

PROFESSOR DANILO

AULA 07

Continuando a atividade anterior, hoje você deverá conseguir fazer um programa que apresenta uma contagem regressiva, de 9 até 8.

TAREFAS

Em casa, você deverá acessar o site do professor em <http://robotica.professordanilo.com/8.html>

Clique em "Segundo Trimestre", Aula 07 e role a página até onde está escrito "Atividade".

Lá você encontra videos e pode pedir ajuda ao seu professor.

DISPLAY DE 8 SEGMENTOS

Use o material da aula passada se ainda tiver alguma dúvida. O circuito básico é o apresentado a seguir, na Figura 1, onde vemos o circuito montado no tinkercad.

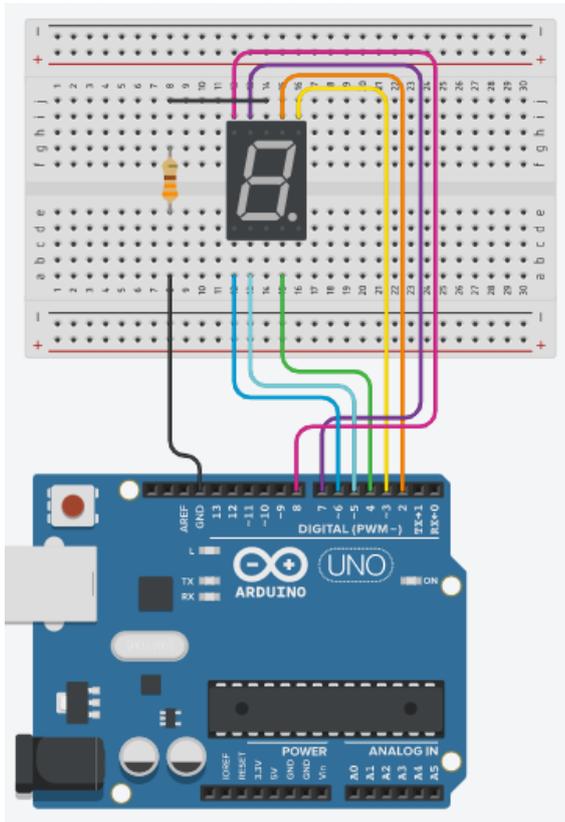


Figura 1: montagem do circuito

PROGRAMAÇÃO INICIAL

Primeiramente tente descobrir se o seu display é do tipo cátodo (GND) comum ou ânodo (V) comum, isto é, se precisa ligar o GND na parte comum ou se precisa ligar 5 V na parte comum. Nos dois casos, os programas serão muito semelhantes: a diferença é apenas que num dos displays você deve usar HIGH para ligar um segmento e no outro tipo você deve usar LOW para ligar o mesmo segmento.

Vou considerar inicialmente que o seu display é do tipo catodo comum, isto é, devemos conectar ao GND o pino em comum, conforme vimos na aula passada. Como HIGH é o mesmo que 1 e LOW é o mesmo que 0, se o programa der errado é só trocar os valores das variáveis ligar e desligar.

Vejamus parte do sketch.

ROBÓTICA – 9 ANO – 05/05/2022

```
/*Se seu programa não funcionar
 * tente inverter os valores
 * das variáveis ligar e desligar
 * abaixo
 */
```

```
int ligar=1;
int desligar=0;
```

```
void setup() {
  pinMode(2, OUTPUT);
  pinMode(3, OUTPUT);
  pinMode(4, OUTPUT);
  pinMode(5, OUTPUT);
  pinMode(6, OUTPUT);
  pinMode(7, OUTPUT);
  pinMode(8, OUTPUT);
}
```

```
void loop() {
  //mostra o número 8
  oito();
}
```

```
void oito(){
  //primeiro desligamos tudo
  for(int i=0; i<=13; i++){
    digitalWrite(i, desligar);
  }
  digitalWrite(2, ligar);
  digitalWrite(3, ligar);
  digitalWrite(4, ligar);
  digitalWrite(5, ligar);
  digitalWrite(6, ligar);
  digitalWrite(7, ligar);
  digitalWrite(8, ligar);
}
```

ANALISANDO O CÓDIGO

Uma das novidades que você terá na aula de hoje é sobre o que é função e o que é o loop for. Vejamos isso no código.

A primeira parte já foi discutida anteriormente e está explicada no comentário abaixo.

```
/*Se seu programa não funcionar
 * tente inverter os valores
 * das variáveis ligar e desligar
 * abaixo
 */
```

Isso quer dizer que se o seu programa não funcionar com as atribuições abaixo:

```
int ligar=1;
int desligar=0;
```

então você pode inverter os valores conforme apresentado abaixo:

```
int ligar=0;
int desligar=1;
```

E aí, seu display é do tipo cátodo comum ou ânodo comum? No setup, declaramos as saídas, como está descrito no código a seguir:

PROFESSOR DANILO

```
void setup() {  
  pinMode(2, OUTPUT);  
  pinMode(3, OUTPUT);  
  pinMode(4, OUTPUT);  
  pinMode(5, OUTPUT);  
  pinMode(6, OUTPUT);  
  pinMode(7, OUTPUT);  
  pinMode(8, OUTPUT);  
}
```

Agora temos nossa primeira novidade: veja que dentro do loop somente escrevemos `oito()`; Isso quer dizer que estamos chamando uma função que chamamos de `oito` e, quando escrevemos isso, no display aparece o número 8.

```
void loop() {  
  //mostra o número 8  
  oito();  
}
```

Como montamos tal função? Basta dizer o que ela retorna (no caso, por hora nenhuma função irá retornar nada, por isso escrevemos `void`, que quer dizer "vazio", ou seja, esta função não retorna nada). Depois, escrevemos o nome da função: este nome você escolhe e poderia ser qualquer outra coisa. Não se esqueça de colocar o parêntesis e os colchetes no estilo `void nome_da_funcao()` {conteúdo da função}.

```
void oito(){
```

Mas o que temos que colocar dentro da função? Bom, quando chamarmos esta função, teremos que desligar todos os LED (ta, nem todos, mas é mais fácil desligar todos do que saber quais teríamos que desligar). Para isso vamos usar o `for` loop.

```
//primeiro desligamos tudo  
for(int i=0; i<=13; i++){  
  digitalWrite(i, desligar);  
}
```

No `for`, temos que inicialmente declarar a variável que vamos usar (`int i=0`). Este é o valor inicial de `i`. Depois, verifica se o valor de `i` obedece a condição que está no meio (`i<=13`), ou seja, se `i` é menor ou igual a 13. Se esta condição é satisfeita, o que está entre `{ }` será executado. Depois disso, o valor de `i` altera-se de uma unidade (`i++`). O comando `i++` significa que temos que somar uma unidade no `i`.

Resumindo, o valor de `i` começa com o valor 0, como 0 é menor ou igual a 13, o comando `digitalWrite(i, desligar)`; será executado com `i = 0`; e o LED ligado na porta 0 se desliga.

No próximo passo do loop, o valor de `i` passa a 1 devido ao `i++` e como este número é menor ou igual a 13, o comando `digitalWrite(i, desligar)`; será executado com `i = 1`; e o LED ligado na porta 1 se desliga.

Isto continuará muito rapidamente. Digamos que `i` agora vale 11. Como 11 é menor ou igual a 13, então `digitalWrite(i, desligar)`; será executado com `i = 11`; e o LED ligado na porta 11 se desliga.

No próximo passo do loop, o valor de `i` passa a 12 devido ao `i++` e como este número é menor ou igual a 13, o comando `digitalWrite(i, desligar)`; será executado com `i = 12`; e o LED ligado na porta 12 se desliga.

No próximo passo do loop, o valor de `i` passa a 13 devido ao `i++` e como este número é menor ou igual a 13, o comando `digitalWrite(i, desligar)`; será executado com `i = 13`; e o LED ligado na porta 13 se desliga.

No próximo passo do loop, o valor de `i` passa a 14 devido ao `i++` e como este número é maior que 13, o comando

ROBÓTICA – 9 ANO – 05/05/2022

`digitalWrite(i, desligar)`; não será executado e o `for` se encerra.

A próxima parte é mais fácil: ligamos os LEDs corretamente para aparecer o número 8 no display.

```
digitalWrite(2, ligar);  
digitalWrite(3, ligar);  
digitalWrite(4, ligar);  
digitalWrite(5, ligar);  
digitalWrite(6, ligar);  
digitalWrite(7, ligar);  
digitalWrite(8, ligar);  
}
```

TAREFA DE PROGRAMAÇÃO

Faça um programa que exiba os números de zero até nove de um em um segundo.

Mesma coisa que na aula passada? Não: agora você vai usar o comando `for` e funções. Para te ajudar, o professor vai montar a função que aciona o número um. Veja abaixo:

```
void um() {  
  //primeiro desligamos tudo  
  for(int i=0; i<=13; i++){  
    digitalWrite(i, desligar);  
  }  
  digitalWrite(3, ligar);  
  digitalWrite(4, ligar);  
}
```

E o `void loop()` como fica? Vou te dar o programa inteiro, pois é fácil:

```
void loop() {  
  zero();  
  delay(1000);  
  um();  
  delay(1000);  
  dois();  
  delay(1000);  
  tres();  
  delay(1000);  
  quatro();  
  delay(1000);  
  cinco();  
  delay(1000);  
  seis();  
  delay(1000); sete();  
  oito();  
  delay(1000);  
  nove();  
  delay(1000);  
}
```

E se você conseguir terminar, coloque outro display e faça um programa que conte de 0 até 99.

Nota: você não precisará montar 100 funções, certo? Você vai precisar montar uma função para cada unidade e uma para cada dezena, ou seja, no total de 20 funções. Mas não é tão difícil quanto parece.