

SEU NOME: \_\_\_\_\_

**AULA 16**

**CIRCUITO COM ARDUÍNO**

Como temos apenas uma aula, vamos direto ao circuito e como você deve montar o circuito.

Lista de materiais:

- Um Arduino UNO;
- Seis resistores de 200 ohms (ou valores próximos, a depender da disponibilidade na escola);
- Dois LEDs vermelhos;
- Dois LEDs amarelos;
- Dois LEDs verdes;
- Diversos fios (*jumpers* MM);
- Uma bateria 9V;
- Cabo *clip* com *plug* P4.

Inicialmente, não ligue o Arduino: monte o circuito primeiro. Veja abaixo um *print* para a simulação que o professor fez no Tinkercad.

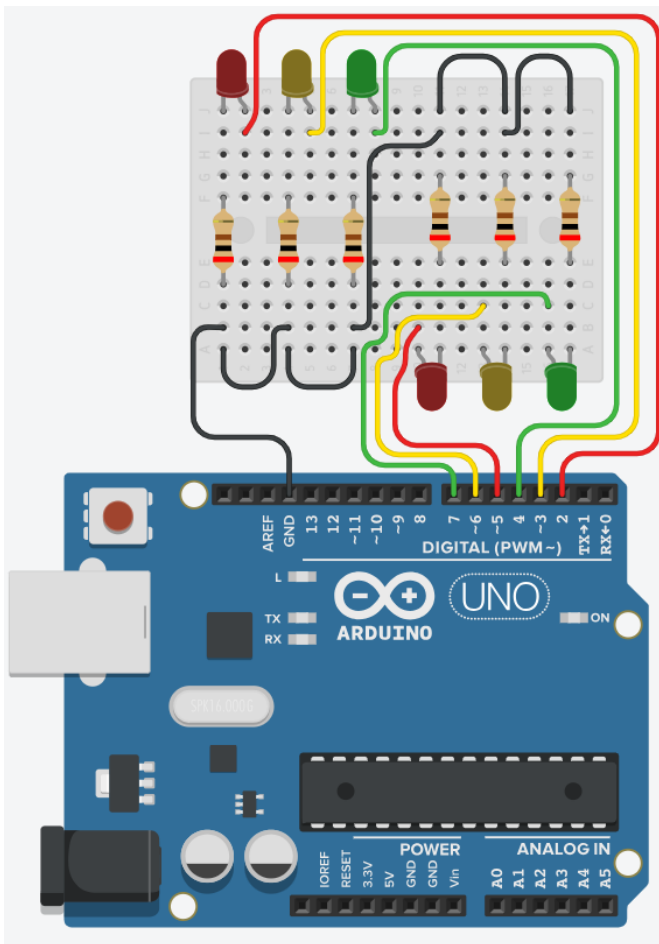


Figura 1: Circuito que irá simular um semáforo.



Figura 2: Acesse a simulação pelo QR-Code acima



Figura 3: Acesso a este arquivo em pdf, caso queira a versão colorida.

**INSTRUÇÕES GERAIS SOBRE A MONTAGEM**

Note que o Arduino possui diversos buracos onde podemos ligar os *jumpers*. Esses buracos são chamados de portas. Note na Figura 1 que o Arduino possui 14 portas (de 0 até 13) na parte superior. Note que é preciso ligar os LEDs nas portas de número 2 até 7 e a escolha das portas é feita durante a programação.

Note que, na sequência, temos as seguintes ligações:

PORTA ARDUÍNO	LED*
2	Vermelho em cima
3	Amarelo em cima
4	Verde em cima
5	Vermelho em baixo
6	Amarelo em baixo
7	Verde em baixo

\*Os pinos maiores dos LEDs é que são conectados no Arduino.

Note também que o PINO MAIOR DOS LEDS é quem são conectados nas portas do Arduino.

O PINO MENOR de cada LED é ligado a um resistor para evitar que o LED queime.

Todos os resistores estão conectados no GND do Arduino. Como o Arduino possui poucas saídas GND você deve ligar o GND do Arduino em apenas um resistor e daí conectar, com outros cabos (*jumpers*), nos demais resistores.

**EXPLICANDO O PROGRAMA DO ARDUÍNO**

Caso tenha interesse em saber como funciona a programação do Arduino, abaixo está a explicação de parte do programa.

CÓDIGO	EXPLICAÇÃO
<pre>void setup() {</pre>	<p>Aqui temos o que chamamos de função. Esta função é uma função de configuração, por isso é chamada de <code>setup()</code>. A palavra <i>void</i>, que significa “vazio”, informa que a função faz algo mas não retorna nada. É como pedir para alguém fazer algo, essa pessoa faz, mas não te diz que fez. Note a abertura da chaves “{” que indica que tudo que estiver entre { } está dentro da função <i>setup</i>.</p>
<pre>pinMode(2, OUTPUT); pinMode(3, OUTPUT); pinMode(4, OUTPUT); pinMode(5, OUTPUT); pinMode(6, OUTPUT); pinMode(7, OUTPUT); }</pre>	<p>Aqui é onde configuramos as portas do Arduino informando que os pinos de número 2 até 7 funcionarão como “saída” (<i>OUTPUT</i>), isto é, fornece 5V quando ligamos a porta. <i>pinMode</i> é uma função usada para configurar as portas de saída. Note também o fechamento das chaves “}”.</p>

PROFESSOR DANILO

3º Trimestre

ROBÓTICA – 7º ANO – 11/11/2024

<pre>void loop() {</pre>	Esta função também não retorna nada ( <i>void</i> ) e é executada indefinidamente como um <i>loop</i> infinito. Note novamente a abertura das chaves.
<pre>// Semáforo de cima vermelho</pre>	O que está a direita das barras duplas é entendido como comentário e o Arduino simplesmente ignora. Esse tipo de texto é importante para documentar e explicar o que cada parte do código faz.
<pre>digitalWrite(2, HIGH); digitalWrite(3, LOW); digitalWrite(4, LOW);</pre>	Agora usamos a função que “escreve” na porta. Numa tradução livre, “digitalWrite” significa “escreva digitalmente”. Isso quer dizer que o Arduino irá simplesmente ligar ou desligar (zero ou um, verdadeiro ou falso). Essa estrutura binária, de ligar ou desligar, é chamado de “digital”. Note que entre os parêntesis colocamos o número da porta (2, 3 e 4) e posteriormente informamos se queremos ligar ( <i>HIGH</i> ) ou desligar ( <i>LOW</i> ) o LED ligado na respectiva porta.
<pre>// Semáforo de baixo verde digitalWrite(5, LOW); digitalWrite(6, LOW); digitalWrite(7, HIGH);</pre>	Note que esta parte do código liga o LED verde do semáforo de baixo (conforme Figura 1).
<pre>delay(20000);</pre>	Neste trecho, pedimos para que o Arduino espere, sem fazer nada mais, por 20000 milissegundos, isto é, por 20 segundos.
<pre>// Semáforo de baixo amarelo digitalWrite(5, LOW); digitalWrite(6, HIGH); digitalWrite(7, LOW); delay(5000);</pre>	Este trecho do código controla o semáforo de baixo ligando o LED Amarelo e desligando o Verde, que estava ligado. Depois, aguarda-se 5 segundos.
<pre>// Semáforo de baixo vermelho digitalWrite(5, HIGH); digitalWrite(6, LOW); digitalWrite(7, LOW); // Semáforo de cima verde digitalWrite(2, LOW); digitalWrite(3, LOW); digitalWrite(4, HIGH); delay(20000); // Semáforo de cima amarelo digitalWrite(2, LOW); digitalWrite(3, HIGH); digitalWrite(4, LOW); delay(5000); }</pre>	Observe a parte final do código: <ul style="list-style-type: none"> <li>• Comentário informando que será programado o sinal vermelho;</li> <li>• Liga o sinal vermelho e desliga os demais;</li> <li>• Informa que irá ligar o LED verde de cima;</li> <li>• Liga o semáforo verde de cima deligando as demais luzes de cima;</li> <li>• Aguarda 20 segundos;</li> <li>• Aciona o semáforo amarelo de cima;</li> <li>• Finalmente, fecha a função <i>setup</i> com a chave “}”.</li> </ul>

#### PROGRAMA ENVIADO PARA O ARDUÍNO

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

void loop()
{
  // Semáforo de cima vermelho
  digitalWrite(2, HIGH);
  digitalWrite(3, LOW);
  digitalWrite(4, LOW);
  // Semáforo de baixo verde
  digitalWrite(5, LOW);
  digitalWrite(6, LOW);
  digitalWrite(7, HIGH);
  delay(20000);
  // Semáforo de baixo amarelo
  digitalWrite(5, LOW);
  digitalWrite(6, HIGH);
  digitalWrite(7, LOW);
  delay(5000);
  // Semáforo de baixo vermelho
  digitalWrite(5, HIGH);
  digitalWrite(6, LOW);
  digitalWrite(7, LOW);
  // Semáforo de cima verde
  digitalWrite(2, LOW);
  digitalWrite(3, LOW);
  digitalWrite(4, HIGH);
  delay(20000);
  // Semáforo de cima amarelo
  digitalWrite(2, LOW);
  digitalWrite(3, HIGH);
  digitalWrite(4, LOW);
  delay(5000);
}
```

#### PROGRAMA ENVIADO PARA O ARDUÍNO

E aí, gostou?  
Você pode programar um Arduino usando o simulador apresentado na Figura 2.  
Nem precisa comprar um Arduino.  
E sobre como aprender, há muitos vídeos e tutoriais na internet. Fique a vontade para conversar com seu professor para aprender mais sobre programação, tecnologia, robótica e informática.