Semáforo de 3 cores com interatividade (para carros e pedestres)

Esta experiência refere-se a um semáforo de 3 cores com interatividade, ou seja, dotado de um botão de acionamento (botoeira) para a travessia de pedestres, dos dois lados de uma rua.

Programação:

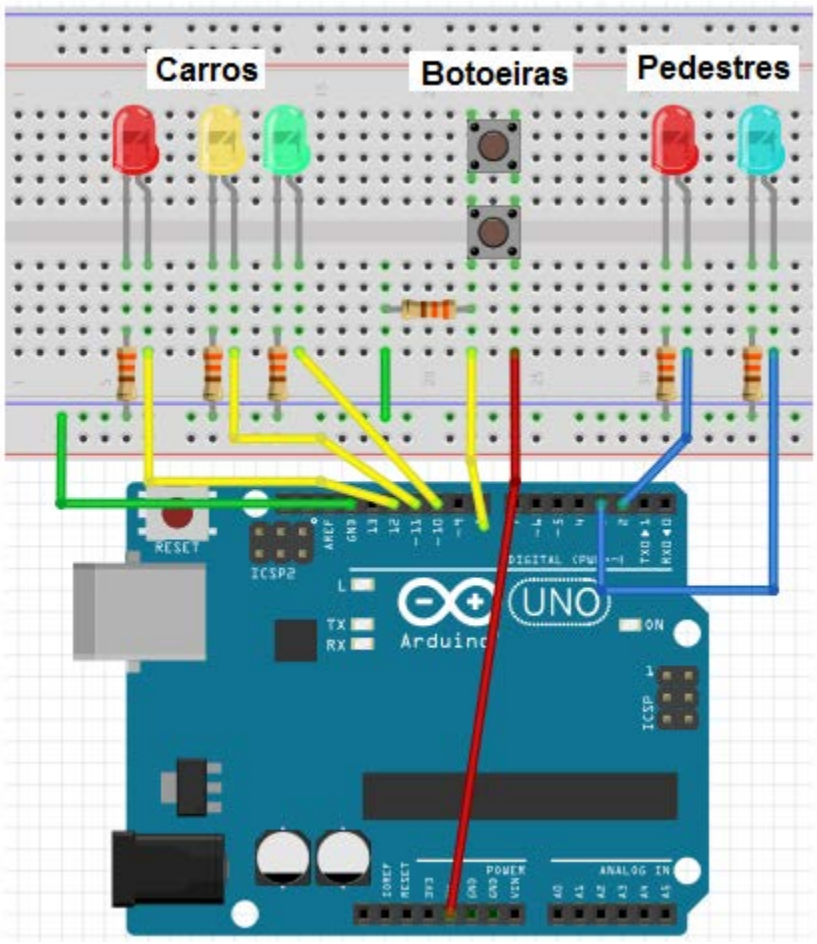
```
int carVm=12; // semáforo para carros – luz vermelha
int carAm=11; // semáforo para carros – luz amarela
int carVd=10; // semáforo para carros – luz verde
int button=8; // botoeira para pedestre
int pedVm=2; // semáforo para pedestre – luz vermelha
int pedVd=3; // semáforo para pedestre – luz verde (ou azul)
/* apenas um botão (botoeira) está assignado para o pino 8, pois na verdade para cada lado da rua
temos uma botoeira e ambas estão ligadas em paralelo, conforme se pode verificar mais adiante, no
layout desenvolvido no Fritzing*/
void setup() {
  pinMode(carVm, OUTPUT);
  pinMode(carAm, OUTPUT);
  pinMode(carVd, OUTPUT);
  pinMode(pedVm, OUTPUT);
  pinMode(pedVd, OUTPUT);
  pinMode(button, INPUT);
  digitalWrite (carVd, HIGH);
  digitalWrite (pedVm, HIGH);
}

void loop() {
  int buttonOn = digitalRead (button);
  if (buttonOn == HIGH) {
  digitalWrite(carVd, LOW);
  digitalWrite(carAm, HIGH);
  delay (3000);
  digitalWrite(carAm, LOW);
  digitalWrite(carVm, HIGH);
  delay(2000);
  digitalWrite(pedVm, LOW);
  digitalWrite(pedVd, HIGH);
  delay(6000);
  for (int i=0; i<=10; i++) // a luz verde do semáforo pisca 10 vezes, alertando o pedestre
{
    digitalWrite(pedVd, HIGH);
    delay(250);
    digitalWrite(pedVd,LOW);
    delay(250);
  }
  {
    digitalWrite (pedVm, HIGH);
    delay(1000);
    digitalWrite(carAm, HIGH);
```

```
  digitalWrite(carVm, LOW);
  delay(1000);
  digitalWrite(carVd, HIGH);
  digitalWrite(carAm, LOW);
 }
 }
 }
```

A figura a seguir mostra o layout do projeto desenvolvido no Fritzing, onde se observa os semáforos para carros e pedestres.

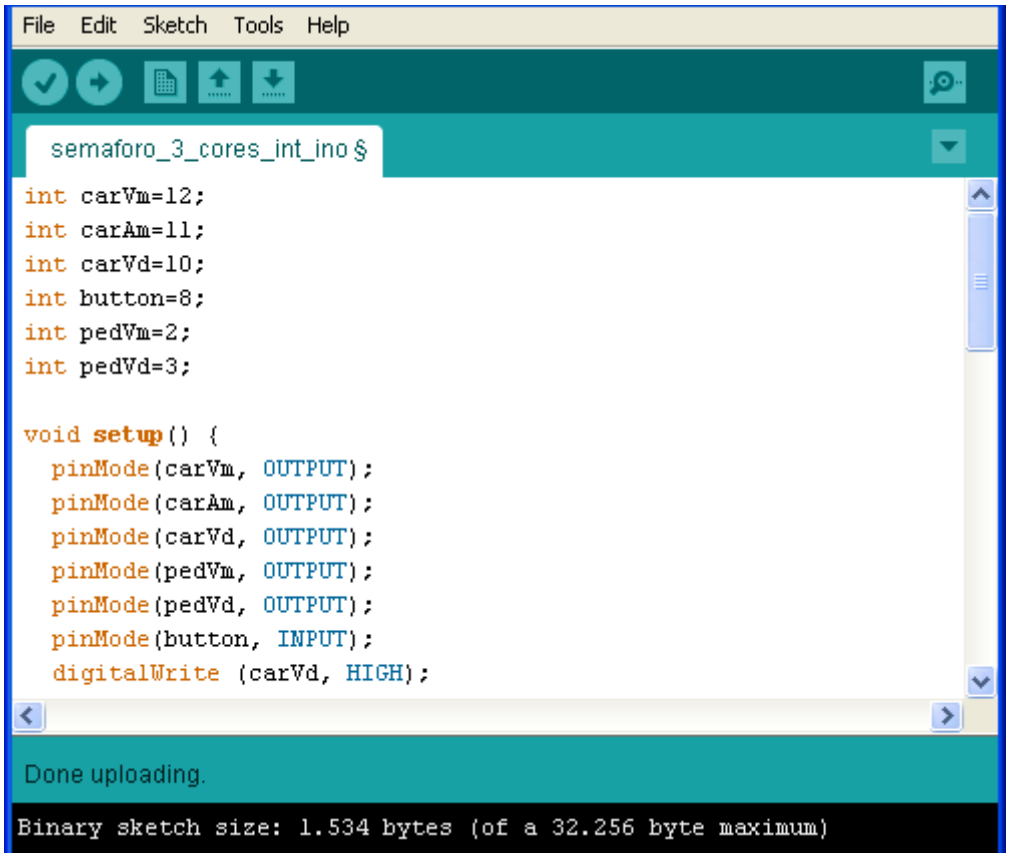
As botoeiras (uma de cada lado da rua) permitem a interação dos pedestres.



Caso não seja acionada a botoeira para a travessia de pedestres a luz verde para tráfego de carros estará sempre acesa.

Quando um pedestre acionar a botoeira de qualquer lado da rua, imediatamente o processo se inicia, bloqueando a passagem dos carros e liberando a travessia dos pedestres por um tempo, determinado na programação que poderá ser modificado de acordo com as características do local.

Transcorrido esse tempo, a luz verde do semáforo para pedestres começa a piscar, alertando-o. De acordo com a programação, a mesma pisca 10 vezes.



No entanto o semáforo para pedestres apresenta um inconveniente.

Quando a botoeira é acionada imediatamente o fluxo de carros é interrompido, o que acarretará sérios problemas em local de muito tráfego se o mesmo for constantemente acionado devido ao fluxo de pedestres.

Podemos então dar uma aprimorada na programação de tal forma que o semáforo de pedestre seja liberado após um determinado tempo.

A programação abaixo mostra essa modificação na cor azul:

```
int carVm=12;
int carAm=11;
int carVd=10;
int button=8;
int pedVm=2;
int pedVd=3;
unsigned long tempo;

void setup() {
pinMode(carVm, OUTPUT);
 pinMode(carAm, OUTPUT);
 pinMode(carVd, OUTPUT);
 pinMode(pedVm, OUTPUT);
```

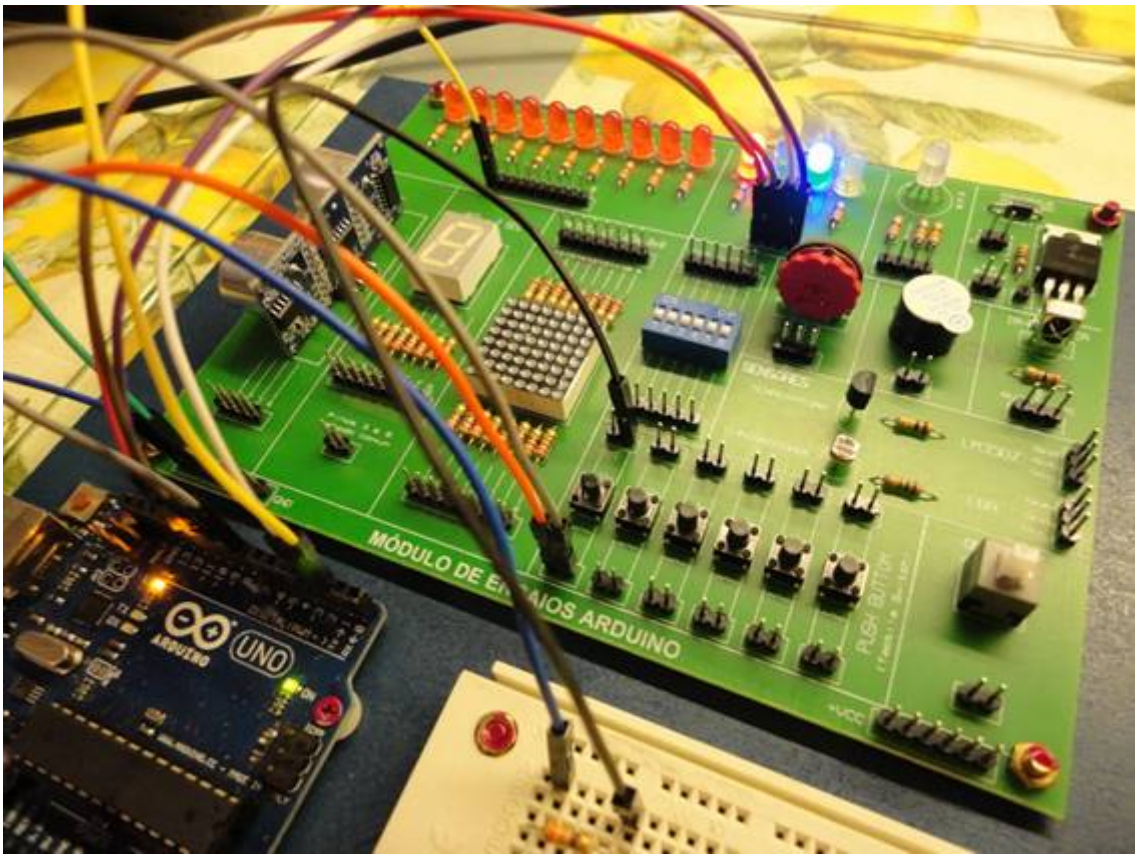
```
 pinMode(pedVd, OUTPUT);
 pinMode(button, INPUT);
 digitalWrite (carVd, HIGH);
 digitalWrite (pedVm, HIGH);
}

void loop() {
 int buttonOn = digitalRead (button);
 if (buttonOn == HIGH &&(millis() - tempo) > 6000) {
 digitalWrite(carVd, LOW);
 digitalWrite(carAm, HIGH);
 delay (3000);
 digitalWrite(carAm, LOW);
 digitalWrite(carVm, HIGH);
 delay(2000);
 digitalWrite(pedVm, LOW);
 digitalWrite(pedVd, HIGH);
 delay(6000);
 for (int i=0; i<=10; i++) {
  digitalWrite(pedVd, HIGH);
  delay(250);
  digitalWrite(pedVd,LOW);
  delay(250);
 }
 {
  digitalWrite (pedVm, HIGH);
  delay(1000);
  digitalWrite(carAm, HIGH);
  digitalWrite(carVm, LOW);
  delay(1000);
  digitalWrite(carVd, HIGH);
  digitalWrite(carAm, LOW);
  tempo = millis();
 }
 }
 }
```

O semáforo para pedestres voltará a operar somente depois de decorrido um tempo de 6 segundos, sendo que esse tempo poderá ser alterado para adaptação às condições do local.

A figura a seguir mostra a montagem no Módulo de Ensaios Arduino.

Para o semáforo dos carros foram usados os leds RGB e para o semáforo dos pedestres, o led azul e o led 1.


ARDUINO
UNO
PUSH BUTTON