

# **EA871 – Laboratório de Programação Básica de Sistemas Digitais**

Atividade 03 – 2º semestre de 2023

## **1. Objetivos**

- Familiarização com os ambientes de desenvolvimento (MPLAB X, IDE Arduino e Tinkercad) e seus recursos de depuração, bem como com o manual do microcontrolador ATmega328P.
- Configuração e utilização das portas de entrada e saída de propósito geral (GPIOs, do inglês *general-purpose input-output*).
- Estruturação de um código em linguagem C para trabalhar com dispositivos externos (periféricos) do microcontrolador.

## **2. Atividade de aula**

### **Exercício 1**

Neste exercício, o objetivo é utilizar uma GPIO como saída para acender um LED externo, o qual deve ser conectado ao pino 23 do microcontrolador ATmega328P.

- a) Use o diagrama de pinagem da placa de desenvolvimento Arduino UNO para identificar (1) o terminal a ser empregado dentre os conectores de acesso ao ATmega328P, assim como (2) a porta e o bit correspondente que devem ser utilizados para o funcionamento desejado.
- b) Quais registradores de configuração da porta devem ser utilizados?
- c) Consultando o manual do fabricante, obtenha os endereços de memória associados a cada um dos registradores.
- d) Monte o circuito no Tinkercad e escreva o programa que faça com que o LED permaneça aceso. Simule o sistema e verifique o seu funcionamento.
- e) Monte o sistema na prática utilizando o kit e verifique o seu funcionamento.

### **Exercício 2**

Agora, vamos acrescentar ao projeto um botão tátil, de modo a controlar o acionamento do LED. No caso, somente enquanto o botão estiver pressionado é que o LED deve acender. A fim de exercitarmos um pouco mais o uso de GPIOs, vamos utilizar o pino 13 do microcontrolador para a conexão com o botão.

- a) Use o diagrama de pinagem da placa de desenvolvimento Arduino UNO para identificar (1) o terminal a ser empregado dentre os conectores de acesso ao ATmega328P, assim como (2) a porta e o bit correspondente que devem ser utilizados para o funcionamento desejado.
- b) Quais registradores de configuração da porta devem ser utilizados?
- c) Consultando o manual do fabricante, obtenha os endereços de memória associados a cada um dos registradores.
- d) Monte o circuito no Tinkercad e escreva o programa que implemente o comportamento desejado.
- e) Monte o sistema na prática utilizando o kit e verifique o seu funcionamento.

### **Exercício 3**

Aproveitando a montagem do exercício anterior, modifique o comportamento do sistema de modo que o estado do LED seja alterado cada vez que o botão é pressionado. Ou seja, se o LED estava aceso e o botão foi pressionado, o LED é apagado; ou, analogamente, se o LED estava apagado e o botão foi pressionado, o LED acende. Para evitar comportamentos indesejados do sistema, vamos tratar o efeito de bounce (transitório) no botão explorando uma abordagem em software. Além disso, vamos utilizar o próprio resistor de pull-up interno do Arduino UNO para simplificar a montagem. Para inserir os atrasos necessários no programa para o debounce do botão, utilize as funções da biblioteca `delay.h`, cujos detalhes são discutidos no link:

[https://www.microchip.com/webdoc/AVRLibcReferenceManual/group\\_\\_util\\_\\_delay.html](https://www.microchip.com/webdoc/AVRLibcReferenceManual/group__util__delay.html)

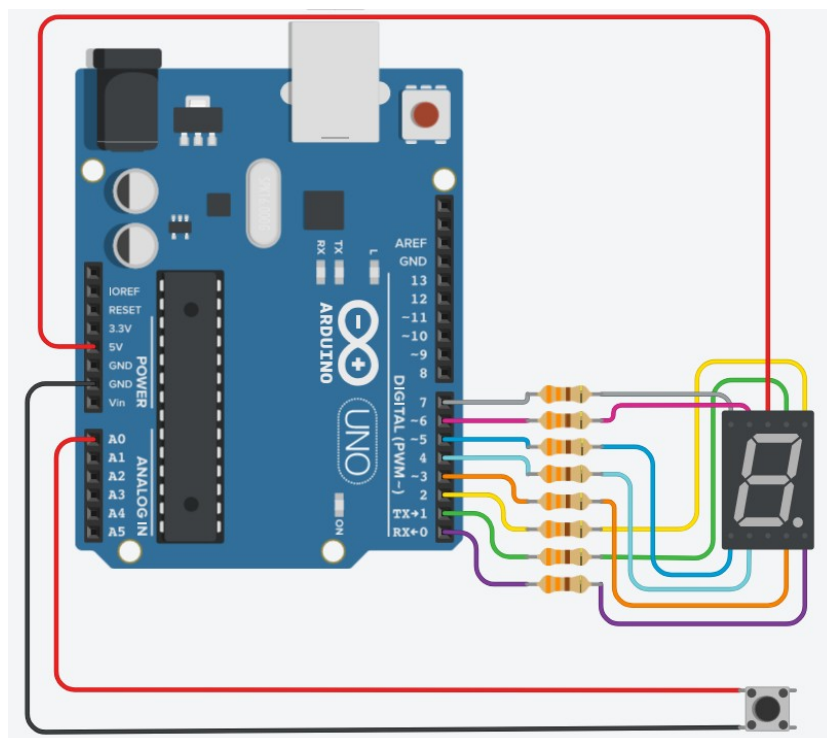
### 3. Resumo da atividade para entrega

O desafio proposto nesta atividade é desenvolver um programa que faça com que o microcontrolador ATmega328P conte quantas vezes um botão tátil (*push-button*) foi pressionado, exibindo o valor atual da contagem em um *display* de 7 segmentos. Além disso, enquanto o botão tátil estiver pressionado, o ponto decimal do *display* deve permanecer aceso, caso contrário deve permanecer apagado. Por simplicidade, vamos exibir somente um dígito, de maneira que a contagem estará limitada ao intervalo de 0 a F (hexadecimal). Ou seja, caso a contagem ultrapasse o valor F, ela deve ser reiniciada a partir de 0.

Para evitar comportamentos indesejados do sistema, utilize a mesma técnica de *debounce*, via *software*, implementada no exercício 3.

O *display* de 7 segmentos (anodo comum) pode ser acionado por um microcontrolador utilizando-se GPIOs distintas para acender ou apagar cada um dos seus segmentos. Nesse caso, considerando o ponto decimal, são necessárias oito GPIOs. Notem que, como estamos usando um *display* do tipo anodo comum, os segmentos se acendem com nível zero e apagam com nível 1 (lógica invertida).

**Importante:** Por convenção, vamos empregar os pinos 0 a 7 da porta D (ou seja, PD0 a PD7, indicados no diagrama de pinos do Arduino UNO, disponibilizado na página do curso) para acionar os segmentos de *a* a *g* do *display*, incluindo o ponto decimal *dp*. O botão, por sua vez, deverá ser conectado ao pino 0 da porta C (PC0). O link para o *template* do Tinkercad, mostrado na figura abaixo, está disponível em: <https://www.tinkercad.com/things/liSNWjIikD4>



Montagem do sistema.

**Materiais de consulta:** Para esta atividade, é recomendado consultar o material sobre **botões** na página do curso, assim como o material sobre **LEDs** e **displays de 7 segmentos**.

#### Instruções para a submissão do trabalho

- 1) Nos comentários do código-fonte (*main.c*), justifique as operações e os valores carregados em todos os registradores. Caprichem nos comentários, eles fazem parte do relatório.
- 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 3 no Google Classroom.