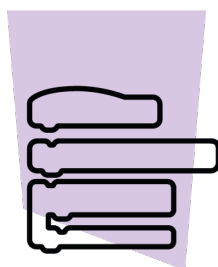
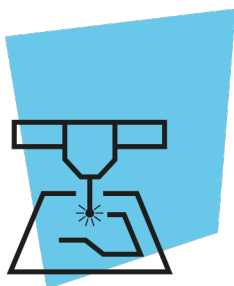
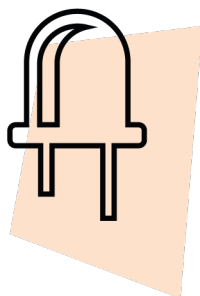
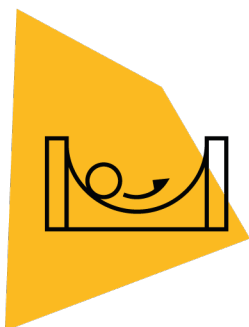
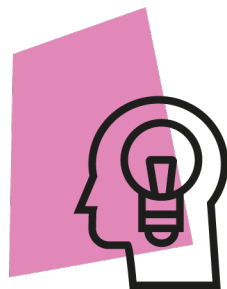


ROBÓTICA

Módulo 2



Semáforo completo com
display de 7 segmentos
com 1 dígito

AULA 05

GOVERNADOR DO ESTADO DO PARANÁ

Carlos Massa Ratinho Júnior

SECRETÁRIO DE ESTADO DA EDUCAÇÃO

Roni Miranda Vieira

DIRETOR DE TECNOLOGIA E INOVAÇÃO

Claudio Aparecido de Oliveira

COORDENADOR DE TECNOLOGIAS EDUCACIONAIS

Marcelo Gasparin

Produção de Conteúdo

Darice Alessandra Deckmann Zanardini

Simone Sinara de Souza

Validação de Conteúdo

Cleiton Rosa

Revisão Textual

Orlando de Macedo Junior

Projeto Gráfico e Diagramação

Edna do Rocio Becker

Modelagem Blender

Cleiton Rosa

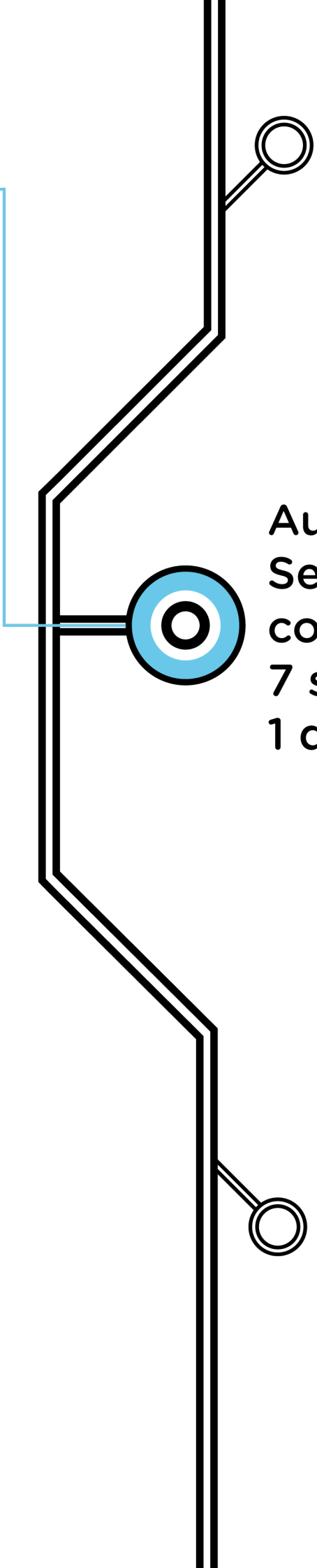
2023

- Aula 01 O que já vimos?
- Aula 02 Arduino: Bibliotecas e Funções
- Aula 03 Código Morse
- Aula 04 Semáforo Inteligente com IR
- Aula 05 Semáforo Completo com Display**
- Aula 06 Matriz de LED 8X8
- Aula 07 Desenhando na matriz de LEDs
- Aula 08 Painel de Senhas
- Aula 09 Escrevendo mensagens
- Aula 10 Robô Autônomo
- Aula 11 Sensor de Chuva
- Aula 12 Sensor de Umidade do Solo
- Aula 13 Irrigador Automático
- Aula 14 Feedbacks + Inventário I
- Aula 15 Teclado Matricial de Membrana
- Aula 16 Servos Motores
- Aula 17 Fechadura Eletrônica
- Aula 18 Controlando Servos Motores
- Aula 19 JoyStick Shield
- Aula 20 Braço Robótico
- Aula 21 Sensor de Movimento Presença
- Aula 22 Sensor de Som
- Aula 23 Sensor de Umidade e Temperatura
- Aula 24 Termômetro Digital
- Aula 25 Sensor de Gás e Fumaça
- Aula 26 Acelerômetro e Giroscópio
- Aula 27 Motor de Passo
- Aula 28 Feedbacks + Inventário II
- Aula 29 Receptor IR e Controle Remoto
- Aula 30 Relé
- Aula 31 Módulo RF 433mhz - I
- Aula 32 Módulo RF 433mhz - II
- Aula 33 Projeto CHAT via RF
- Aula 34 Módulo Wireless
- Aula 35 Comunicação do Módulo WI-FI em HTML
- Aula 36 Módulo WI-FI - IoT com Sensores
- Aula 37 Módulo WI-FI - IoT com Atuadores (LED)
- Aula 38 Módulo WI-FI - IoT com Atuadores (Relé)
- Aula 39 Monitor de Sensores em HTML I
- Aula 40 Monitor de Sensores em HTML II
- Aula 41 Mostra de Robótica
- Aula 42 Feedbacks + Inventário III

Aula 04
Semáforo Inteligente com IR

Aula 05
Semáforo completo com display com 7 segmentos com 1 dígito

Aula 06
Matriz de LED 8X8



Sumário

Introdução	2
Objetivos 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	7
3. Feedback e finalização	18
Videotutorial	19



Introdução

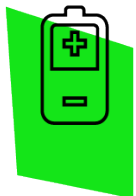
Em aulas anteriores da nossa jornada pela Robótica, vimos que os semáforos são dispositivos de sinalização formados por sistemas de luzes com cores (vermelho, amarelo e verde) instalados em ruas e avenidas de centros urbanos para auxiliar a travessia de veículos e pedestres, e aprendemos sobre o funcionamento e a sincronização deste equipamento.

Nesta aula, vamos simular, via programação, um semáforo que informe ao pedestre, utilizando um painel com contagem regressiva, o tempo exato para atravessar na faixa de segurança antes do fechamento do semáforo.



Objetivos desta aula

- Simular o funcionamento dos semáforos de veículos e pedestres;
- Utilizar funções para programar a abertura e o fechamento dos semáforos;
- Controlar a abertura do semáforo de pedestres através do acionamento de um *push button* (chave tátil);
- Programar a contagem regressiva para o fechamento do semáforo de pedestres.



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

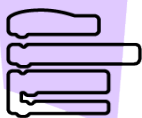
- Pensamento crítico;
- Afinidade digital;
- Resiliência;
- Resolução de problemas;
- Colaboração;
- Comunicação.



Lista de materiais

- 1 placa protoboard;
- 1 placa Arduino Uno R3;
- 1 LED amarelo 5mm;
- 2 LEDs verde 5mm;
- 2 LEDs vermelho 5mm;
- 12 resistores 220 ohms;
- 1 resistor 10 quilo-ohms;
- 24 jumpers macho-macho;
- 1 display 7 segmentos;
- 1 *push button*;
- 1 cabo USB;
- 1 computador;
- Software Arduino IDE.





Roteiro da aula

1. Contextualização

Os acidentes de trânsito são uma das principais causas de óbitos no Brasil e uma das inúmeras situações que levam os engenheiros de tráfego terrestre a elaborar ações que possam minimizar problemas de conflitos no trânsito. Uma delas é a instalação de semáforos em ruas e avenidas de centros urbanos, com o objetivo de informar aos usuários (motoristas, pedestres, ciclistas, entre outros) o momento permitido à circulação, à espera e à transição entre uma ou outra circunstância. Quando sincronizados, reduzem a formação de congestionamentos e beneficiam a travessia de pedestres e ciclistas em vias urbanas.

Além da nossa última aula, na qual prototipamos um semáforo inteligente com infravermelho, no Módulo 1 as aulas **9, 10, 11** e **12** também apresentaram conteúdos voltados aos semáforos de veículos, pedestres ou ambos, com as quais você pôde entender o funcionamento e sincronização

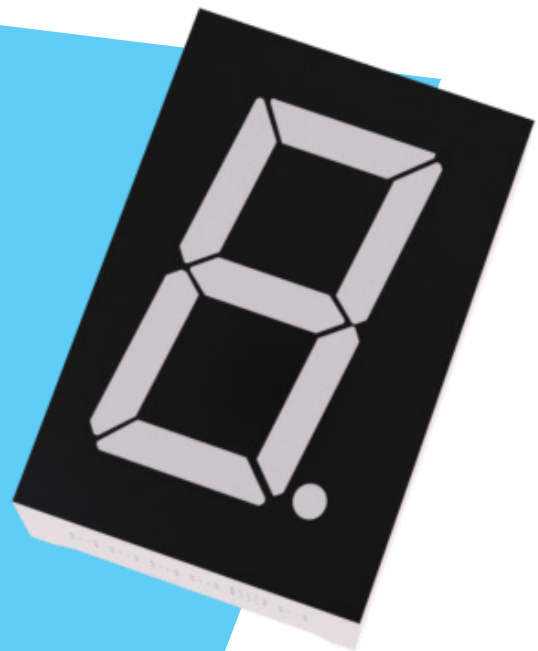
deles e entre eles. E na **Aula 15 - Semáforo [Carros + Pedestres com Botão]** do módulo anterior, você inseriu, na programação do semáforo de pedestres, o componente eletrônico *push button* com intenção de simular o botão que o pedestre pressiona para reduzir o tempo de espera do fechamento do semáforo de veículos, permitindo sua travessia. Para recordar estes funcionamentos e programações, retorne às aulas citadas.



Nesta aula, você aprenderá a programar, com o uso de funções, outra versão de semáforo completo, a qual adiciona o componente eletrônico display 7 segmentos com um dígito (figura 1). Na **Aula 16 - Display 7 segmentos com um dígito**, comentamos que este componente eletrônico é formado por 7 LEDs, com cada LED correspondendo a um segmento do display, dispostos de maneira a formar números decimais, símbolos e/ou caracteres que possibilitam a comunicação visual. Ainda tem mais um LED correspondente ao ponto, destinado à representação da posição do display para, por exemplo diferenciarmos entre os algarismos 6 e 9.

No semáforo completo a função do display é informar ao pedestre, com a contagem regressiva, o tempo que ele possui para atravessar a rua/avenida com segurança, antes de o semáforo de veículos abrir.

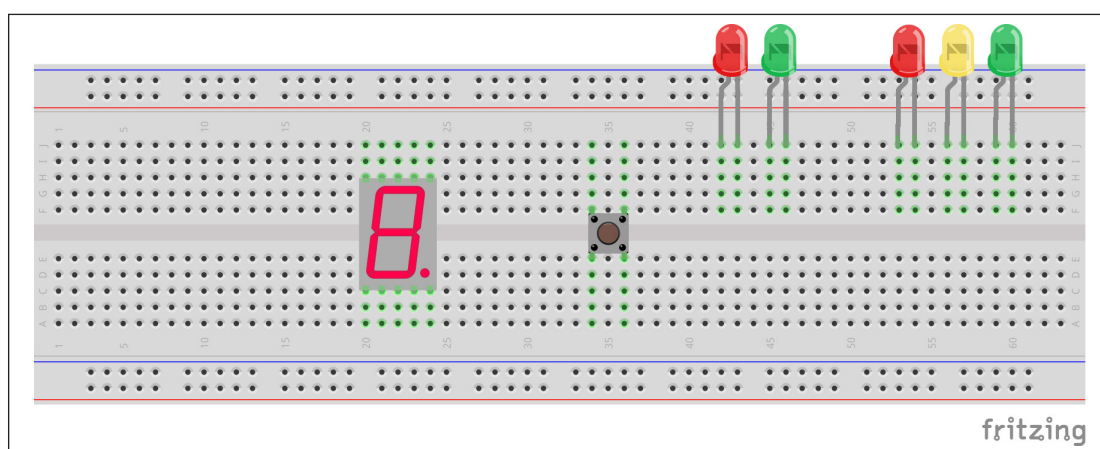
Figura 1 - Display 7 segmentos com um dígito



2. Montagem e programação

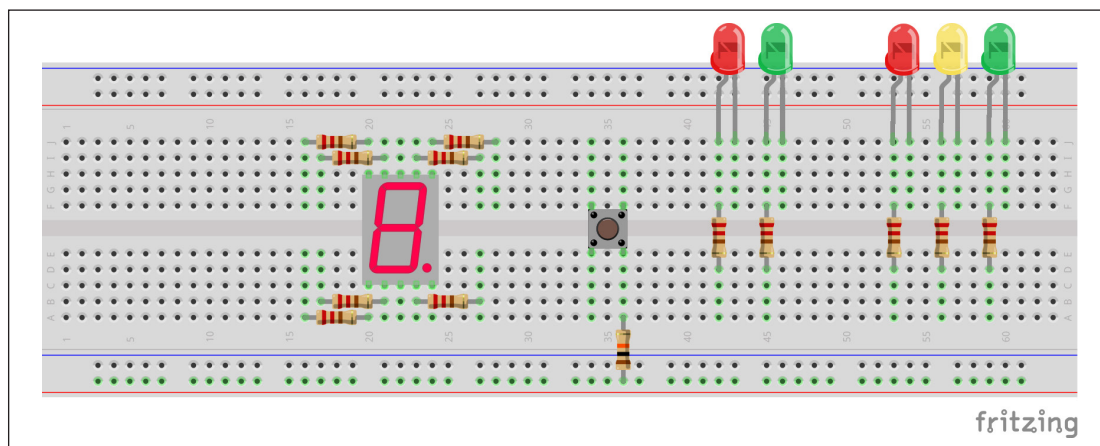
Inicie a montagem dos componentes eletrônicos encaixando, na região central da protoboard, o display 7 segmentos com um dígito e o *push button*. A seguir, insira os LEDs conforme as posições indicadas, observando a polaridade de cada LED (figura 2).

Figura 2 - Inserindo os primeiros componentes eletrônicos na protoboard



Conecte 7 resistores de 220 ohms aos terminais do display de 7 segmentos com um dígito e outros 5 resistores aos terminais positivos dos LEDs, conectando também o resistor de 10 quilo-ohms entre um terminal do *push button* e a linha lateral vermelha da protoboard (figura 3).

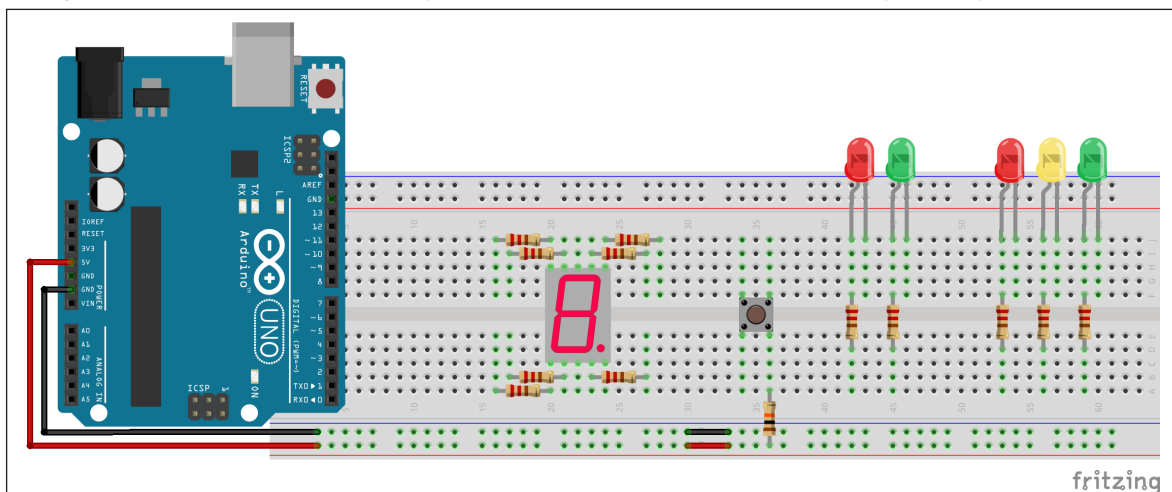
Figura 3 - Conectando resistores na protoboard



SEMÁFORO COMPLETO COM DISPLAY DE 7 SEGMENTOS COM 1 DÍGITO

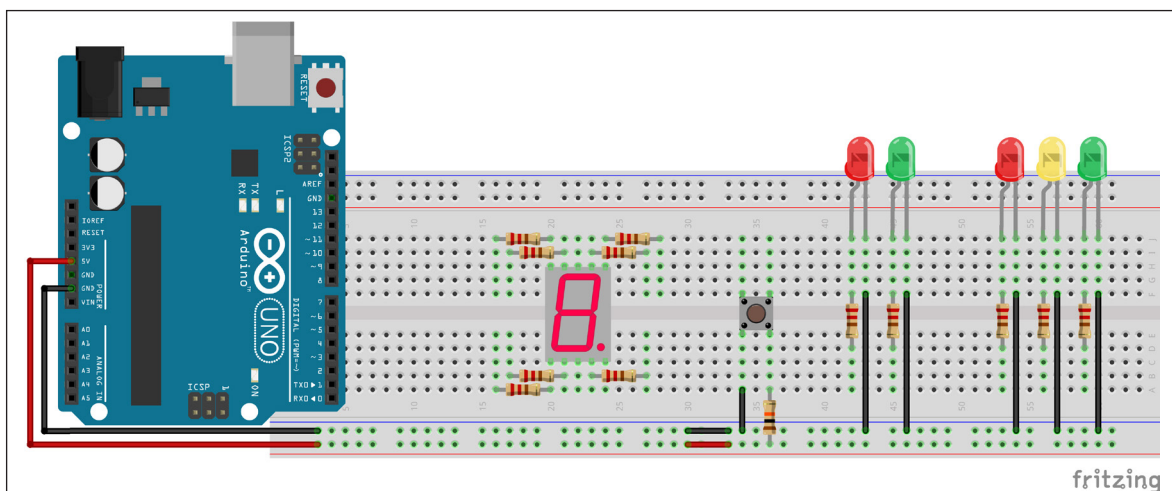
Ligue, com o uso de jumpers, o pino **GND** da placa Arduino a um dos furos da linha lateral **azul** da protoboard e o pino **5V** do Arduino a um dos furos da linha lateral **vermelha** da protoboard, interligando, também com jumpers, cada uma destas linhas na região central da protoboard (figura 4).

Figura 4 - Conectando a placa Arduino à extensão da placa protoboard



Utilizando 6 jumpers, interligue a linha lateral **azul** da protoboard ao terminal da **esquerda** do *push button* e aos terminais **negativos** dos LEDs (figura 5).

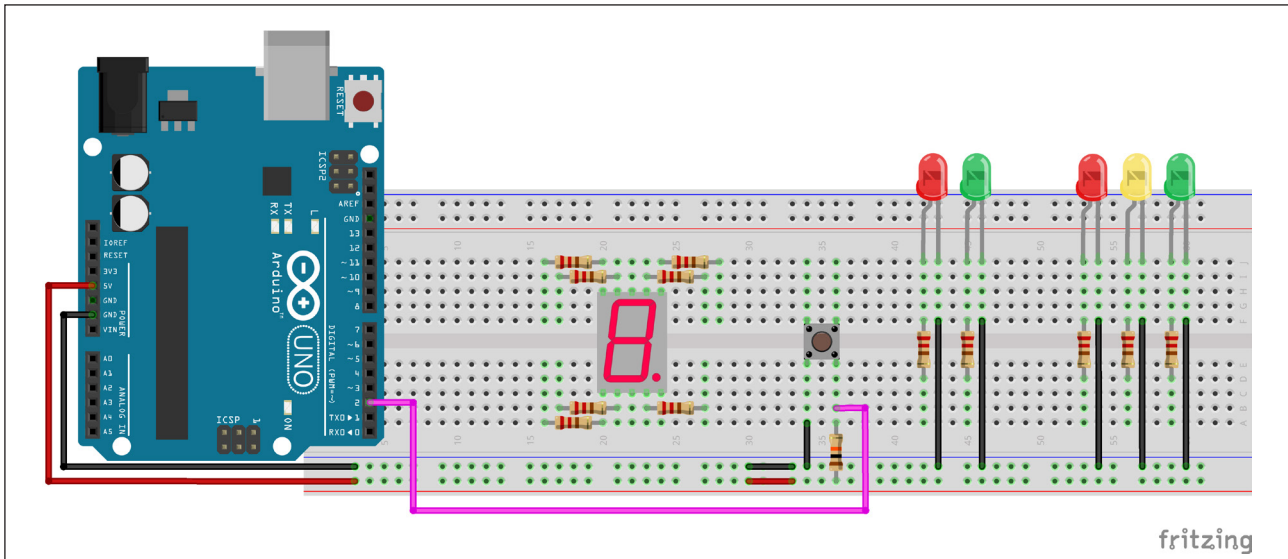
Figura 5 - Interligando jumpers aos terminais negativos do *push button* e dos LEDs



SEMÁFORO COMPLETO COM DISPLAY DE 7 SEGMENTOS COM 1 DÍGITO

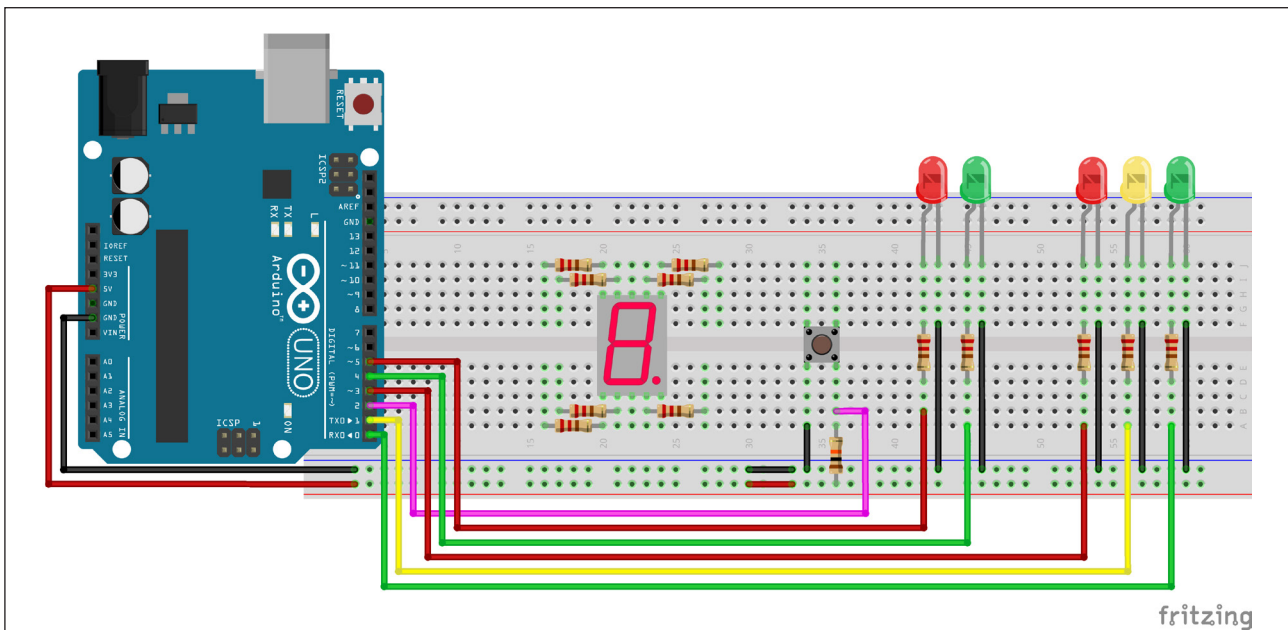
Com auxílio de um jumper, conecte a porta digital **2** do Arduino à coluna que liga o *push button* e seu respectivo resistor (figura 6).

Figura 6 - Ligando o resistor do *push button* à porta 2 do Arduino



Com outros jumpers, interligue as portas digitais **0, 1, 3, 4 e 5** do Arduino aos terminais dos resistores ligados aos LEDs (figura 7).

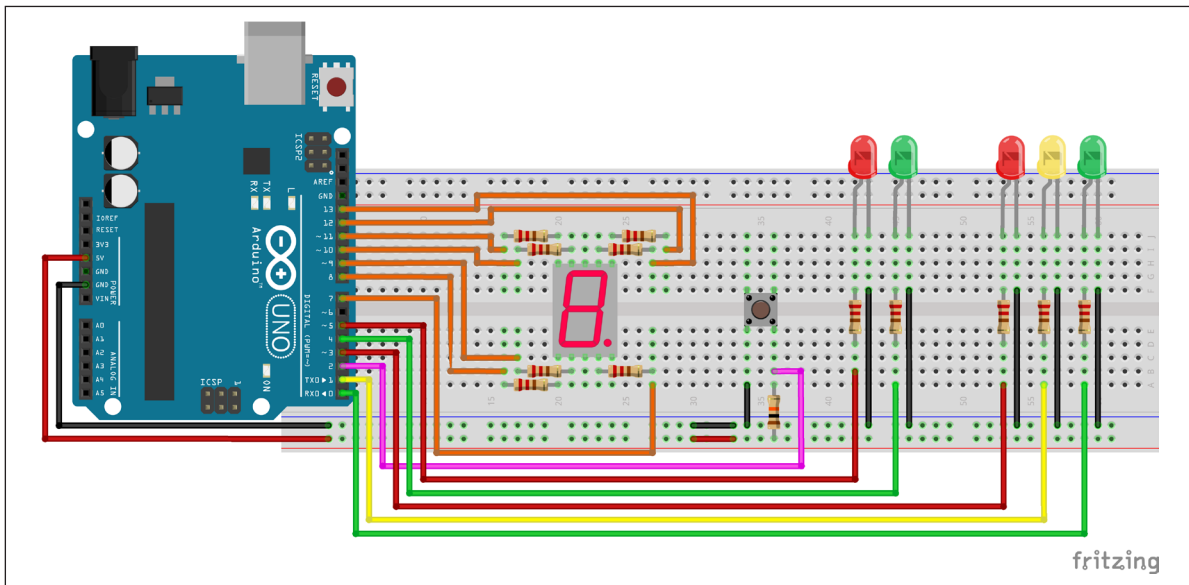
Figura 7 - Conectando o Arduino aos resistores ligados aos LEDs



SEMÁFORO COMPLETO COM DISPLAY DE 7 SEGMENTOS COM 1 DÍGITO

Interligue também, utilizando mais 7 jumpers, as portas digitais **7, 8, 9, 10, 11, 12 e 13** do Arduino aos resistores ligados ao display (figura 8).

Figura 8 - Conexão entre Arduino e resistores ligados ao display



Para finalizar a montagem, identifique se o display de 7 segmentos com um dígito que você está utilizando em seu protótipo é do modelo **ânodo comum** ou **cátodo comum**, verificando o seu código impresso na lateral da base do display. Essa informação é necessária tanto para se realizar a última conexão quanto para adequar o código da programação.

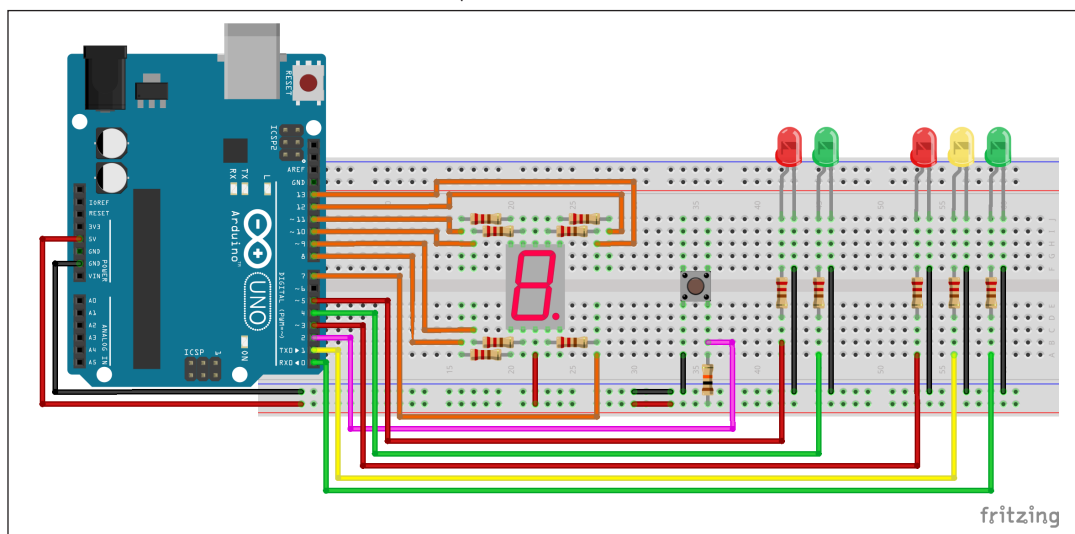
Modelos de display 7 segmentos com um dígito

Ânodo comum	Cátodo comum
xxxx B x	xxxx A x
Conexão VCC 5V	Conexão GND

SEMÁFORO COMPLETO COM DISPLAY DE 7 SEGMENTOS COM 1 DÍGITO

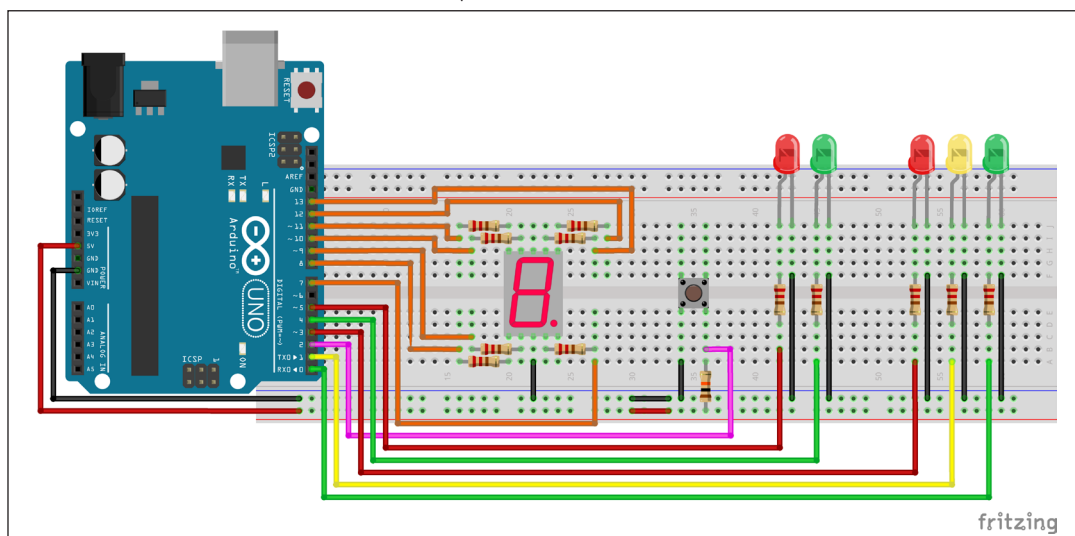
Se seu display é modelo **ânodo comum** (código xxxx**B**x), finalize a montagem conectando, com o uso de um jumper, o pino central do display à linha **vermelha** da protoboard (figura 9).

Figura 9 - Conexão do pino central do display modelo ânodo comum à protoboard



Mas se seu display é modelo **cátodo comum** (código xxxx**A**x), finalize a montagem conectando, usando um jumper, o pino central do display à linha lateral **azul** da protoboard (figura 10).

Figura 10 - Conexão do pino central do display de cátodo comum à protoboard



SEMÁFORO COMPLETO COM DISPLAY DE 7 SEGMENTOS COM 1 DÍGITO

Com os componentes eletrônicos montados, considerando o display 7 segmentos com um dígito disponível no kit de robótica da sua escola, o próximo passo desta aula é a programação por codificação.

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

No software IDE ou no Arduino Editor Online, escreva ou copie e cole o código-fonte de programação para o **display 7 segmentos com um dígito** (quadro 1).

Quadro 1 - Código-fonte da programação na linguagem do Arduino (Wiring) para display 7 segmentos com um dígito

```
/* *****  
/* Aula 05 - Semáforo Completo com Display. */  
/* Programação do semáforo completo com display de 7 segmentos */  
/* com um dígito. Seu funcionamento consiste em alterar seu */  
/* modo de operação assim que o botão for pressionado e, */  
/* durante o período de travessia, o display irá iniciar uma */  
/* contagem regressiva, informando */  
/* o tempo restante. Neste código, */  
/* utilizamos a função de interrupção (attachInterrupt). */  
/* *****  
/* ATENÇÃO! De acordo com o modelo de seu Display 7 Segmentos, */  
/* descomente uma das duas linhas abaixo. */  
  
//String modelo = "ânodo comum"; /* Display de Ânodo Comum. */  
//String modelo = "cátodo comum"; /* Display de Cátodo Comum. */  
  
/* Definições das portas utilizadas. */  
#define botao 2  
#define LED_VERDE_CAR 0  
#define LED_AMARELO_CAR 1  
#define LED_VERMELHO_CAR 3  
#define LED_VERDE_PED 4  
#define LED_VERMELHO_PED 5
```

```

/*
  Variável que seleciona o modo de operação do semáforo
  0 - Modo aberto para carros;
  1 - Modo aberto para pedestres;
*/
int modo = 0;

/* Variável matricial que armazena os estados de cada segmento*/
/* do display para cada dígito. */
/*
      --- 13 ----
      |           |
      10          12
      |           |
      ---- 11 ----
      |           |
      8           7
      |           |
      ---- 9 ----
{c, e, d, f, g, b, a} Segmentos
{7, 8, 9, 10, 11, 12, 13} Portas
*/
bool displaySeteSeg[10][7] = {
  { 1, 1, 1, 1, 0, 1, 1 }, //DIGITO 0
  { 1, 0, 0, 0, 0, 1, 0 }, //DIGITO 1
  { 0, 1, 1, 0, 1, 1, 1 }, //DIGITO 2
  { 1, 0, 1, 0, 1, 1, 1 }, //DIGITO 3
  { 1, 0, 0, 1, 1, 1, 0 }, //DIGITO 4
  { 1, 0, 1, 1, 1, 0, 1 }, //DIGITO 5
  { 1, 1, 1, 1, 1, 0, 1 }, //DIGITO 6
  { 1, 0, 0, 0, 0, 1, 1 }, //DIGITO 7
  { 1, 1, 1, 1, 1, 1, 1 }, //DIGITO 8
  { 1, 0, 1, 1, 1, 1, 1 } //DIGITO 9
};

void setup() {
  /* Loop para configurar os pinos como saída. */
  for (int i = 0; i <= 13; i++) {
    pinMode(i, OUTPUT);
  }
}

```



```
/* Configuramos o pino do botão como entrada. */
pinMode(botao, INPUT);

/* Ativa a função de interrupção no pino que o botão está
/* conectado. */
attachInterrupt(digitalPinToInterrupt(botao), botaoAcionado, FALLING);

/* Comando para o Display de Ânodo Comum iniciar desligado. */
if (modelo == "ânodo comum") {
  for (int i = 7; i <= 13; i++) {
    digitalWrite(i, HIGH);
  }
}
}

void loop() {
  switch (modo) {
    case 0:
/* Caso o botão não for pressionado mantém o semáforo aberto */
/* para os carros chamando a função "carros()". */
    carros();
    break;
    case 1:
/* Caso o botão seja pressionado, o semáforo entrará no modo */
/* de travessia chamando a função "travessia()". */
    travessia();
    modo = 0;
    break;
  }
}

/* Após pressionar o botão, será executada essa função, */
/* selecionando o modo de operação do semáforo. */
void botaoAcionado() {
  modo = 1;
}
```

```
/* Função que mantém o semáforo aberto para os carros. */
void carros() {
    digitalWrite(LED_VERDE_CAR, HIGH);
    digitalWrite(LED_VERMELHO_PED, HIGH);
    digitalWrite(LED_VERMELHO_CAR, LOW);

/* Desliga o display. */
    if (modelo == "ânodo comum") {
        for (int i = 7; i <= 13; i++) {
            digitalWrite(i, HIGH);
        }
    } else {
        for (int i = 7; i <= 13; i++) {
            digitalWrite(i, LOW);
        }
    }
}

/* Função que controla o semáforo para a travessia de */
/* pedestre. */
void travessia() {
/* Aguarda 1 segundo para entrar no modo travessia. */
    delay(1000);
/* Aciona o amarelo do semáforo dos carros. */
    digitalWrite(LED_VERDE_CAR, LOW);
    digitalWrite(LED_AMARELO_CAR, HIGH);
    delay(3000);
/* Aciona vermelho para os carros e verde para pedestre. */
    digitalWrite(LED_AMARELO_CAR, LOW);
    digitalWrite(LED_VERMELHO_CAR, HIGH);
    digitalWrite(LED_VERDE_PED, HIGH);
    digitalWrite(LED_VERMELHO_PED, LOW);

/* Inicia a contagem de 9 a 0 para a travessia. */
    boolean led = HIGH;
    for (int contador = 9; contador >= 0; contador--) {
/* Chama a função para acionar o display. */
```

```
ligaSegmentosDisplay(contador);

if (contador <= 6) {
    digitalWrite(LED_VERDE_PED, LOW);
    digitalWrite(LED_VERMELHO_PED, led);
    led = !led;
}
}
}

/* Função para acionar o display com o dígito correspondente. */
void ligaSegmentosDisplay(int digito) {
    byte pino = 7;
    for (byte contadorSegmentos = 0; contadorSegmentos <= 6; ++contadorSegmentos) {
        if (modelo == "ânodo comum") {
            // displaySeteSeg[linha][coluna]
            digitalWrite(pino, !displaySeteSeg[digito][contadorSegmentos]);
        } else {
            digitalWrite(pino, displaySeteSeg[digito][contadorSegmentos]);
        }
    }
    /* Incrementa o valor do pino (vai de 7 a 13). */
    pino++;
}
/* Aguarda 1 segundo para reiniciar a contagem. */
delay(1000);
}
```

Com o código-fonte inserido no Arduino IDE, compile o programa pressionando o botão **Verificar** para verificar se não há erros de sintaxe. Estando o código correto, pressione o botão **Carregar** para realizar a transferência do programa para o Arduino.

Após a transferência do programa para o Arduino, o semáforo permanecerá aberto para a passagem dos carros até o botão ser pressionado. Pressionado o botão, após 1 segundo, o semáforo entrará no modo travessia de pedestres e o Display iniciará a contagem regressiva avisando o pedestre o tempo exato para travessia antes do fechamento do semáforo.



Desafio:

Que tal adicionar, ao protótipo, um *buzzer* para alertar ao pedestre, via sonorização, o tempo disponível para travessia antes do fechamento do semáforo?



E se...?

O projeto não funcionar - atente para alguns possíveis erros:

- a. Verifique se os jumpers estão na mesma coluna dos terminais dos componentes, fazendo assim a conexão;
- b. Verifique se os jumpers estão ligados aos pinos corretos no Arduino;
- c. Verifique se você seguiu a montagem indicada para o modelo de display disponível em sua escola;
- d. Verifique se a programação está adequada ao modelo de display disponível no kit de robótica da sua escola;
- e. Verifique se a programação está adequada a cada porta digital.

3. Feedback e finalização

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

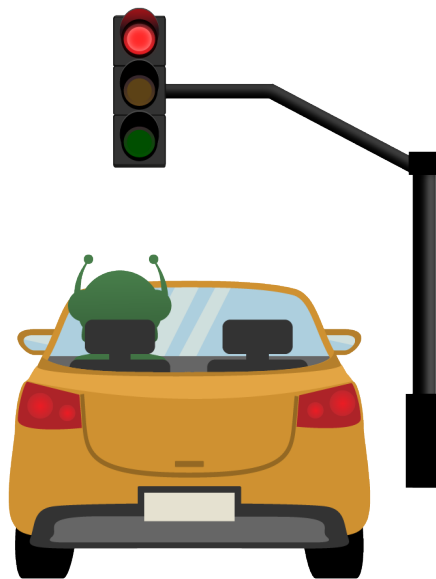
b. Analise seu projeto desenvolvido, de modo a atender aos requisitos para o funcionamento do semáforo completo com display 7 segmentos com um dígito.

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?

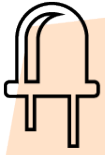
d. Reúna todos os componentes utilizados nesta aula e os organize novamente, junto aos demais, no kit de robótica.



AULA

05

SEMÁFORO COMPLETO COM DISPLAY DE 7 SEGMENTOS COM 1 DÍGITO



Videotutorial

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



<https://rebrand.ly/a05robotica2>

Acesse, também, pelo QRCode:



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

EQUIPE ROBÓTICA PARANÁ

Andrea da Silva Castagini Padilha
Cleiton Rosa
Darice Alessandra Deckmann Zanardini
Edgar Cavalli Junior
Edna do Rocio Becker
José Feuser Meurer
Marcelo Gasparin
Michele Serpe Fernandes
Michelle dos Santos
Orlando de Macedo Junior
Roberto Carlos Rodrigues

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 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 - Compartilha Igual 4.0](https://creativecommons.org/licenses/by-nc-sa/4.0/)

