

Prática 3 - 2019.2

O objetivo desta prática é testar as entradas digitais do Arduino. Além disso, será testado um sistema um pouco mais complexo, com comportamentos diferentes dependendo de qual estado/modo o circuito se encontra. Neste tipo de problema, muito comum em sistemas digitais, a modelagem por **máquinas de estados** se torna interessante. Para que essa prática seja feita, o circuito e as funções feitas na Prática 2 serão reutilizadas.

1 Primeira Parte: entradas digitais

Usaremos a placa de expansão de entradas digitais, mostrada na Figura 1, para testar as entradas digitais. Para definir um pino como entrada digital basta usar a função

```
pinMode(ENTRADA, INPUT);
```

em que **ENTRADA** é um *define* com o número do pino que estamos usando como entrada e **INPUT** configura o pino como entrada. Para essa prática, usaremos os pinos 18 e 19 como entradas digitais (ver Figura 2).

Para ler o valor da entrada, utilize a função:

```
digitalRead(ENTRADA);
```

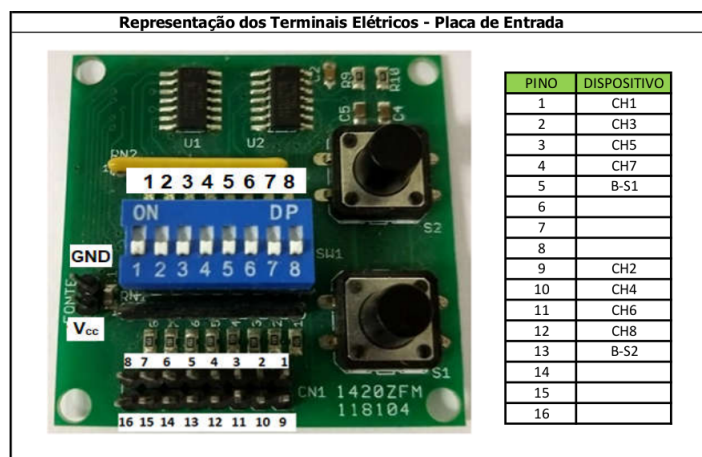


Figura 1: Pinagem da placa de entradas digitais (OBS: ligar a o V_{CC} em 3.3V).

Os dois pinos estão ligados a dois botões (utilizaremos **BOTA01** e **BOTA02** como os nomes dos botões) na placa de entradas digitais. A primeira parte da prática consiste em usar um dos botões para incrementar (**BOTA01**)¹ o valor que será mostrados nos displays de sete segmentos, e o outro botão (**BOTA02**) usado para decrementar essa variável.

Ou seja, nosso display de sete segmentos não será mais usado como um cronômetro, ele será usado para imprimir o valor atual de uma variável “contador”, que mudará de valor apenas quando um dos botões for pressionado.

¹Como um dos botões da placa está com defeito, utilizaremos um dos botões e uma das chaves, a mais próxima aos botões (DP8 na Figura 1).

Arduino Uno R3 Pinout

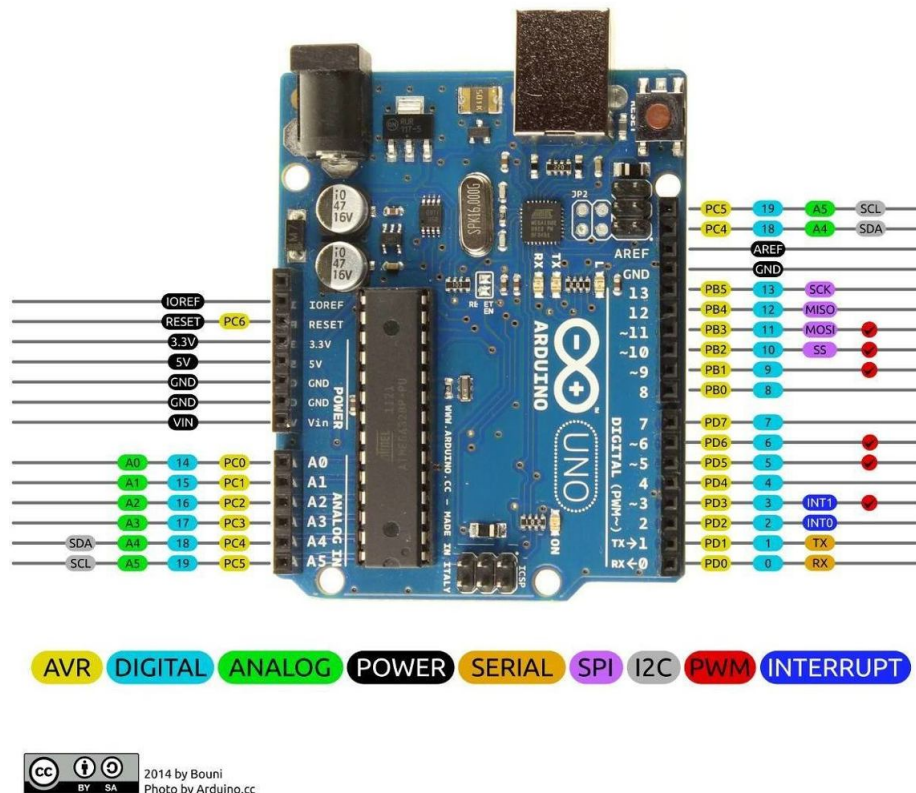


Figura 2: Pinagem do Arduino UNO - Rev. 3.

1.1 Detecção de borda de subida

Crie duas funções, `bordaBotao1()` e `bordaBotao2()`, que detecta a borda de subida dos sinais BOTAO1 e BOTAO2, respectivamente. Essa função será usada no próximo passo dessa prática para que apenas na borda de subida do botão a variável `contador` seja incrementada, ou decrementada.

1.2 Incremento e Decremento

Utilize as funções criadas na Seção 1.1 para criar uma nova função que atualiza o valor da variável `contador`. Lembrando que uma borda de subida BOTAO1 deve incrementar o valor de `contador` e uma borda de subida de BOTAO2 decrementa `contador`.

1.3 Mostrar nos displays

Utilize as funções criadas na Prática 2 para ficar imprimindo o valor atual da variável `contador` nos displays de sete segmentos.

2 Segunda Parte: modificando o comportamento do circuito

Para que não seja necessário ficar apertando várias vezes o botão para conseguir chegar a valores grandes, faça uma função que se um botão for apertado por 3 segundos, após isso, a variável começa a ser incrementada/decrementada a cada 100 ms que se passarem com o botão apertado.

Ou seja:

- um acionamento simples (menos de 3 segundo): incrementa/decrementa **contador** em uma unidade;
- um acionamento longo (mais de 3 segundo): incrementa/decrementa **contador** em uma unidade a cada 100 ms passados com o botão apertado.

Exemplo: se o botão for apertado por 3,5 segundos, ao final desse tempo, contador terá sido incrementado/decrementado em 5 unidades.

- ao soltar o botão, o sistema volta a esperar um novo acionamento.

Para ajudar no projeto, projete/desenhe a máquina de estados finitos que implementa esse controle lógico.

3 Terceira Parte: a volta do cronômetro

Nesta parte da prática, daremos uma atualização no contador de tempo implementado na Prática 2. A ideia é adicionarmos funcionalidades ao contador para podermos mudar os valores de cada um dos dígitos. Para isso, adicionaremos um modo “configuração” ao sistema.

3.1 Modo cronômetro

O modo “normal” do nosso sistema é funcionar como o contador de tempo, como o sistema do final da Prática 2.

3.2 Modo configuração

Caso algum dos botões seja segurado por 3 segundos, o contador deve entrar no modo de “configuração”. Ao entrar nesse modo, uma série de coisas deve para ocorrer.

1. **O display “pisca” 3 vezes, como intervalos de 50 ms:** Isso significa que os displays ficam apagados por 50 ms, seguidos por 50ms com o eles acesos, 50 ms apagados, etc. Durante esse tempo o sistema não responde as entradas.
2. **O BOTA01 vai mudar qual dos dígitos está sendo configurado:** para indicar qual dígito está sendo configurado, o dígito deve ficar piscando (50 ms aceso, 50 ms apagado, e assim por diante);
3. Assim que entra no modo configuração, o dígito D4 fica em modo de configuração. Um aperto no BOTA01, o sistema passa para configuração de D3, outro aperto passa para D2, outro para D1, caso ocorra outro aperto, volta para D4 e assim por diante. Mudar de dígito é sensível pela borda de subida.

4. **O BOTA02 vai incrementar o valor do dígito que está sendo configurado:** obedecendo, se for o dígito das dezenas de segundo, ele só varia entre 0 e 5. O incremento é cíclico, uma vez chegando em 9 (ou 5, dependendo do dígito) ele volta para 0.
5. Se o sistema estiver no modo de configuração e passar 3 segundos sem nenhum dos botões ter sido apertado, é para o sistema voltar para o modo cronômetro (modo “normal”).
6. Não é para ficar contando o tempo enquanto estiver no modo de configuração. O valor fica estático (a não ser que BOTA02 seja acionado).
7. Para voltar do modo “configuração” para o modo “cronômetro”, o sistema deve passar por um modo intermediário em que a tela ficará piscando por 3 vezes. Ou seja, do mesmo modo que acontece quando sai do modo “cronômetro” para o modo “configuração”.