
  - OFF Resistance: 100M;
  - ON Resistance: 100m;
  - Switching Time: 1m;

## 5 Requisitos Obrigatórios

- a) O código deve ser programado para o microcontrolador ATmega328P;
- b) O código deve ser programado em C.
- c) A animação a ser implementada é do tipo "*push-LEDs*", e funciona da seguinte forma:
  - (a) Considere que os LEDs são enumerados de 7 a 0;
  - (b) Um LEDs é chamado de *pushed* e um LED é chamado de *pusher*;
  - (c) Somente os LEDs *pusher* e *pushed* devem ser acessos;
  - (d) Inicialmente, o LED 0 é *pusher* e o LED 1 é *pushed*;
  - (e) A cada *frame* da animação, o LED *pusher* é incrementado até que ele se sobreponha ao LED *pushed*;
  - (f) Quando os LEDs *pusher* e *pusher* tornam-se coincidentes, o LED imediatamente à esquerda é aceso e torna-se o novo LED *pushed*;
  - (g) O LED *pusher* volta a ser o LED 0 e a animação continua a partir do passo do item c)e até que o LED *pushed* seja expelido do conjunto de LEDs, quando, então, a animação reinicia do item c)d.
- d) A duração (ou seja, o atraso) de cada passo da animação (*frame*) é de 250ms;
- e) Com exceção do primeiro *frame* do *loop* de animação, a lógica envolvida na formação de todos os outros *frames* deve, **obrigatoriamente**, utilizar operações de manipulação bit a bit, sendo **proibido** o uso da instrução de carregamento de imediato (LDI).
- f) Quando o botão é pressionado, a animação é pausada até que o botão seja pressionado novamente.
- g) O botão deve ser dotado de anti-ripple com *delay* de 10ms.
- h) O botão somente deve resumir a animação quando ele for liberado.




































## 6 Resultado Desejado

Cada um dos *frames* da animação é apresentado na tabela a seguir. Após o *frame* 35, a animação reinicia a partir do *frame* 1.

---

<sup>1</sup>Também são aceitos os componentes LED-AQUA, LED-BLUE, LED-ORANGE, LED-PINK, LED-PURPLE, LED-RED, LED-WHITE e LED-YELLOW. Mantendo o posicionamento dos componentes, o aluno está livre para usar a criatividade!

Tabela 1: Sequência de *frames* da animação.

<i>Frame</i>	LEDs	<i>Frame</i>	LEDs
1		19	
2		20	
3		21	
4		22	
5		23	
6		24	
7		25	
8		26	
9		27	
10		28	
11		29	
12		30	
13		31	
14		32	
15		33	
16		34	
17		35	
18		1	