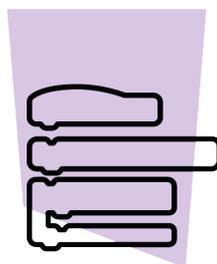
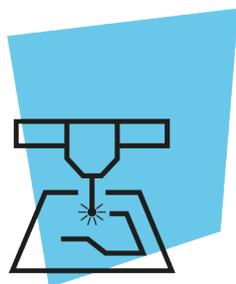
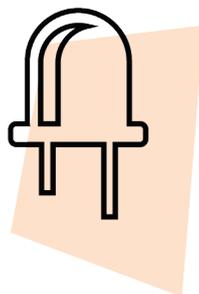
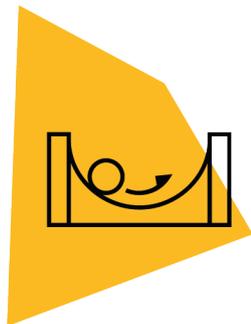
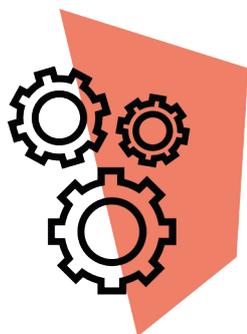


ROBÓTICA

Módulo 1



Display 7 Segmentos

AULA 16

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Renato Feder

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Andre Gustavo Souza Garbosa

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Cleiton Rosa

Michelle Santos

Simone Sinara de Souza

Revisão Textual

Adilson Carlos Batista

Leitura Crítica e Normalização Bibliográfica

Ricardo Hasper

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Ilustração

Jocelin Vianna (Educa Play)

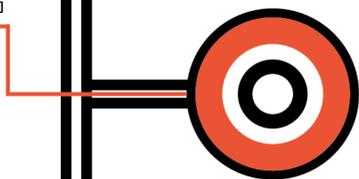
2021

- Aula 01 Por Que Robótica?
- Aula 02 Tensão, Corrente e Resistência
- Aula 03 Kit de Robótica
- Aula 04 Arduino Uno R3
- Aula 05 Softwares Arduino IDE e mBlock
- Aula 06 Portas Digitais
- Aula 07 Circuito Elétrico
- Aula 08 LED e Resistor
- Aula 09 Semáforo [Carros]
- Aula 10 Semáforo [Cruzamento Carros]
- Aula 11 Semáforo [Pedestres]
- Aula 12 Semáforo [Cruzamento Carros + Pedestres]
- Aula 13 Push Button
- Aula 14 Feedbacks + Inventário I
- Aula 15 Semáforo [Carros + Pedestres com Botão]
- Aula 16 Display 7 Segmentos
- Aula 17 Fonte DC + Plug P4
- Aula 18 Portas PWM
- Aula 19 LED Fade-In
- Aula 20 LED Fade-Out
- Aula 21 Super Máquina 80's
- Aula 22 Super Máquina 2008
- Aula 23 Potenciômetro
- Aula 24 Buzzer Passivo
- Aula 25 LED RGB
- Aula 26 Arco-Iris
- Aula 27 Sensor LDR
- Aula 28 Feedbacks + Inventário II
- Aula 29 Sensor de Temperatura
- Aula 30 Sensor de Obstáculo IR
- Aula 31 Controle Motor DC
- Aula 32 Kit Chassi 2WD Robô
- Aula 33 Seguidor de Linha
- Aula 34 Sensor de Distância
- Aula 35 Sensor de Estacionamento
- Aula 36 Display LCD 16x2
- Aula 37 Trena Digital
- Aula 38 Robô Sumô [Estrutura]
- Aula 39 Robô Sumô [Programação + Treinamento I]
- Aula 40 Robô Sumô [Programação + Treinamento II]
- Aula 41 Disputa de Sumôs
- Aula 42 Feedbacks + Inventário III

Aula 15
Semáforo
[Carros + Pedestres
com Botão]

Aula 16
Display
7 Segmentos

Aula 17
Fonte DC
+ Plug P4



Sumário

Introdução	2
Objetivo desta Aula	2
Competências Gerais Previstas na BNCC	3
Habilidades do Século XXI a Serem Desenvolvidas	4
Lista de Materiais	4
Roteiro da Aula	5
1. Contextualização	5
2. Montagem e Programação	6
3. Feedback e Finalização	17
Videotutorial	18
Referências	19

16 DISPLAY 7 SEGMENTOS



Introdução

Você, provavelmente, já deve ter visto uma cena de filme em que alguém precisa desativar uma bomba, a qual, geralmente, apresenta um componente eletrônico com contagem do tempo de forma regressiva. Este componente é o **Display 7 Segmentos**, mostrador muito utilizado em calculadoras digitais, medidores eletrônicos, relógios digitais, painéis de comunicação visual, rádios, entre outros. Nesta aula, você aprenderá a programar o funcionamento deste componente eletrônico e sua aplicabilidade em projetos voltados à robótica.



Objetivos desta Aula

- Programar um contador que varia de 0 a 9, através do dispositivo Display 7 segmentos.





Competências Gerais Previstas na BNCC

[CG02] - Exercitar a curiosidade intelectual e recorrer à abordagem própria das ciências, incluindo a investigação, a reflexão, a análise crítica, a imaginação e a criatividade, para investigar causas, elaborar e testar hipóteses, formular e resolver problemas e criar soluções (inclusive tecnológicas) com base nos conhecimentos das diferentes áreas.

[CG04] - Utilizar diferentes linguagens – verbal (oral ou visual-motora, como Libras, e escrita), corporal, visual, sonora e digital –, bem como conhecimentos das linguagens artística, matemática e científica, para se expressar e partilhar informações, experiências, ideias e sentimentos em diferentes contextos e produzir sentidos que levem ao entendimento mútuo.

[CG05] - Compreender, utilizar e criar tecnologias digitais de informação e comunicação de forma crítica, significativa, reflexiva e ética nas diversas práticas sociais (incluindo as escolares) para se comunicar, acessar e disseminar informações, produzir conhecimentos, resolver problemas e exercer protagonismo e autoria na vida pessoal e coletiva.

[CG09] - Exercitar a empatia, o diálogo, a resolução de conflitos e a cooperação, fazendo-se respeitar e promovendo o respeito ao outro e aos direitos humanos, com acolhimento e valorização da diversidade de indivíduos e de grupos sociais, seus saberes, identidades, culturas e potencialidades, sem preconceitos de qualquer natureza.

[CG10] - Agir pessoal e coletivamente com autonomia, responsabilidade, flexibilidade, resiliência e determinação, tomando decisões com base em princípios éticos, democráticos, inclusivos, sustentáveis e solidários.



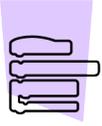
Habilidades do Século XXI a Serem Desenvolvidas

- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



Lista de Materiais

- 01 Placa Arduino;
- 01 Display 7 Segmentos Cátodo ou Ânodo Comum;
- 08 Resistores 220 Ω ;
- 01 Placa Protoboard;
- 09 Jumpers Macho-Macho;
- 01 Notebook;
- Software Arduino IDE;
- Software mBlock.

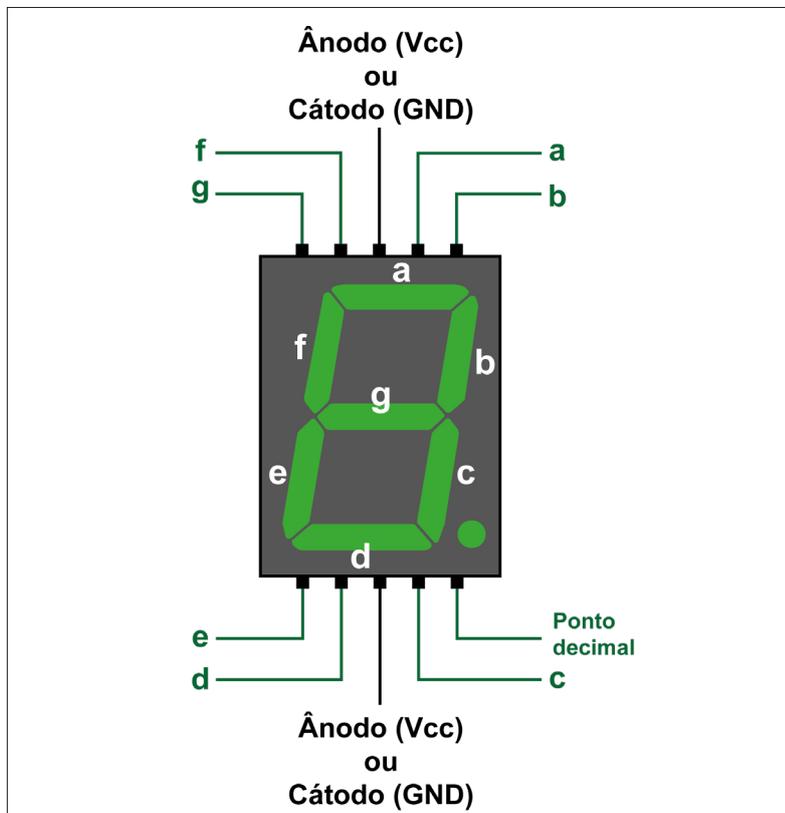


Roteiro da Aula

1. Contextualização:

O **Display 7 Segmentos** é um componente eletrônico formado por 7 LEDs, ou seja, cada LED forma um segmento do Display dispostos de maneira a formar números decimais, símbolos e/ou caracteres que possibilitam a comunicação visual. Os segmentos podem ser representados por letras do alfabeto e a Figura 1 apresenta o pino correspondente a cada um dos segmentos e o pino equivalente ao ponto decimal, localizado no canto inferior direito do Display. Os terminais centrais superior e inferior representam o eletrodo que pode ser ânodo (conexão VCC) ou cátodo (conexão GND).

Figura 1 - Display 7 Segmentos



Fonte: SEED/DTI/CTE, 2022

Conforme mencionamos anteriormente, de acordo com o eletrodo que possui, o Display 7 Segmentos pode ser classificado como:

a) **Display de 7 Segmentos ânodo comum (AC)**, quando os terminais ânodos (polos positivos) de todos os LEDs estão interligados entre si e, em um deles é ligado, através de um Resistor, uma tensão de 5V e adicionado uma tensão de 0V (GND) para que o Display funcione.

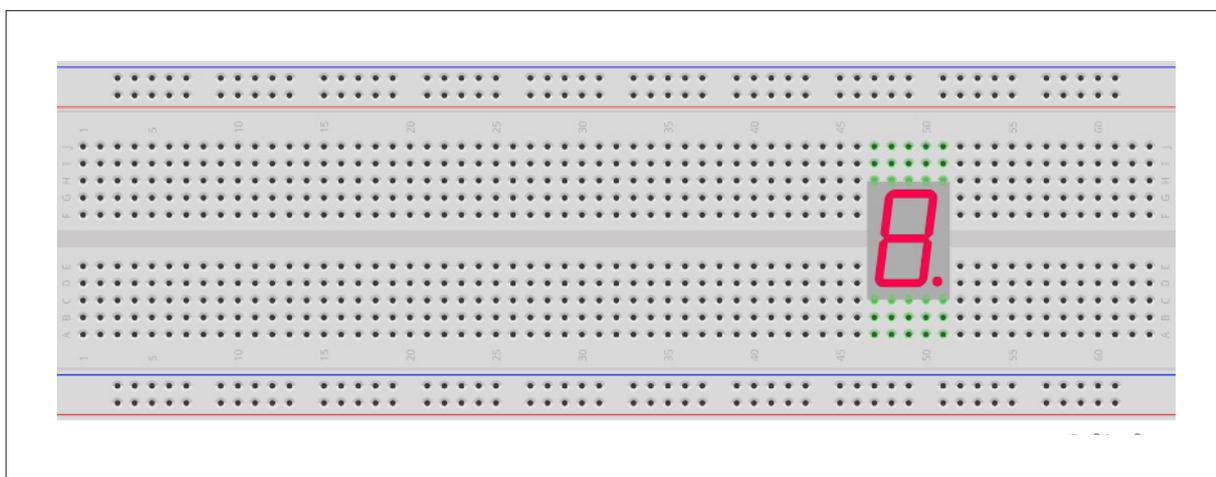
b) **Display de 7 Segmentos cátodo comum (CC)**, quando os terminais cátodos (polos negativos) de todos os segmentos estão interligados entre si e em um deles é ligado, através de um Resistor, uma tensão de 0V (GND) e adicionado uma tensão de 5V nos demais terminais. O Display funcionará de acordo com os segmentos que estão recebendo a tensão positiva.

Para programar um contador que varia de 0 a 9, utilizaremos, nesta aula, o Display de 7 Segmentos. A exibição de dígitos no Display ocorrerá a partir da programação de acendimento dos segmentos correspondentes.

2. Montagem e Programação:

O primeiro passo é a montagem dos componentes eletrônicos. Encaixe o Display 7 Segmentos na região central da Protoboard, como mostra a figura 2.

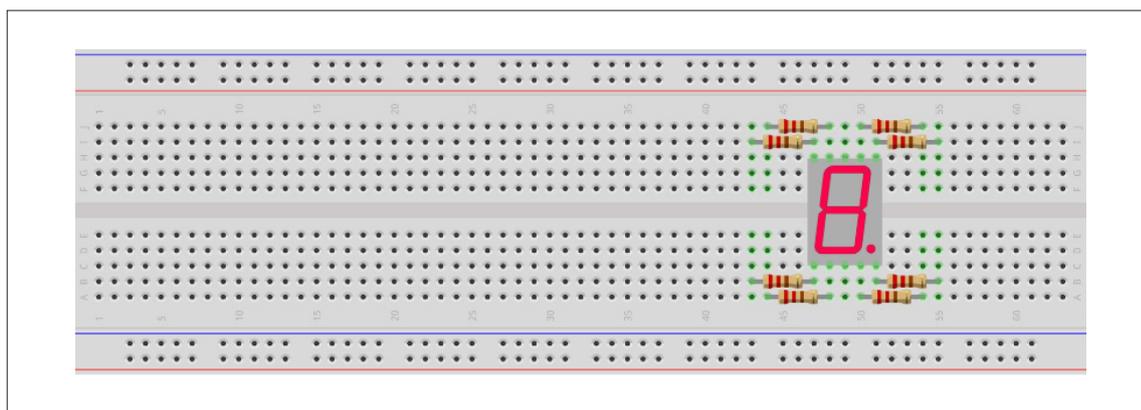
Figura 2 - Inserindo Display 7 segmentos na placa Protoboard



Fonte: Fritzing, 2022

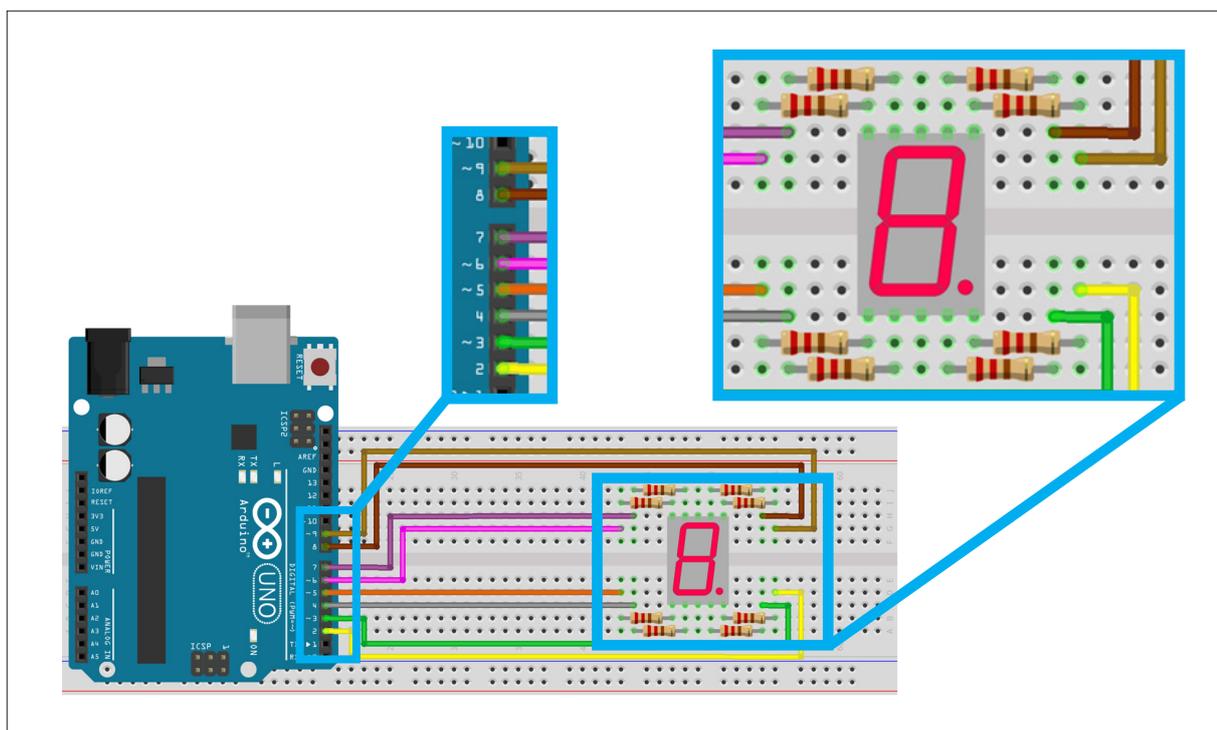
A seguir, insira na Protoboard os 8 Resistores, dispostos conforme mostra a figura 3. Os Resistores são utilizados para limitar a corrente elétrica no circuito, evitando uma possível queima do Display.

Figura 3 - Inserindo Resistores à placa Protoboard



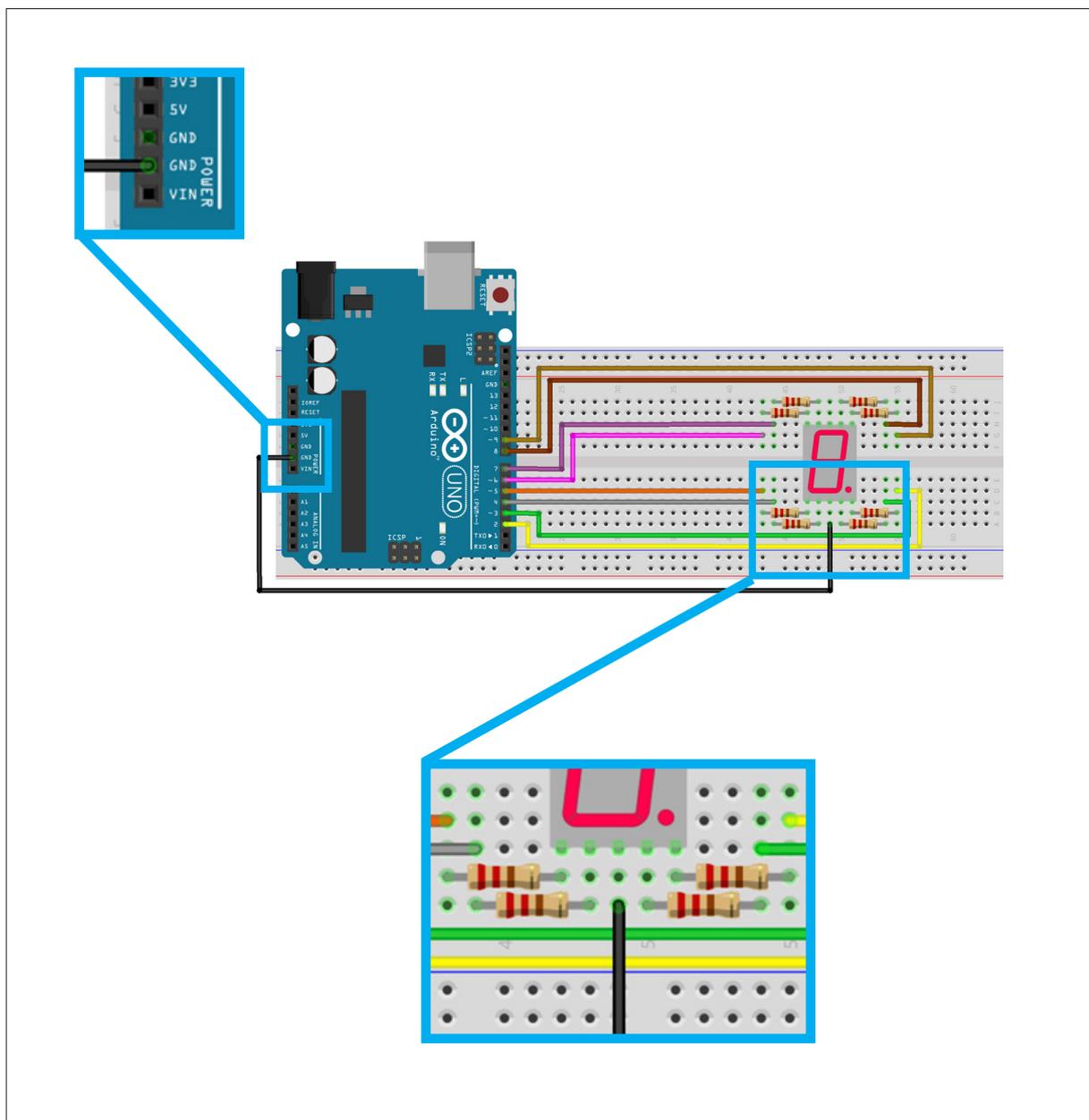
Utilize 8 Jumpers Macho-Macho para conectar os segmentos do Display às portas da placa Arduino, respeitando a sequência apresentada na figura 4.

Figura 4 - Conectando os segmentos do Display aos pinos do Arduino.



E se o seu Display é o modelo com Cátodo Comum, utilize um Jumper para interligar o pino **GND** do Arduino à coluna de furos da Protoboard, a mesma que se encontra o pino central do Display, conforme indicado na figura 6.

Figura 6 - Conectando o terminal central do Display ao Arduino



Fonte: Fritzing, 2022



Agora, vamos programar!

Com os componentes eletrônicos montados, vamos programar, por código e por blocos, o Display de 7 Segmentos cátodo comum ou ânodo comum para que atue como um contador representando valores de 0 a 9.

i. Linguagem de programação por código

Para iniciar a programação, conecte a placa Arduino ao computador, através de um cabo USB, para que ocorra a comunicação entre a placa microcontroladora e o software Arduino IDE.

No software IDE, crie um sketch e lembre-se de selecionar a porta que o computador atribuiu ao Arduino; então, escreva ou copie e cole o código-fonte de programação, conforme apresentado no quadro 1:

Atenção! Para o correto funcionamento de seu protótipo, você deverá descomentar as linhas 10 e 11 ou 14 e 15, correspondentes ao modelo do seu Display (Ânodo Comum ou Cátodo Comum).

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring)

```
/*
*****
/* Aula 16 - Display 7 Segmentos com Ânodo ou Ânodo Comum */
/* Programação de um contador com o display 7 segmentos. */
*****

/* ATENÇÃO! Descomente as linhas correspondente ao modelo */
/* de seu display. */

/* Descomente essas linhas para Display de Ânodo Comum. */
//bool liga = LOW;
//bool desliga = HIGH;

/* Descomente essas linhas para Display de Cátodo Comum. */
//bool liga = HIGH;
//bool desliga = LOW;
*/
```

```
/* Define os pinos do para cada segmento do display. */
#define a 8
#define b 9
#define c 3
#define d 4
#define e 5
#define f 7
#define g 6
#define ponto 2

void setup() {
  /* Define os pinos como saída. */
  pinMode(a, OUTPUT);
  pinMode(b, OUTPUT);
  pinMode(c, OUTPUT);
  pinMode(d, OUTPUT);
  pinMode(e, OUTPUT);
  pinMode(f, OUTPUT);
  pinMode(g, OUTPUT);
  pinMode(ponto, OUTPUT);
  /* Inicia com todos os segmentos desligados. */
  digitalWrite(a, desliga);
  digitalWrite(b, desliga);
  digitalWrite(c, desliga);
  digitalWrite(d, desliga);
  digitalWrite(e, desliga);
  digitalWrite(f, desliga);
  digitalWrite(g, desliga);
  digitalWrite(ponto, desliga);
}
void loop() {
  /* Loop para a passagem dos 10 dígitos (0-9). */
  for (int digito = 0; digito < 10; digito++)
  {
    /* Condição para ligar o segmento a. */
    /* Se o dígito não for 1 e 4, faça... */
    if (digito != 1 && digito != 4)
      digitalWrite(a, liga);
    /* Condição para ligar o segmento b. */
    /* Se o dígito não for 5 e 6, faça... */
    if (digito != 5 && digito != 6)
```

```
    digitalWrite(b, liga);
    /* Condição para ligar o segmento c.                */
    /* Se o dígito não for 2, faça...                  */
    if (dígito != 2)
        digitalWrite(c, liga);
    /* Condição para ligar o segmento d.                */
    /* Se o dígito não for 1, 4 e 7 faça...            */
    if (dígito != 1 && dígito != 4 && dígito != 7)
        digitalWrite(d, liga);
    /* Condição para ligar o segmento e                */
    /* Se o dígito for 2 ou 6 ou 8 ou 0, faça...        */
    if (dígito == 2 || dígito == 6 || dígito == 8 || dígito == 0)
        digitalWrite(e, liga);
    /* Condição para ligar o segmento f                */
    /* Se o dígito não for 1, 2, 3 e 7 faça...          */
    if (dígito != 1 && dígito != 2 && dígito != 3 && dígito != 7)
        digitalWrite(f, liga);
    /* Condição para ligar o segmento d.                */
    /* Se o dígito não for 0, 1 e 7 faça...            */
    if (dígito != 0 && dígito != 1 && dígito != 7)
        digitalWrite(g, liga);
    /* Liga e desliga de forma alternada o ponto.      */
    digitalWrite(ponto, !digitalRead(ponto));
    /* Aguarda 1 segundo para trocar de dígito.        */
    delay(1000);
    /* Apaga todos os segmentos para apresentar o próximo dígito */
    digitalWrite(a, desliga);
    digitalWrite(b, desliga);
    digitalWrite(c, desliga);
    digitalWrite(d, desliga);
    digitalWrite(e, desliga);
    digitalWrite(f, desliga);
    digitalWrite(g, desliga);
}
}
```

Atenção! Ao copiar o código diretamente do pdf, evite quebra da página (e conseqüentemente erros na compilação), copiando o código por partes.

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verificar** para analisar se não há erros de sintaxe. Estando o código correto, o próximo passo é realizar a transferência do programa para o Arduino. Pressione o botão **Carregar**, para realizar upload do programa para o Arduino.

Após a transferência do programa para o Arduino, os segmentos do Display acenderão, conforme comando estabelecidos na programação.

ii. Linguagem de programação por blocos

Outra forma de programar o funcionamento do Display 7 segmentos é por meio da linguagem de programação que utiliza blocos de funções prontas, os quais representam comandos de programação. Vamos utilizar o software mBlock.

Para conectar o mBlock ao Arduino, você deve clicar no ícone **Adicionar**, localizado no campo **Dispositivos**, e selecionar o Arduino, na biblioteca de dispositivos do mBlock, clicando, na sequência, no botão **OK**.

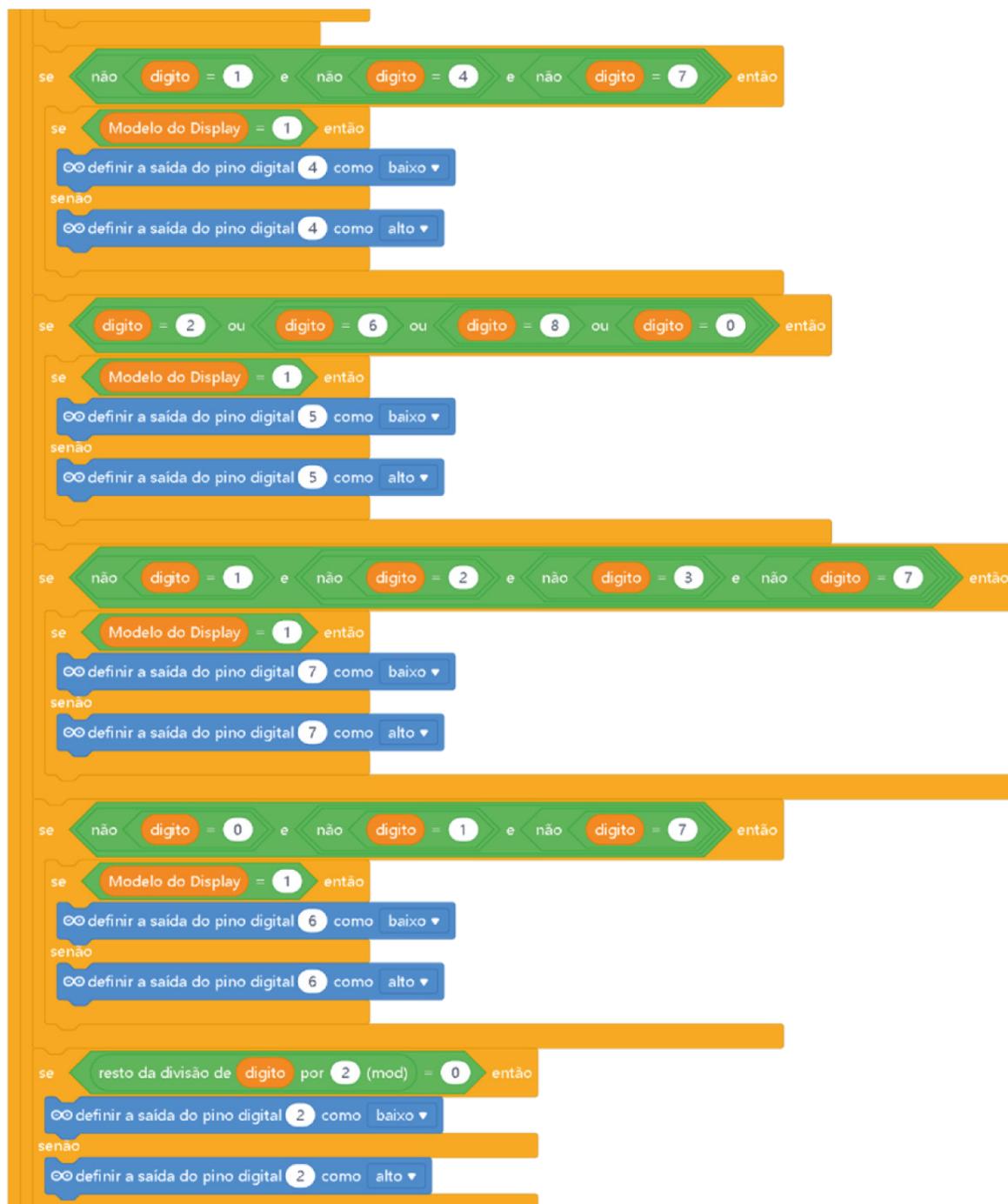
Nessa programação, utilizaremos variáveis que auxiliarão na estrutura do nosso programa (para recordar como criar uma variável, consulte a **Aula 05 - Softwares Arduino IDE e mBlock0**).

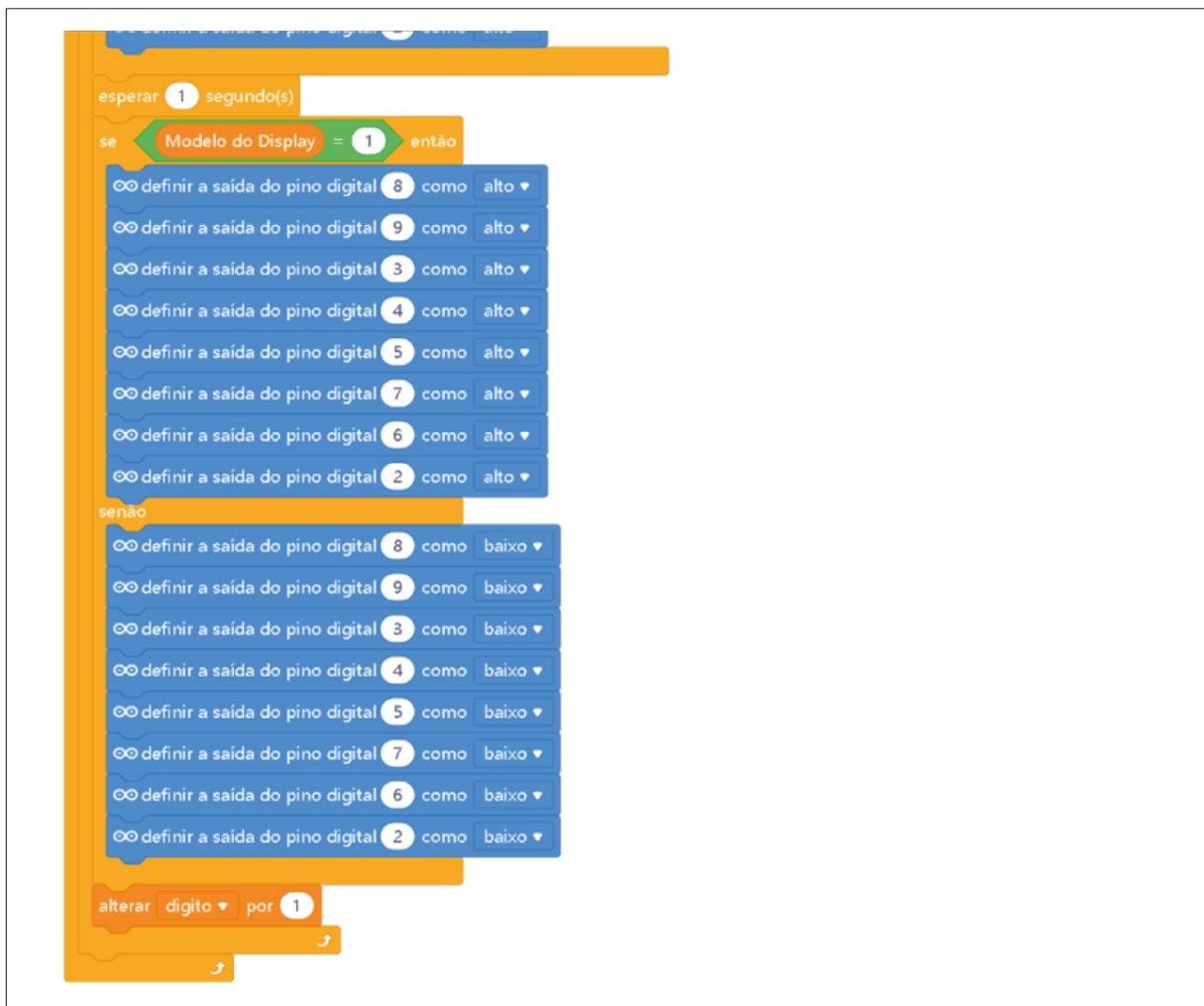
Monte os blocos, arrastando e soltando, de acordo com a programação do projeto, como mostra a figura 7.

Atenção! Para o correto funcionamento de seu protótipo, você deverá informar o número correspondente ao modelo do seu Display (1 - Ânodo Comum ou 2 - Cátodo Comum).

Figura 7 - Programação em blocos para funcionamento do Display

```
quando o Arduino Uno começar
definir Modelo do Display para 1
se Modelo do Display = 1 então
  definir a saída do pino digital 8 como alto
  definir a saída do pino digital 9 como alto
  definir a saída do pino digital 3 como alto
  definir a saída do pino digital 4 como alto
  definir a saída do pino digital 5 como alto
  definir a saída do pino digital 7 como alto
  definir a saída do pino digital 6 como alto
  definir a saída do pino digital 2 como alto
repetir para sempre
  definir digito para 0
  repita até digito > 9
  se não digito = 1 e não digito = 4 então
    se Modelo do Display = 1 então
      definir a saída do pino digital 8 como baixo
    senão
      definir a saída do pino digital 8 como alto
  se não digito = 5 e não digito = 6 então
    se Modelo do Display = 1 então
      definir a saída do pino digital 9 como baixo
    senão
      definir a saída do pino digital 9 como alto
  se não digito = 2 então
    se Modelo do Display = 1 então
      definir a saída do pino digital 3 como baixo
    senão
      definir a saída do pino digital 3 como alto
```





Assim que os blocos estiverem montados, clique no botão **Conectar** para iniciar a comunicação entre o software mBlock com a placa Arduino. Ao clicar sobre o botão **Conectar**, aparecerá um Tooltip solicitando a confirmação da conexão entre os dois dispositivos.

Uma vez realizada a conexão entre os dispositivos, será ativado, na interface do mBlock, o botão **Upload**, o qual ao ser clicado, o software verificará se não há erros na estrutura do programa e, então, compilará para enviar o programa à placa Arduino.

Com a transferência do código para o dispositivo Arduino Uno, inicia-se o funcionamento dos segmentos do Display, exibindo dígitos, conforme comandos estabelecidos na programação.



Desafios:

i. Acrescente no sketch as letras A, B, C, D, E e F. Se tiver dúvidas quanto a referência dos pinos do Display com os LEDs dos segmentos citados acima, retorne a figura 1, desta aula.

ii. Crie uma contagem regressiva de 9 a 0, invertendo o sentido do projeto original. Utilize o sistema de funções criado no desafio anterior. Dica: crie no "Void loop()" um "for(int = 9; i >= 0; i--)" para criar uma contagem regressiva de 9 a 0 repetidamente.



E se... ?

i. O Display não acender, verifique todas as conexões dos segmentos e se todos os LEDs estão funcionando.

3. Feedback e Finalização:

a. Confira, compartilhando seu projeto com os demais colegas, se o objetivo foi alcançado.

b. Analise seu projeto desenvolvido, de modo a atender aos requisitos para funcionamento do Display 7 segmentos.

c. Reflita se as seguintes situações ocorreram:

i. Colaboração e Cooperação: você e os membros de sua equipe interagiram entre si, compartilhando ideias que promoveram a aprendizagem e o desenvolvimento deste projeto?

ii. Pensamento Crítico e Resolução de Problemas: você conseguiu identificar os problemas, analisar informações e tomar decisões de modo a contribuir para o projeto desenvolvido?

d. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.

AULA

16



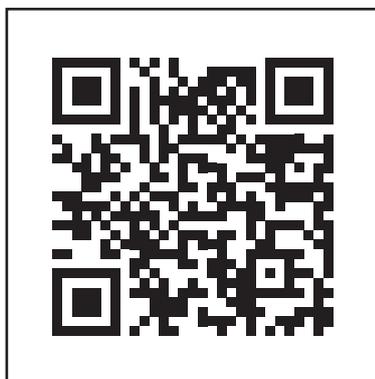
Videotutorial

Com o intuito de auxiliar na montagem e na programação desta aula, apresentamos um videotutorial, disponível em:



<https://rebrand.ly/a16robotica>

Acesse, também, pelo QRCode:



18

Robótica



Referências

ARDUINO. Site oficial. **Ambiente de Programação do Arduino**. Disponível em: <https://create.arduino.cc/editor>. Acesso em: 15 out. 2021.

ARDUINO. Site oficial. **Downloads**. Disponível em: www.arduino.cc/en/Main/Software. Acesso em: 15 out. 2021.

BRASIL. Ministério da Educação. **Base Nacional Comum Curricular**. Brasília, 2018. Disponível em: http://basenacionalcomum.mec.gov.br/images/BNCC_EI_EF_110518_versaofinal_site.pdf. Acesso em: 18 out. 2021.

MAKEBLOCK. mBlock. **Download mBlock**. Disponível em: <https://mblock.makeblock.com/en-us/download/>. Acesso em: 15 out. 2021.

MAKEBLOCK. mBlock. **Programação em blocos**. Disponível em: <https://ide.mblock.cc/>. Acesso em: 15 out. 2021.

DIRETORIA DE TECNOLOGIAS E INOVAÇÃO (DTI)
COORDENAÇÃO DE TECNOLOGIAS EDUCACIONAIS (CTE)

EQUIPE ROBÓTICA PARANÁ

Adilson Carlos Batista
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edna do Rocio Becker
Marcelo Gasparin
Michelle dos Santos
Ricardo Hasper
Roberto Carlos Rodrigues
Simone Sinara de Souza

Os materiais, aulas e projetos da “Robótica Paraná”, foram produzidos pela Coordenação de Tecnologias Educacionais (CTE), da Diretoria de Tecnologia e Inovação (DTI), da Secretaria de Estado da Educação e do Esporte do Paraná (Seed), com o objetivo de subsidiar as práticas docentes com os estudantes por meio da Robótica.

Este material foi produzido para uso didático-pedagógico exclusivo em sala de aula.



Este trabalho está licenciado com uma Licença
Creative Commons – CC BY-NC-SA
[Atribuição - NãoComercial - Compartilhalgual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

