

## Laboratório de Arduino Semáforo

Baseado no livro: Arduino Básico

### Objetivo:

Utilizar conceitos lógicos de programação para o controle de um semáforo em duas vias.

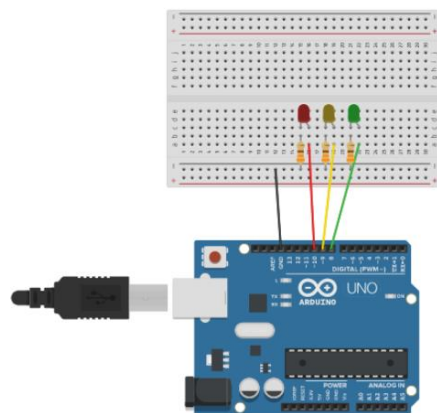
### Parte 1 - Semáforo simples.

Neste projeto, você criará um semáforo que irá do verde ao vermelho, passando pelo amarelo, e que retornará depois de um intervalo de tempo. Este projeto poderia ser utilizado para criar um conjunto funcional de semáforos para uma maquete de ferrovia ou para uma pequena cidade de brinquedo.

Para esse exemplo, você precisará:

- Uma placa protoboard,
- LED vermelho difuso
- LED amarelo difuso
- LED verde difuso
- 3 resistores de 300 Ohms e alguns fios de conexão.

A Figura abaixo mostra a disposição dos componentes. Atenção nesta hora!



Dessa vez, você conectará três LEDs, com o ânodo (positivo -perna grande) de cada um indo para os pinos digitais 8, 9 e 10, por meio de um resistor de 150  $\Omega$  cada (ou do valor necessário para seu caso). Leve um fio jumper do terra do Arduino para o barramento do terra no topo da protoboard; um fio terra vai do terminal cátodo de cada LED para o barramento terra comum por meio de um resistor — dessa vez conectado ao cátodo. (Para esse circuito simples, não importa se o resistor está conectado ao ânodo ou ao cátodo).

Codificando:

Implemente e faça o upload para seu Arduino do código . Os LEDs devem passar pelos três estados, ficando 2 segundos na luz verde, 1 segundo na luz amarela e 4 segundos na luz vermelha.

## Parte 2 - Semáforo de carros com semáforo de pedestre.

Dessa vez, você estenderá o projeto anterior para incluir um farol de pedestre e um botão, que será pressionado pelos pedestres para solicitar a travessia da rua. O Arduino reagirá quando o botão for pressionado, alterando o estado das luzes para que os carros parem e os pedestres possam atravessar em segurança.

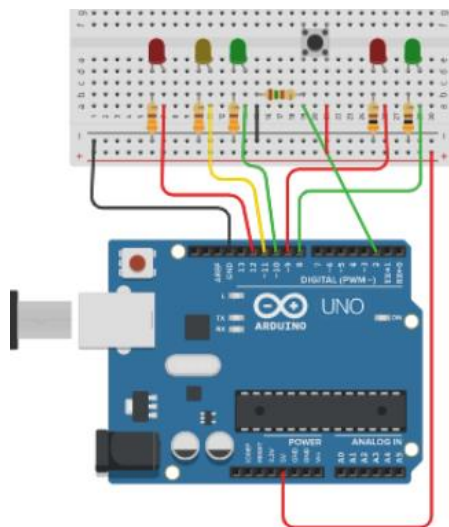
Neste projeto, você também aprenderá como criar suas próprias funções dentro do código. Embora o protoboard e os os fios jumper não estejam na lista de componentes necessários, você sempre terá de utilizá-los.

A seguir, veremos como funcionam as interrupções de hardware. Para esse projeto, você precisará:

- 2 LEDs vermelho difuso
- LED amarelo difuso
- 2 LED sverde difuso
- Resistor de 150 Ohms
- 4 resistores de 300 Ohms e alguns fios de conexão.
- Botão interruptor tátil

O resistor de 150  $\Omega$  é para o botão; ele é conhecido como um resistor pull-down<sup>1</sup>. A Figura abaixo ilustra o esquemático a ser montado.

Implemente o circuito da seguinte maneira:



Quando você executar o programa, ele iniciará com o semáforo no verde, para permitir que os carros passem, e a luz para pedestres no vermelho.

Quando você pressionar o botão, o programa verificará se, ao menos, cinco segundos transcorreram desde a última vez em que o semáforo mudou (para permitir que o trânsito flua) e, se afirmativo, executará o código para a função que você criou, `changeLights()`. Nessa função, o semáforo vai de verde para amarelo e depois para vermelho, e o semáforo para pedestres vai para verde.

Depois de um intervalo de tempo definido na variável **crossTime** (tempo suficiente para que os pedestres atravessem), a luz verde para pedestres pisca, acendendo e apagando, para avisar aos pedestres que atravessem rapidamente antes que o semáforo feche. Então, a luz vermelha do semáforo para pedestres acende, e a luz para os veículos vai de vermelho para verde, e finalmente para verde, permitindo novamente o fluxo do tráfego.

<sup>1</sup> **Resistor Pull-Down:** O novo componente de hardware apresentado na parte 2 do projeto é o botão, ou interruptor tátil. Ao analisar o circuito, o botão não está diretamente conectado entre a linha de alimentação e o pino de entrada; há um resistor entre o botão e o barramento do terra. Isso é o que chamamos de um resistor pull-down, elemento essencial para garantir que botões funcionem corretamente.

Em um circuito pull-down, quando o botão for pressionado, a eletricidade tomará o caminho de menor resistência, e fluirá entre os 5 V e o pino de entrada (há um resistor de 100  $\Omega$  no pino de entrada e um resistor de 10 k $\Omega$  no terra). Entretanto, quando o botão não é pressionado, a entrada está conectada ao resistor de 100 k $\Omega$  e é direcionada para o terra. Sem isso, o pino de entrada ficaria conectado a nada quando o botão não estivesse pressionado, e flutuaria entre 0 V e 5 V. Nesse circuito, a entrada sempre será direcionada para o terra, ou zero volt, quando o botão não estiver pressionado, e para 5 V quando ele estiver pressionado. Em outras palavras, com isso você impede que o pino flutue entre dois valores.