

Atividade_03 - Livro AVR e Arduino – Técnicas de Projeto

Capítulo: 5 (Portas)

Obs.: Deve ser entregue arquivo contendo as perguntas e respectivas respostas.

Título: Acessando portas de entrada/saída usando C

Objetivos: Utilizar os registradores que controlam as portas de entrada / saída para controlar dispositivos.

Nesta prática utilizaremos o Tinkercad para simular um circuito simples usando o microcontrolador Atmega328, utilizado nas placas Arduino UNO. Desta vez, programaremos usando um código C com as diretivas/macros definidos pela AVR para acesso aos registradores.

1. Procedimentos:

1. Acesse sua conta no Tinkercad ([tinkercad.com](https://www.tinkercad.com)) e vá para a aba circuits (<https://www.tinkercad.com/circuits>).
2. Você deve fazer um circuito capaz de ler um botão. Note que este projeto já está disponível (na aba Starters → Arduino). O botão pode ser ligado na porta 2 (PD2) na placa do Arduino UNO (como no projeto Starter).
3. Você deve agora modificar a lógica do botão eliminando a necessidade do resistor externo. Você deve usar somente dois fios conectados ao botão, sendo um deles conectado ao GND. Agora adicione um segundo botão, utilizando a mesma lógica do primeiro, mas na porta 3 (PD3).
4. Você deve adicionar 3 leds, nas portas 7, 8 e 28 (PD7, PB0, PC5) (obrigatoriamente). Somente um led deve ser aceso por vez. A cada clique do botão, um led se apaga e o próximo acende. O segundo botão ativa os leds na ordem inversa. Use resistores onde necessário, pois vamos implementar estes circuitos usando a placa e não podemos sair queimando as coisas. Veja o apêndice desta prática.
5. Implemente um clique longo no primeiro botão (apertar e segurar por 800ms) para que todos os leds sejam apagados (uma espécie de reset dos leds). Note que a ação deve ocorrer assim que passados os 800ms. Ao primeiro clique com os leds todos apagados, um led se acende e volta a funcionar como descrito no item 4.

coloquei os 2 botões para resetar os leds

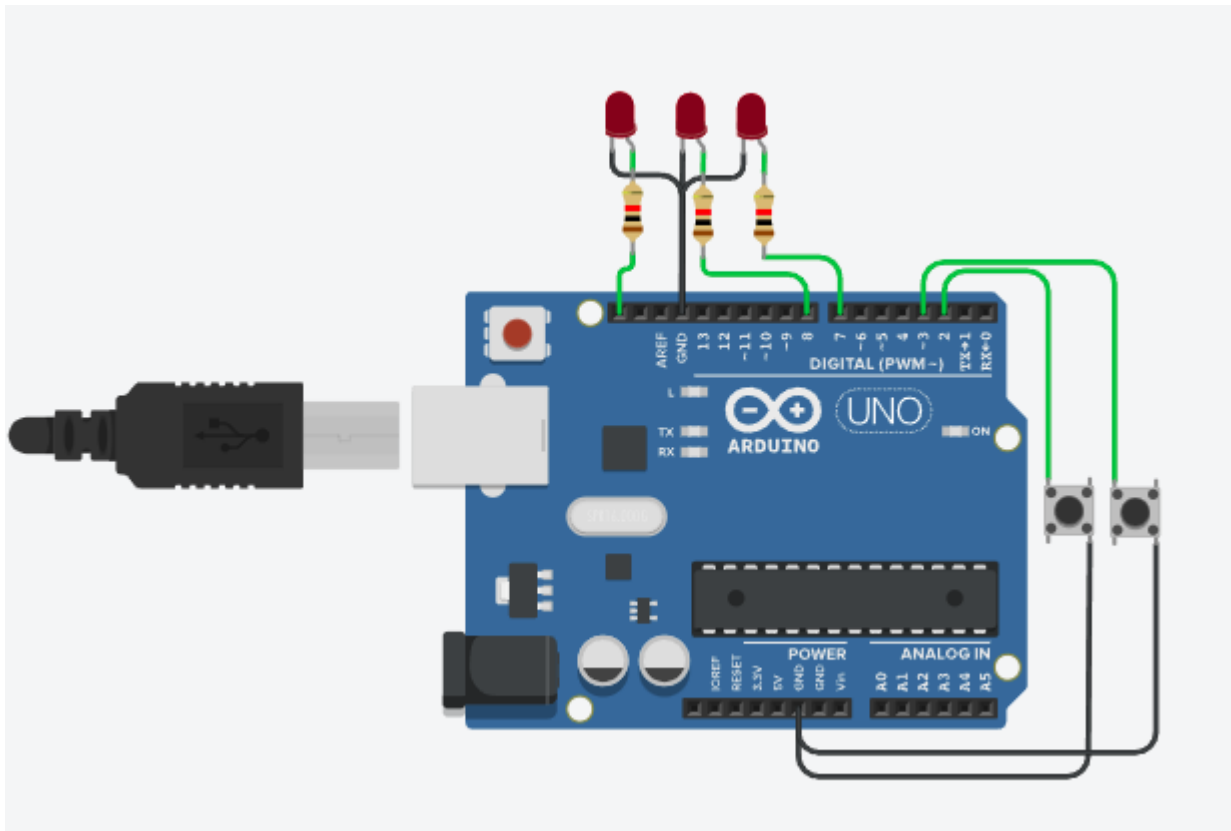
6. Os pinos não usados devem ser configurados como pinos de entrada.
7. Pergunta teórica: É interessante ficar preso em um laço lendo se um botão foi apertado? Quais problemas podem acontecer com o circuito implementado (problemas elétricos quando o botão é pressionado e quando está sendo despressionado – procure por button debouncing)? Seria mais interessante usar uma interrupção (imagine que temos muito código para executar no Loop)?

Não, pois o código não consegue realizar outras funções além do loop. um dos problemas é que, se em um pequeno espaço de tempo o botão for pressionado mais de uma vez pode ocorrer pequenas vibrações que tem como consequência toque nas placas.

Sim, pois com a interrupção não há necessidade de utilizar o loop.

8. Cole o código fonte do microcontrolador ao final deste arquivo e inclua a imagem de seu design. Importante: Deixe seu circuito público no Tinkercad e cole o link para ele aqui:

https://www.tinkercad.com/things/fOJym9N19U7-cool-vihelmo-elzing/editel?sharecode=VZ7AFwk2qu_3MIKIotJcY2tcPGqw8bd9HCupjAC74B8



```
#define timeToTurnOffLed 800
unsigned int debounceTime = 0;
unsigned int button = 0;
int led = -1;

void setup(){
    DDRB = (1 << PB0); // define PB0 como saida
    PORTB = 0;
    DDRC = (1 << PC5); // define PC5 como saida
    PORTC = 0;
    DDRD = (1 << PD7); // define PD7 como saida
    PORTD = 0b00001100; // define PD2 e PD3 como entrada
}

void loop(){
    // botao1
    if(!(PIND & (1 << PD2)))
    {
        debounceTime = millis(); // salva o tempo atual
        // utilizado para conta da linha 28
        while(!(PIND & (1 << PD2))); // n sai do while ate soltar o botao
        if((millis() - debounceTime) >= timeToTurnOffLed) // desliga o led e
reseta o contador se apertar botao por 800ms
        {
            PORTC &= ~(1 << PC5);
```

```

    PORTB &= ~(1 << PB0);
    PORTD &= ~(1 << PD7);

    led = -1;
}
else
{
    led++; // incrementa o contador a cada interação com o botao

    // reseta os leds
    PORTB = 0;
    PORTC = 0;
    PORTD = 0b00001100;

    // faz o contador do led ser 'circular'
    led = led > 2 ? 0 : led;
    // liga o led correto
    if(led == 0) PORTC = 1 << PC5;
    else if(led == 1) PORTB = 1 << PB0;
    else if(led == 2) PORTD |= 1 << PD7;
}
}

// botao2
if(!(PIND & (1 << PD3)))
{
    debounceTime = millis();
    while(!(PIND & (1 << PD3)));
    if((millis() - debounceTime) >= timeToTurnOffLed)
    {
        PORTC &= ~(1 << PC5);
        PORTB &= ~(1 << PB0);
        PORTD &= ~(1 << PD7);

        led = -1;
    }
    else
    {
        led--;

        PORTB = 0;
        PORTC = 0;
        PORTD = 0b00001100;

        led = led < 0 ? 2 : led;
        if(led == 0) PORTC = 1 << PC5;
    }
}

```

```
    else if(led == 1) PORTB = 1 << PB0;  
    else if(led == 2) PORTD |= 1 << PD7;  
  }  
}  
}
```

ATENÇÃO: A função/objetivo deve ocorrer no clique do botão, e não na sua soltura. No caso do clique longo, a função/objetivo deve ocorrer assim que o tempo limite for atingido.

ATENÇÃO: Usar as funções `pinMode()`, `digitalWrite()` e `digitalRead()` estão proibidos nesta prática. O uso delas fará a nota atribuída ser zero.

ATENÇÃO: Documente seu código. Cada linha/bloco deve deixar explícito o seu papel.

RÚBRICA:

Pergunta teórica: 15%

Circuito: 15%

Diretivas PORT e DDR: 30%

Lógica da programação: 40%

Valor desta atividade na média: 1.0