Atividade_03 - Livro AVR e Arduino – Técnicas de Projeto
Capítulo: 5 (Portas)

REGINALDO GREGÓRIO DE SOUZA NETO
2252813

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) 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 6, 9 e 27 (PD6, PB1, PC4) (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 900ms) para que todos os leds sejam apagados (uma espécie de reset dos leds). Note que a ação deve ocorrer assim que passados os 900ms. Ao primeiro clique com os leds todos apagados, um led se acende e volta a funcionar como descrito no item 4.

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)?

RESPOSTA:

Não é interessante ficar preso em um loop após a leitura de um botão que se mantém pressionado, tendo em vista que o principal problema pode ser a irregularidade na leitura do "despressionar" do botão, isso pode acarretar diretamente na ativação ou não do circuito como desejado. Ou seja, o bounce gera múltiplos cliques no botão em uma ocasião em que ele tenha sido clicado apenas um vez. Deste modo, é interessante utilizarmos ferramentas de interrupção para que o debounce ocorra, tais como o - millis() ou delay() –

que foram utilizados no código abaixo.

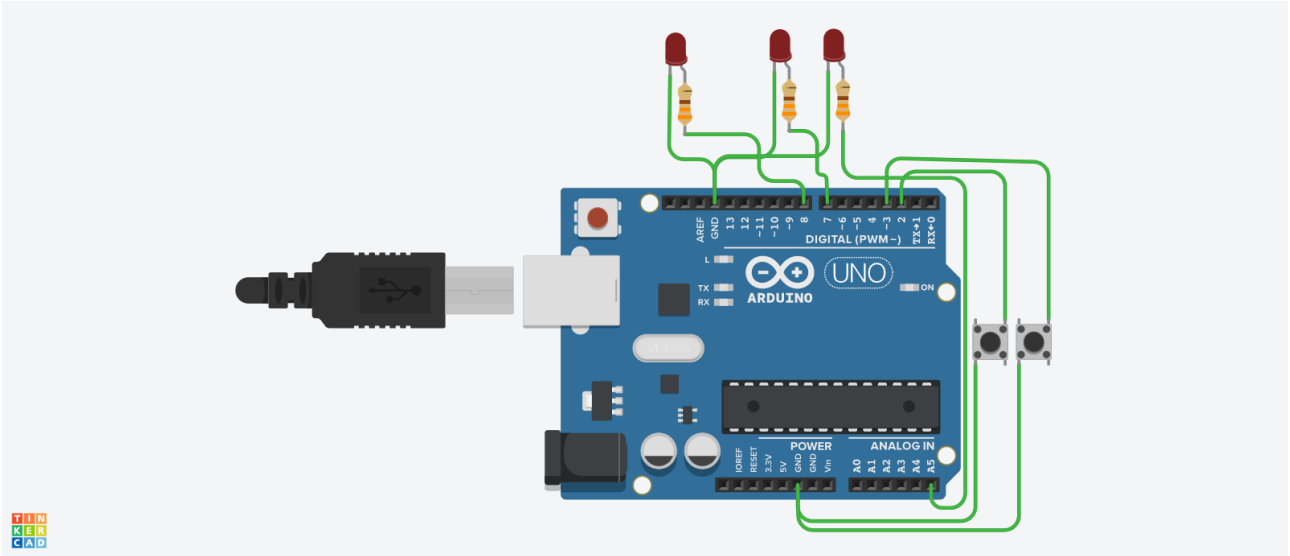
8. Cole o código fonte do microcontrolador ao final deste arquivo e inclua a imagem de seu design. Importante: Ative o compartilhamento de link nas propriedades do projeto e cole o link para ele aqui:

LINK:
https://www.tinkercad.com/things/g5583t8Iczh
CÓDIGO:

```
#define debounceInterval 50 // ms
unsigned int debounceTime = 0;
unsigned int debounceTime2 = 0;


int lastState = 0;
int lastState2 = 0;

int botao = 0; // estado do botao
int botao2 = 0; // estado do botao

int posicao = 0;

void setup()
{
   DDRD |= (1 << PD7);
   DDRB |= (1 << PB0);
   DDRC |= (1 << PC5);

   DDRD &= ~(1 << PD2) & ~(1 << PD3);

   PORTD |= (1 << PD2) | (1 << PD3);
}

void loop()
{
   // Acendendo os leds da esquerda para a direita
   int leitura = PIND & (1 << PD2);

   if (leitura != lastState){
      debounceTime = millis();
   }

   if ((millis() - debounceTime) > debounceInterval){
      if (botao != leitura){
         botao = leitura;
         if (botao == 0){
            if (posicao == 0){          // Quando não tiver nenhum aceso
               PORTB ^= (1 << PB0);    // acende o led1

               PORTC = ~(1 << PC5);    // apaga o led3
               PORTD = ~(1 << PD7);    // apaga o led2

               posicao++;
```

```
            }

            else if (posicao == 1){    // Quando o led1 estiver aceso
               PORTD ^= (1 << PD7);    // acende o led2

               PORTB = ~(1 << PB0);    // apaga o led1
               PORTC = ~(1 << PC5);    // apaga o led3

               posicao++;           // proxima posicao
            }

            else if (posicao == 2){    // Quando o led2 estiver aceso
               PORTC ^= (1 << PC5);    // acende o led3

               PORTD = ~(1 << PD7);    // apaga o led2
               PORTB = ~(1 << PB0);    // apaga o led1

               posicao++;
            }

            else if (posicao == 3){    // Quando o led3 estiver aceso
               PORTB ^= (1 << PB0);    // acende o led1

               PORTD = ~(1 << PD7);    // apaga o led3
               PORTC = ~(1 << PC5);    // apaga o led2

               posicao++;
               if(posicao == 4){
                  posicao = 1;
               }
            }
         }
      }
   }

lastState = leitura;

// Acendendo os leds da direita para a esquerda
int leitura2 = PIND & (1 << PD3);

if (leitura2 != lastState2){
   debounceTime2 = millis();
}

if ((millis() - debounceTime2) > debounceInterval){
   if (botao2 != leitura2){    //
      botao2 = leitura2;
      if (botao2 == 0){
         if (posicao == 0){
            PORTC ^= (1 << PC5);    // Acende o led3

            PORTD = ~(1 << PD7);    // Apaga o led2
            PORTB = ~(1 << PB0);    // Apaga o led1

            posicao--;
            if(posicao == -1){
               posicao = 3;
```

```
            }
        }

        else if (posicao == 3){
            PORTD ^= (1 << PD7);    // Acende o led2

            PORTB = ~(1 << PB0);    // Apaga o led1
            PORTC = ~(1 << PC5);    // Apaga o led3

            posicao--;
        }

        else if (posicao == 2){
            PORTB ^= (1 << PB0);    // Acende o led1

            PORTD = ~(1 << PD7);    // Apaga o led2
            PORTC = ~(1 << PC5);    // Apaga o led3

            posicao--;
        }

        else if (posicao == 1){
            PORTC ^= (1 << PC5);    // Acende o led3

            PORTD = ~(1 << PD7);    // Apaga o led2
            PORTB = ~(1 << PB0);    // Apaga o led1

            posicao--;
            if(posicao == 0){
                posicao = 3;
            }
        }
      }
    }
  }
  lastState2 = leitura2;
        delay(1);
}
```



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: 0.6