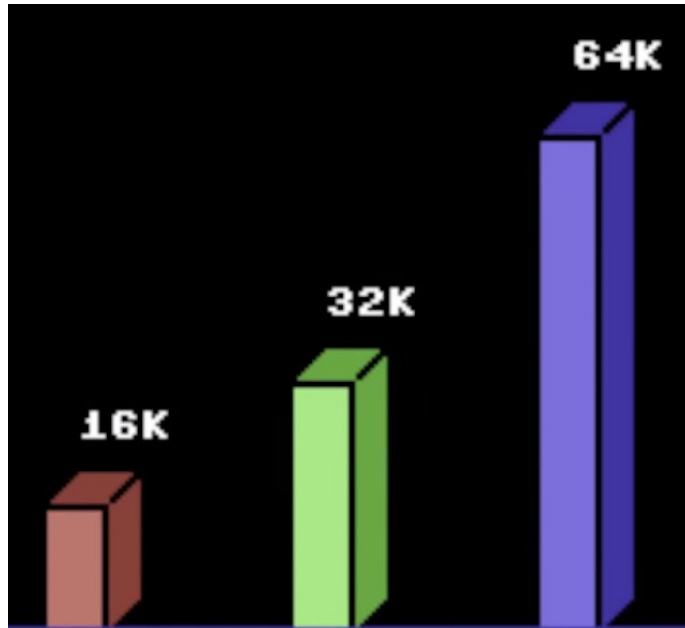


Aula 5

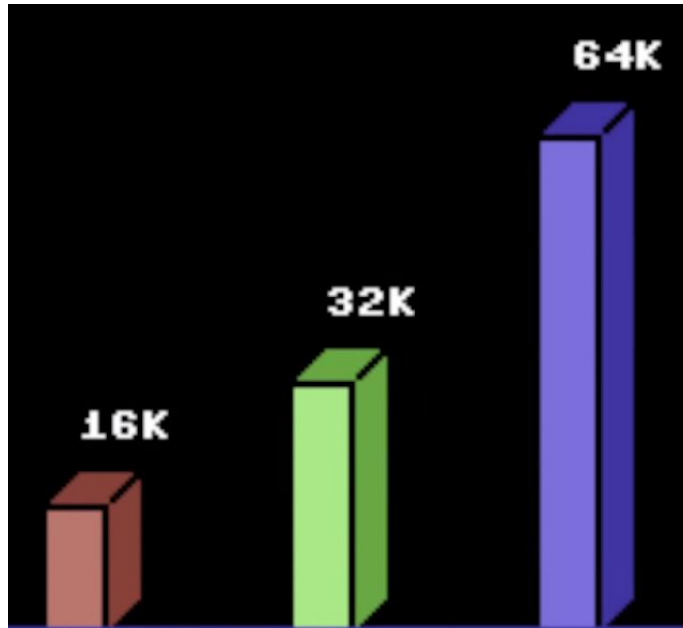
Commodore

320 X 200 PIXELS
= 64,000 PIXELS



Commodore

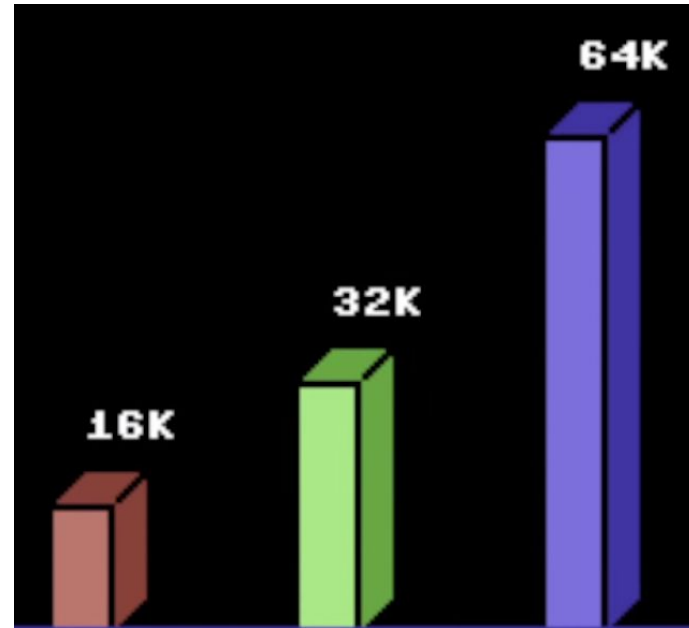
**320 X 200 PIXELS
= 64,000 PIXELS**



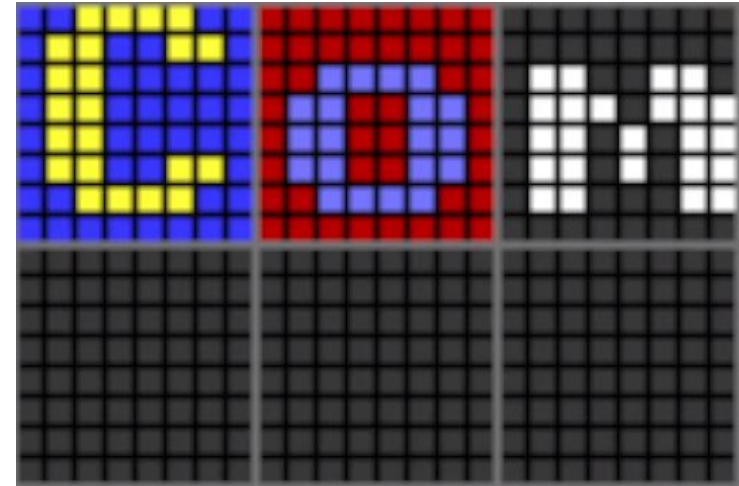
1 BIT (BLACK AND WHITE) = 8K
4 BIT (16 COLORS) = 32K
8 BIT (256 COLORS) = 64K
24 BIT (MILLIONS) = 192K

Commodore

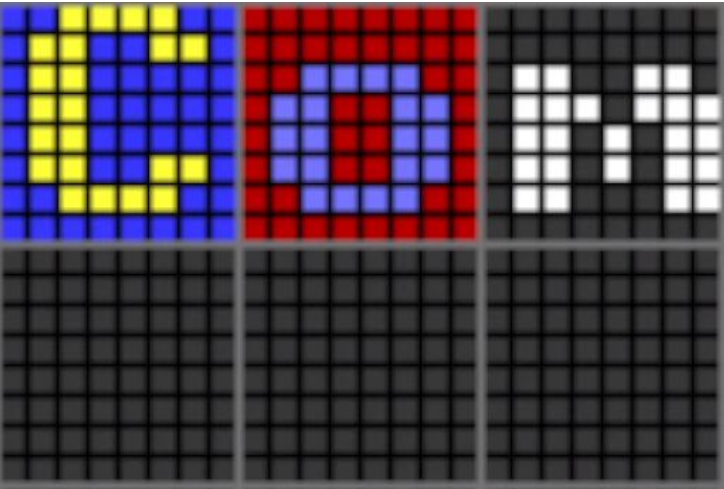
320 X 200 PIXELS
= 64,000 PIXELS

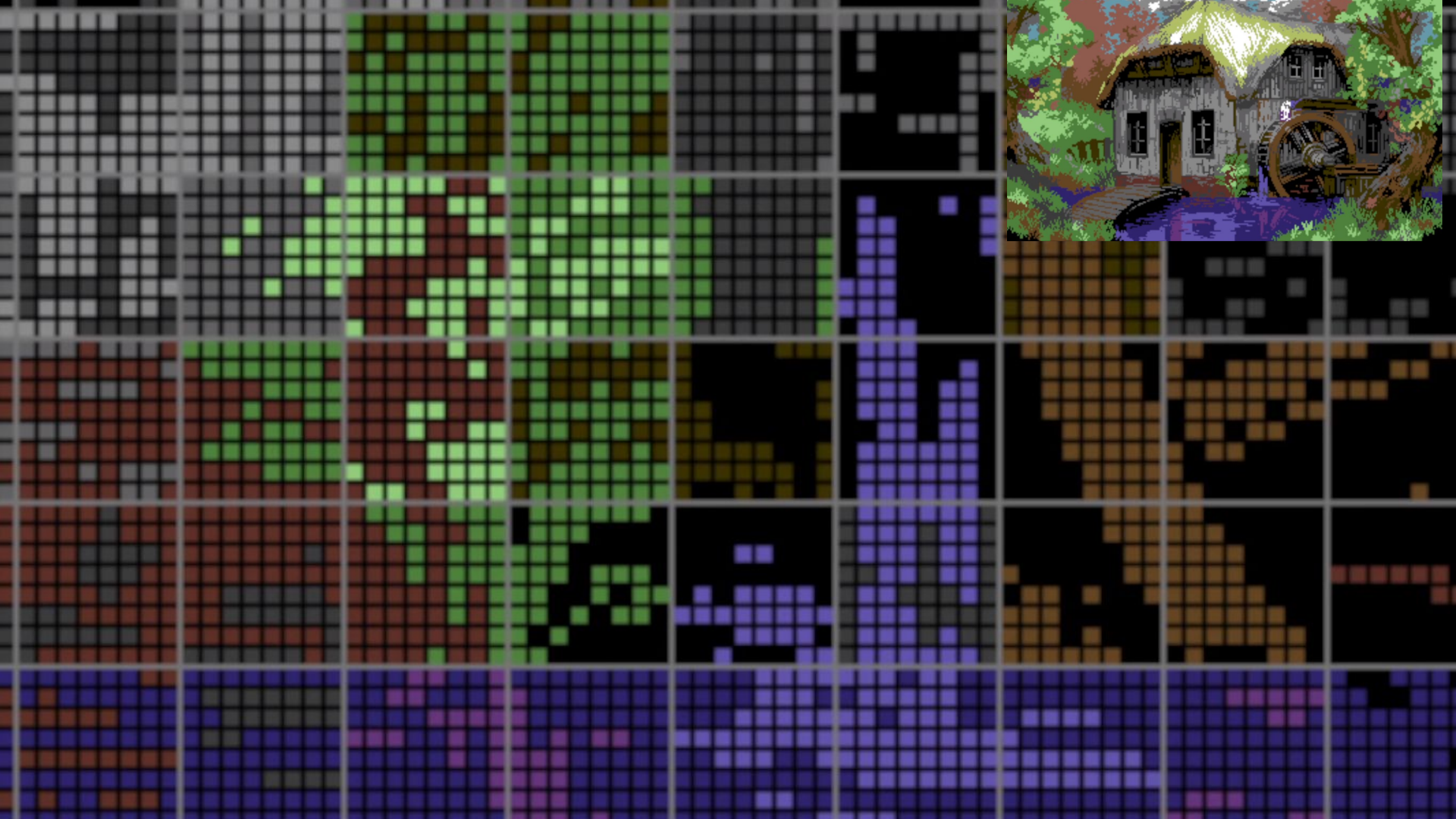


1 BIT (BLACK AND WHITE) = 8K
4 BIT (16 COLORS) = 32K
8 BIT (256 COLORS) = 64K
24 BIT (MILLIONS) = 192K

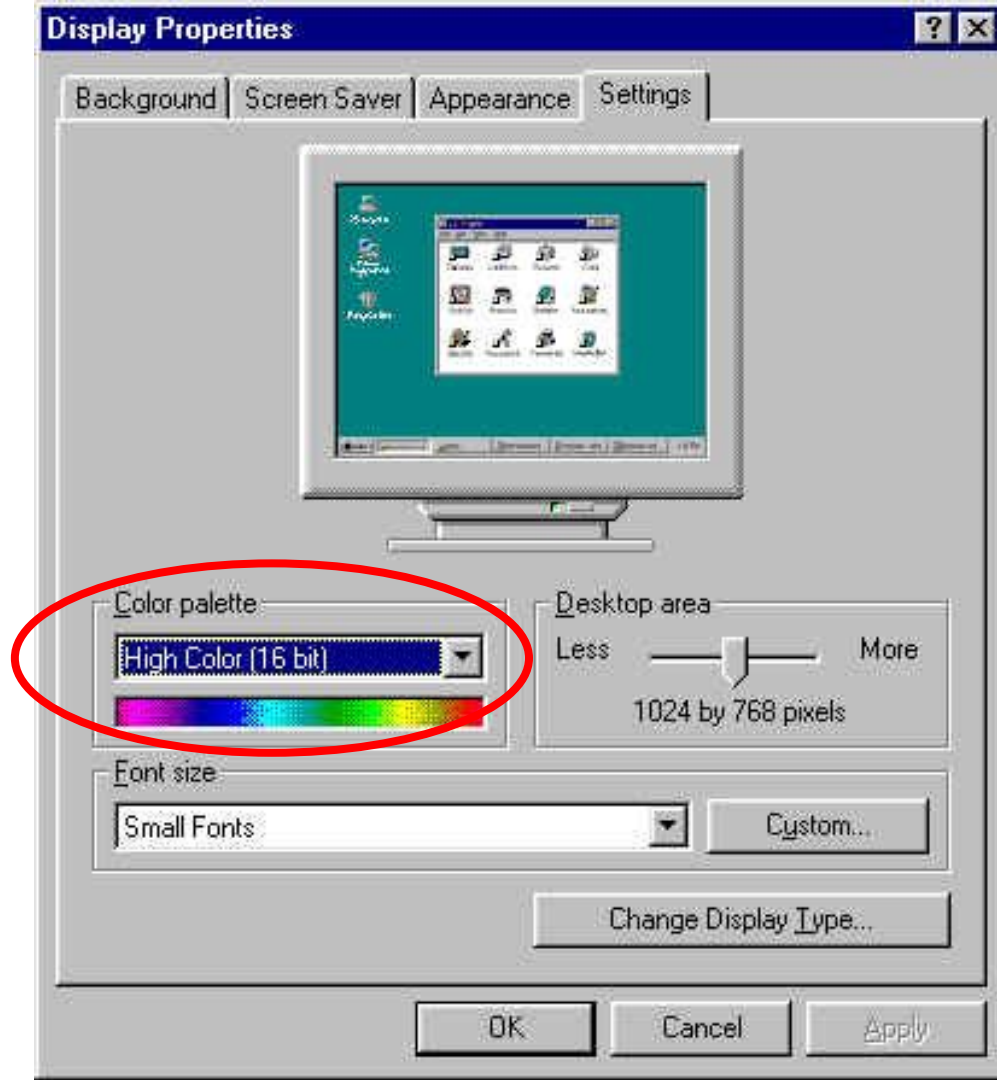


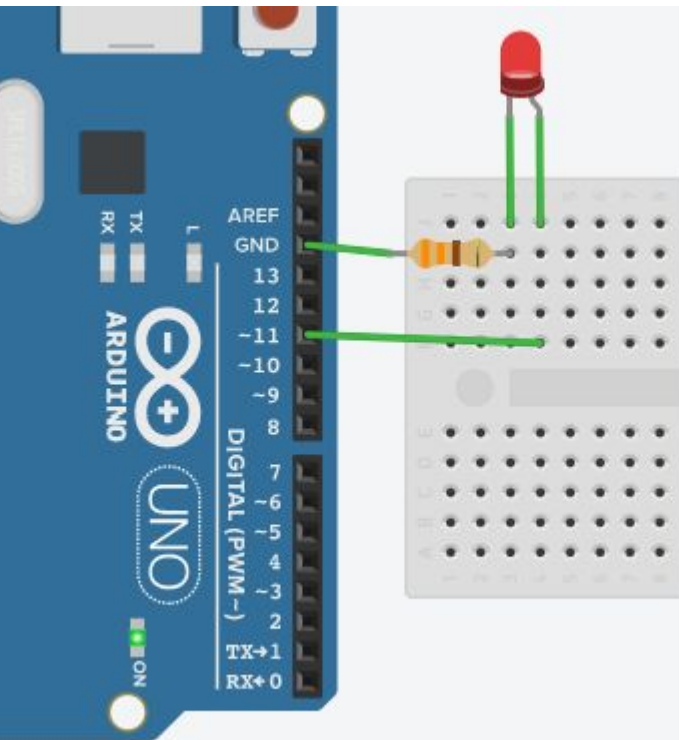
Commodore





Win95





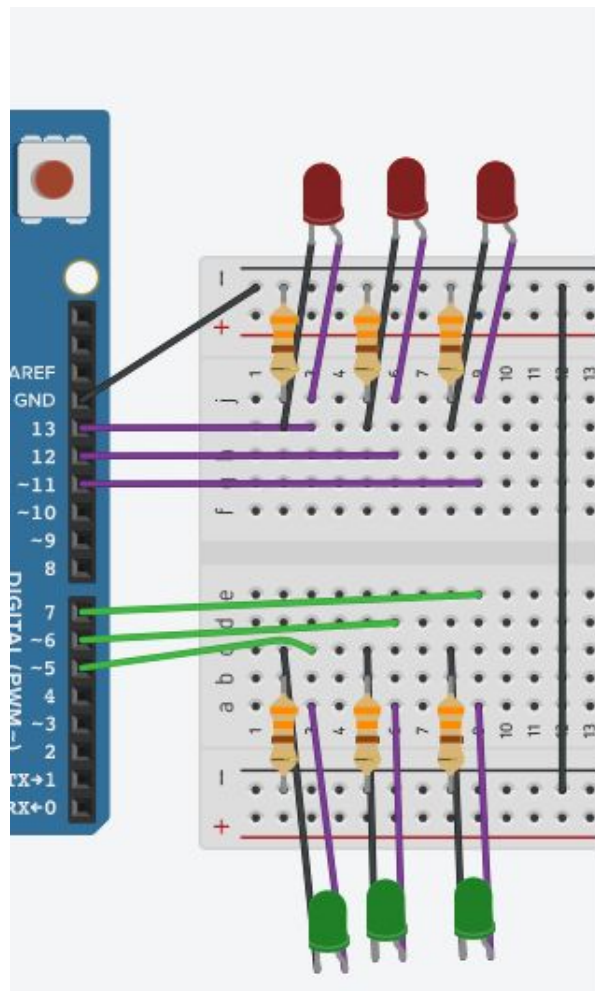
```
1 void setup()  
2 {  
3   pinMode(11, OUTPUT);  
4   digitalWrite(11, HIGH);  
5 }  
6  
7 void loop()  
8 {  
9 }
```

Text

```
1 void setup()  
2 {  
3   pinMode(11, OUTPUT);  
4 }  
5  
6 void loop()  
7 {  
8   digitalWrite(11, HIGH);  
9 }
```

Setup, loop, digitalWrite

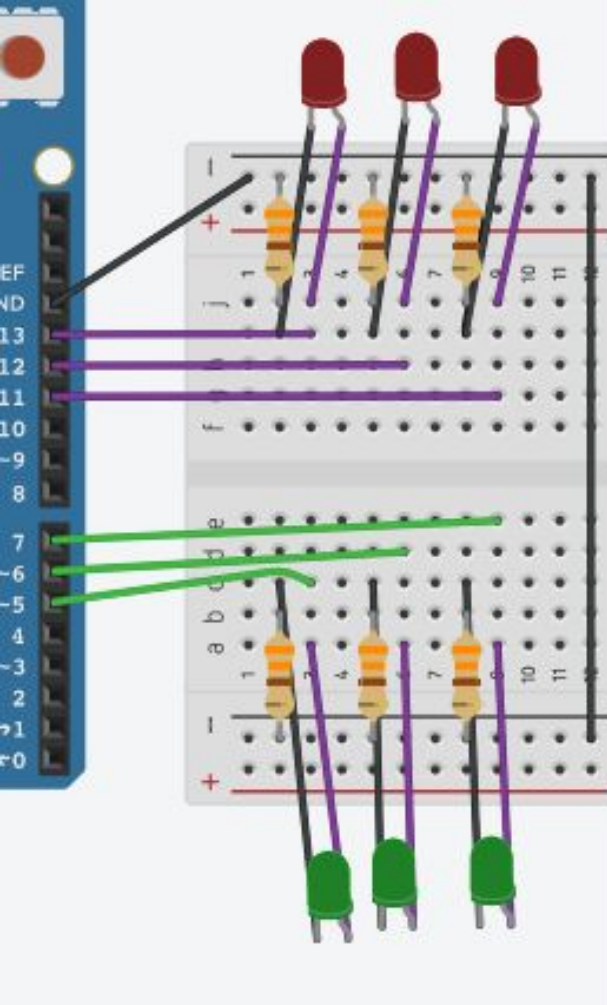
Funções



```
1 void setup()
2 {
3   pinMode(5, OUTPUT);
4   pinMode(6, OUTPUT);
5   pinMode(7, OUTPUT);
6   pinMode(11, OUTPUT);
7   pinMode(12, OUTPUT);
8   pinMode(13, OUTPUT);
9 }
10
11 void loop()
12 {
13   piscarLed1();
14   piscarLed2();
15 }
16
17 void piscarLed1() {
18   digitalWrite(13, HIGH);
19   delay(1000);
20   digitalWrite(13, LOW);
21   delay(1000);
22 }
23
24 void piscarLed2() {
25   digitalWrite(7, HIGH);
26   delay(1000);
27   digitalWrite(7, LOW);
28   delay(1000);
29 }
```

<https://www.tinkercad.com/things/aSWPCKOh8pZ>

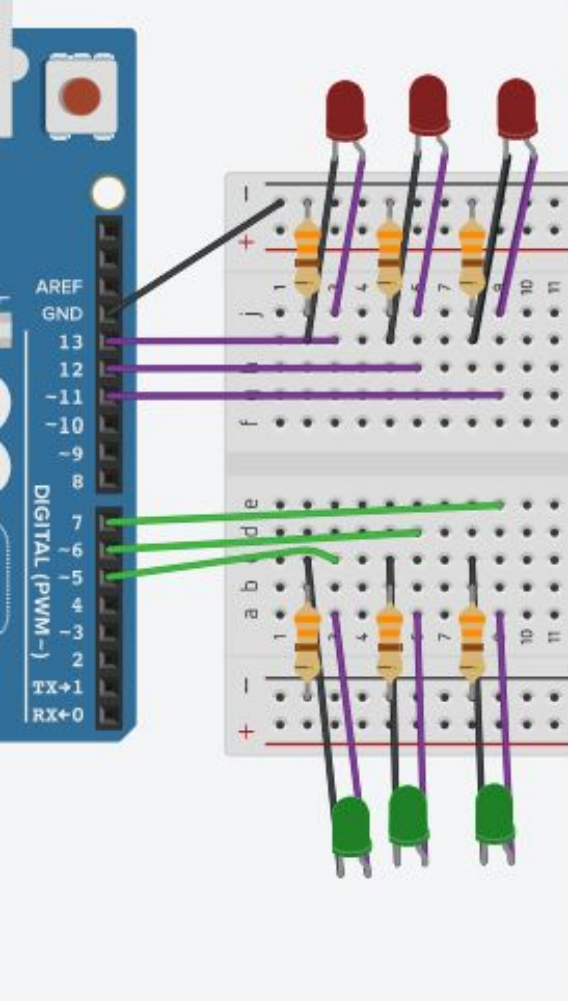
Função parametrizada



```
1 void setup()
2 {
3   pinMode(5, OUTPUT);
4   pinMode(6, OUTPUT);
5   pinMode(7, OUTPUT);
6   pinMode(11, OUTPUT);
7   pinMode(12, OUTPUT);
8   pinMode(13, OUTPUT);
9 }
10
11 void loop()
12 {
13   piscarLed(13, 200);
14   piscarLed(12, 200);
15   piscarLed(11, 200);
16   piscarLed(13, 1000);
17   piscarLed(12, 1000);
18   piscarLed(11, 1000);
19 }
20
21 void piscarLed(byte pino, int tempo){
22   digitalWrite(pino, HIGH);
23   delay(tempo);
24   digitalWrite(pino, LOW);
25   delay(tempo);
26 }
27
```

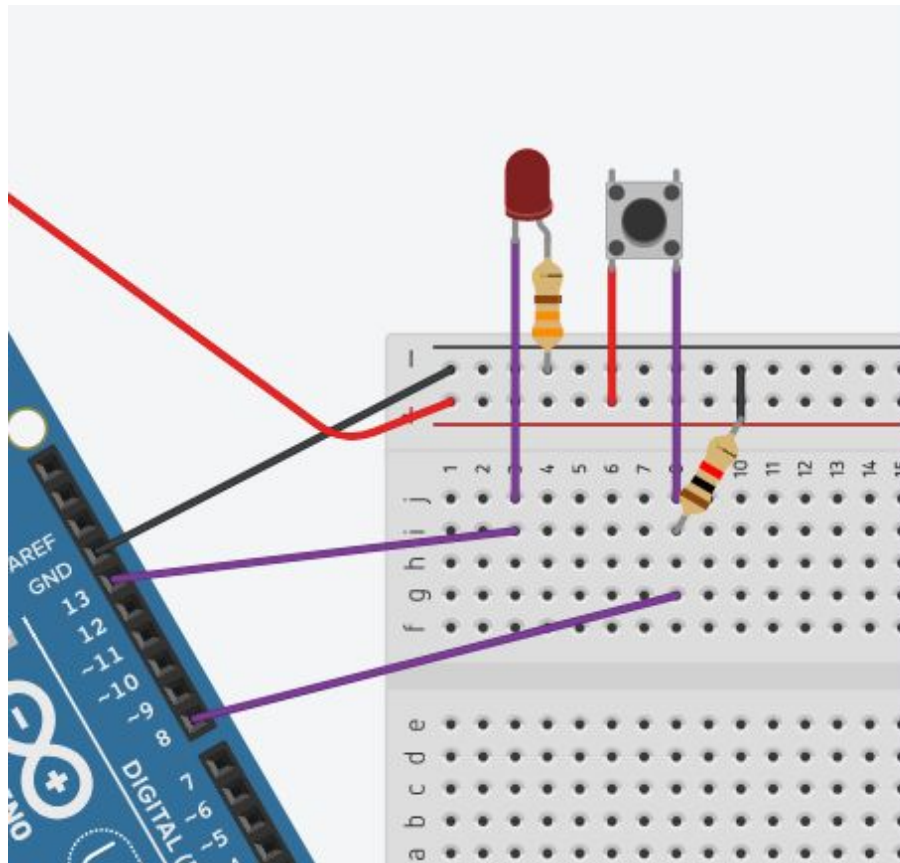
parâmetros

Função com array



```
1 byte pinosVermelhos[] = {13, 12, 11};
2 byte pinosVerdes[] = {5, 6, 7};
3 byte todosOsPinos[] = {5, 6, 7, 11, 12, 13};
4 void setup() {
5     pinMode(5, OUTPUT); pinMode(6, OUTPUT);
6     pinMode(7, OUTPUT); pinMode(11, OUTPUT);
7     pinMode(12, OUTPUT); pinMode(13, OUTPUT);
8 }
9 void loop() {
10     piscarLeds(pinosVermelhos, 3);
11     piscarLeds(pinosVerdes, 3);
12     piscarLeds(todosOsPinos, 6);
13 }
14 void piscarLeds(byte pinos[], byte tamanho) {
15     for (int i = 0; i < tamanho; i++) {
16         digitalWrite(pinos[i], HIGH);
17     }
18     delay(1000);
19     for (int i = 0; i < tamanho; i++) {
20         digitalWrite(pinos[i], LOW);
21     }
22     delay(1000);
23 }
24
25 void piscarLed(byte pino) {
26     digitalWrite(pino, HIGH);
27     delay(1000);
28     digitalWrite(pino, LOW);
29     delay(1000);
30 }
```

arrays



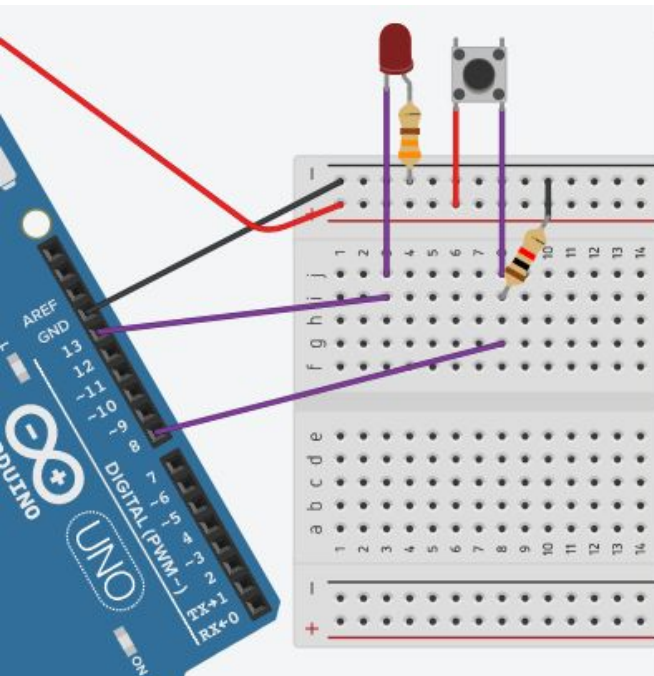
```
1  int contador = 0;
2
3  void setup()
4  {
5      pinMode(8, INPUT);
6      pinMode(13, OUTPUT);
7      Serial.begin(9600);
8  }
9
10 void loop()
11 {
12     if (digitalRead(8) == HIGH) {
13         contador++;
14         Serial.println(contador);
15     }
16 }
```



Serial Monitor

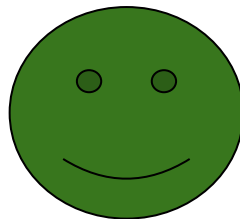


<https://www.tinkercad.com/things/kQ0YTbTGF7f>



DETECÇÃO DE EDGE

```
1 int contador = 0;
2 byte estadoPrevioBotao = LOW;
3
4 void setup()
5 {
6   pinMode(8, INPUT);
7   pinMode(13, OUTPUT);
8   Serial.begin(9600);
9 }
10 void loop()
11 {
12   byte estadoAtualBotao = digitalRead(8);
13   if (estadoPrevioBotao == LOW && estadoAtualBotao == HIGH) {
14     contador++;
15     Serial.println(contador);
16     estadoPrevioBotao = HIGH;
17   }
18   else if (estadoPrevioBotao == HIGH && estadoAtualBotao == LOW) {
19     estadoPrevioBotao = LOW;
20   }
21 }
```

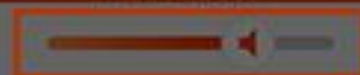


<https://www.tinkercad.com/things/clNrEErV2bi>



Brian

33*33*33333



Aviso - Volume Alto

Volume acima desse nível pode
distorcer a qualidade do áudio, além de
não ser bom para sua audição.
Deseja aumentar mesmo assim?

CANCELAR

OK

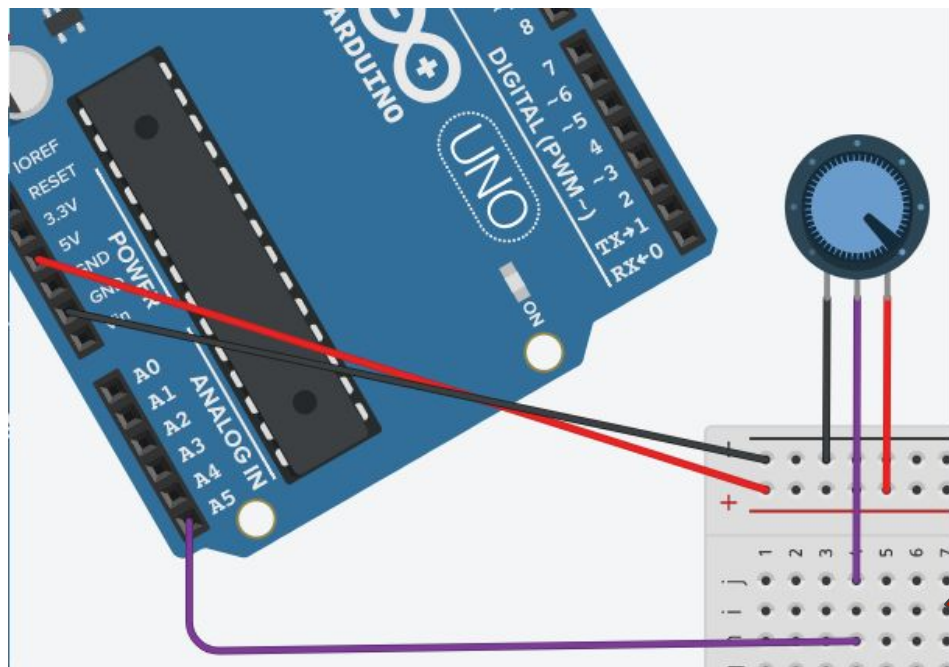
Hoje, 11:49

Compartilhamento de localização desativado



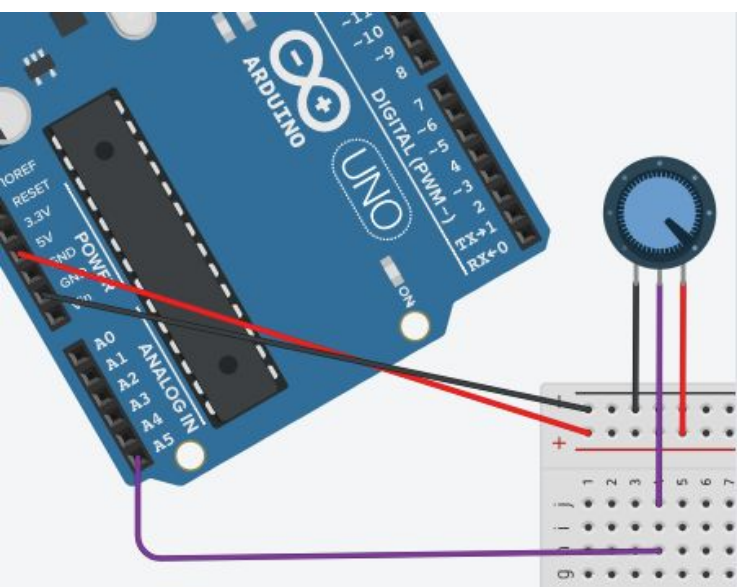
PRESSIONE
PARA FALAR



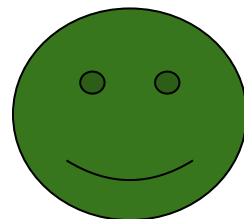


```
1 void setup()  
2 {  
3   Serial.begin(9600);  
4 }  
5  
6 void loop()  
7 {  
8   int leituraVolume = analogRead(A5);  
9   if (leituraVolume == 900) {  
10     Serial.println("Cuidado com o volume alto!");  
11   }  
12 }
```

<https://www.tinkercad.com/things/e1dpxmA5dT0>



```
1 boolean alertaExibido = false;
2 void setup()
3 {
4     Serial.begin(9600);
5 }
6 void loop()
7 {
8     int leituraVolume = analogRead(A5);
9     if (! alertaExibido && leituraVolume > 900) {
10         Serial.println("Cuidado com o volume alto!");
11         alertaExibido = true;
12     }
13     else if (alertaExibido && leituraVolume <= 900) {
14         alertaExibido = false;
15     }
```



<https://www.tinkercad.com/things/0dw7xDjd5ZU>

Máquinas de estado

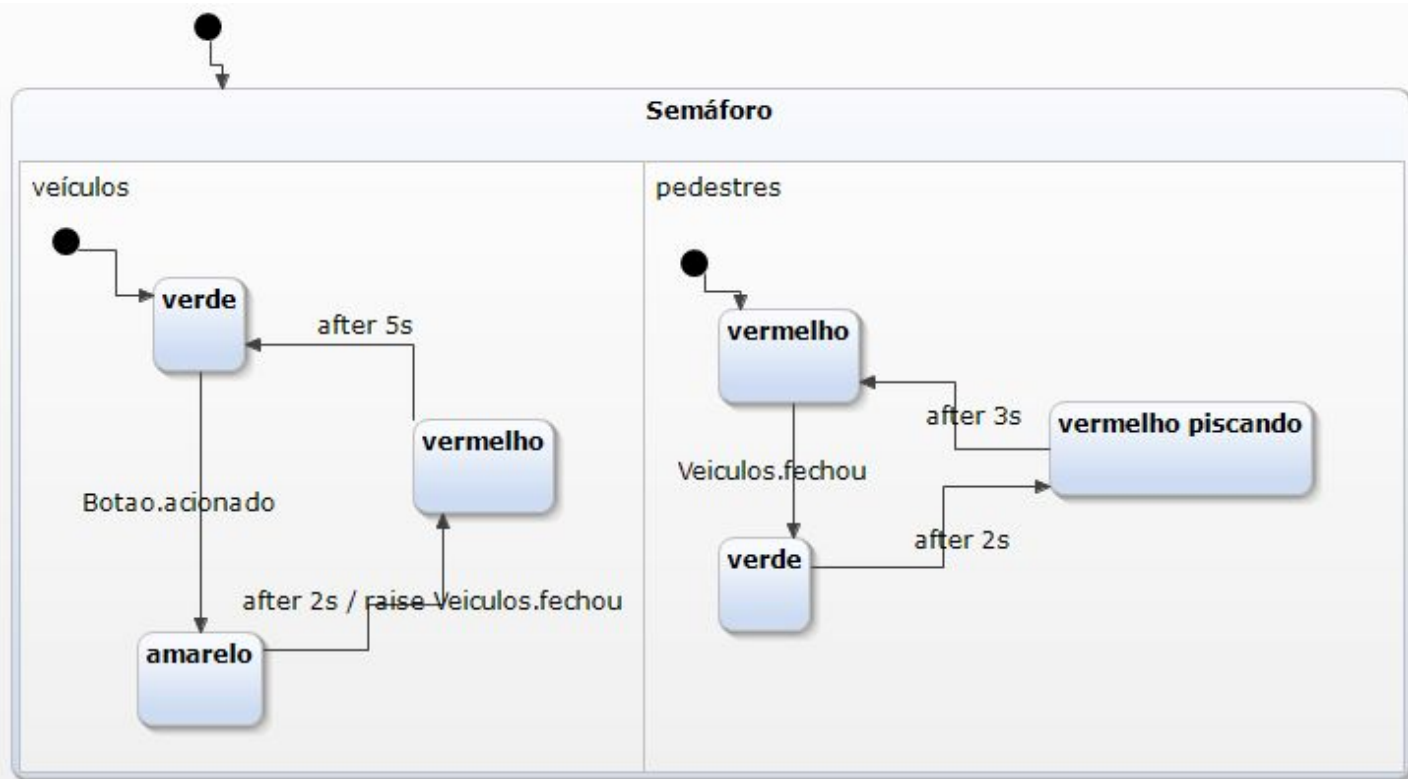
Texto: diferentes pessoas podem interpretar diferentemente

— — —

Exemplo:

Escreva um semáforo combinado de veículos e de pedestre. Por padrão, o semáforo deverá ficar aberto para os veículos e fechado para os pedestres. O projeto deve conter um botão (push button) para pedestres. Quando o botão é pressionado, o semáforo dos veículos deverá ficar amarelo (imediatamente). Após um certo tempo, o semáforo dos veículos passará de amarelo para vermelho. Concomitantemente, o semáforo dos pedestres ficará verde. Após algum tempo, o semáforo dos pedestres ficará piscando em vermelho, indicando que irá fechar em breve, enquanto o semáforo dos veículos permanece vermelho. Após algum tempo piscando, o semáforo dos pedestres ficará vermelho em definitivo, ao mesmo tempo em que o semáforo dos veículos voltará para seu estado inicial (verde).

Diagrama de estados: preciso, não dá margem a interpretação



farolPedestre
interface Veiculos:
out event fechou

interface Botao:
in event acionado

Diagrama de estados: pode ser simulado

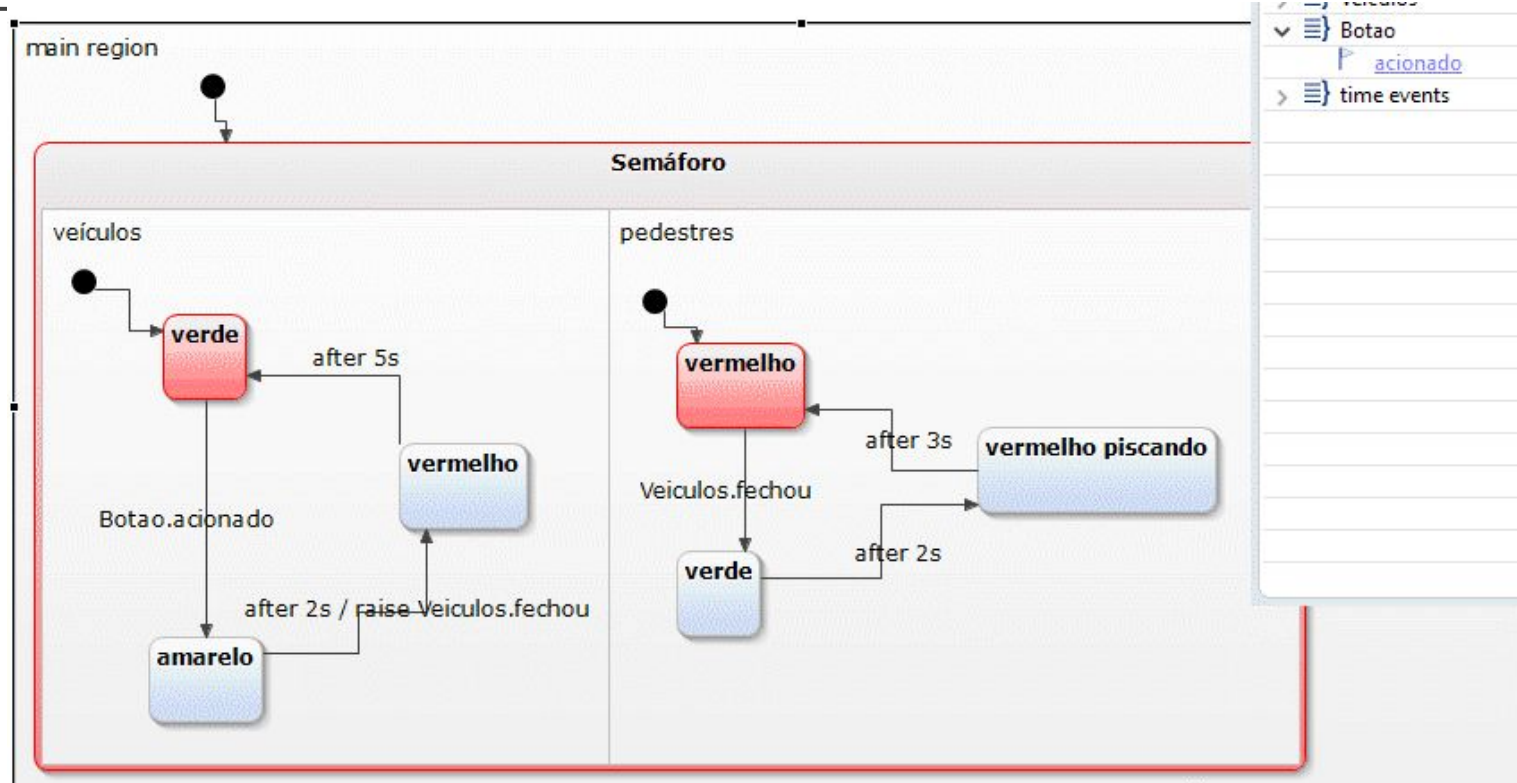


Diagrama de estados: pode criar código C++ automaticamente

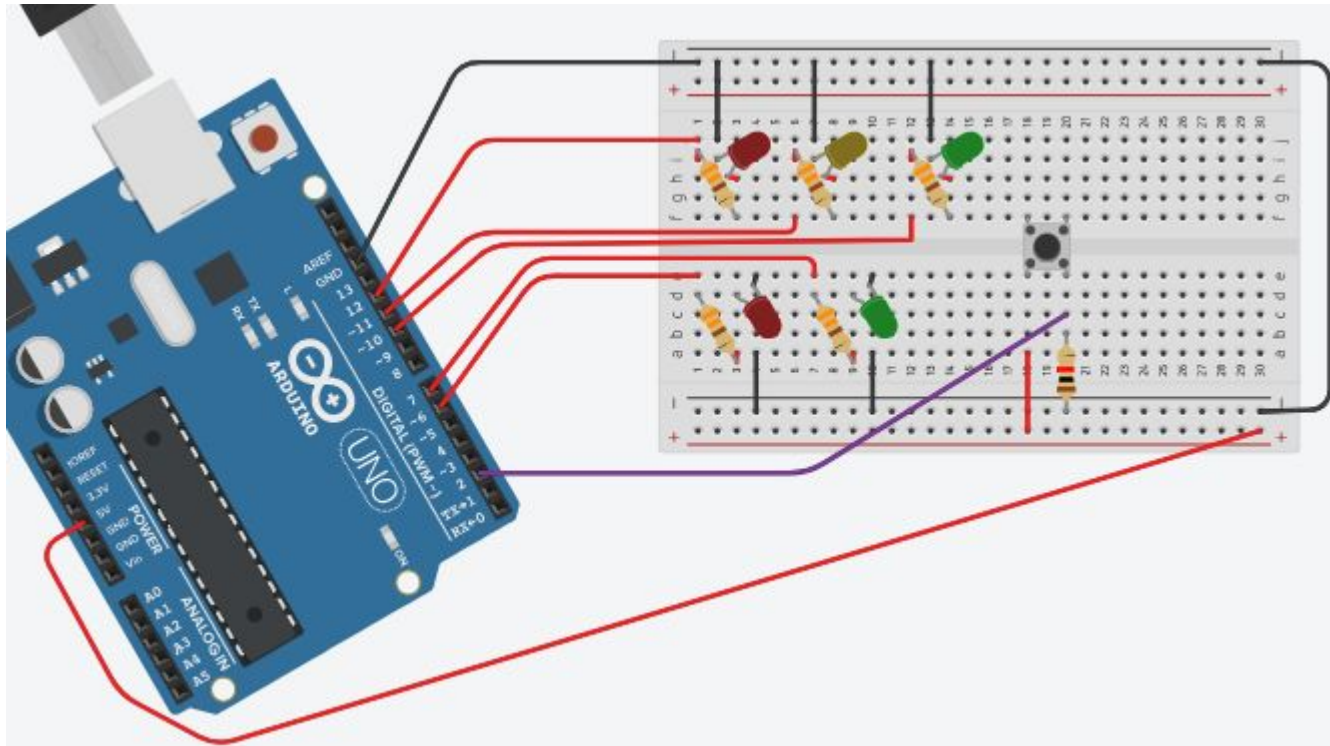
— — —

<https://blogs.itemis.com/en/developing-software-for-arduino-with-yakindu-statechart-tools>

Ferramenta utilizada: Yakindu

<https://info.itemis.com/state-machine/download-yakindu-state-chart-tools>

Exemplo com código manual



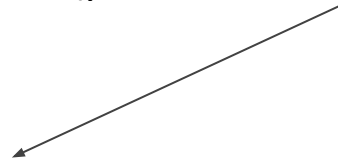
<https://www.tinkercad.com/things/f9wJWOLGteV>

Exemplo com código manual - código (parcial)

```
void loop() {  
  if (estadoVeiculos == VEICULOS_VERDE) {  
    inicializarSemaforosVeiculos();  
    if (digitalRead(botao) == HIGH) {  
      estadoVeiculos = VEICULOS_AMARELO;  
    }  
  }  
  else if (estadoVeiculos == VEICULOS_AMARELO) {  
    acenderAmareloDoCarro();  
    delay(2000);  
    estadoVeiculos = VEICULOS_VERMELHO;  
    estadoPedestres = PEDESTRES_VERDE;  
  }  
  else if (estadoVeiculos == VEICULOS_VERMELHO) {  
    acenderVermelhoDoCarro();  
    delay(3000);  
  }  
  if (estadoPedestres == PEDESTRES_VERDE) {  
    acenderVerdeDoPedestre();  
    estadoPedestres = PEDESTRES_VERMELHO_PISCANDO;  
  }  
  else if (estadoPedestres == PEDESTRES_VERMELHO_PISCANDO) {  
    apagarVerdeDoPedestre();  
    piscarVermelhoDoPedestre(3);  
    estadoPedestres = PEDESTRES_VERMELHO;  
    estadoVeiculos = VEICULOS_VERDE;  
  }  
  else if (estadoPedestres == PEDESTRES_VERMELHO) {  
    inicializarSemaforosPedestres();  
  }  
}
```

Filtrando ruidos

- Realizar várias leituras da medida e calcular a média, mediana, ou moda desses valores
- Descartar outliers/variações muito bruscas
- Combinar a leitura atual com leituras prévias (possivelmente com média aritmética ou média ponderada)



$$\text{valorSuavizado}_n = (\text{alfa} \times \text{valorSuavizado}_{n-1}) + ((1 - \text{alfa}) \times \text{leitura}_n)$$

Alfa é uma constante entre 0 e 1. Quanto maior o valor de *alfa*, maior o efeito de suavização.