Nome: Túlio Brant Silva Guerra

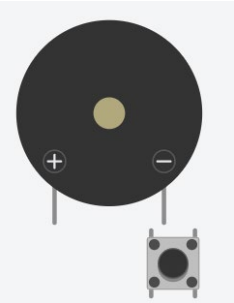
Semáforo: Veículos e Pedestres

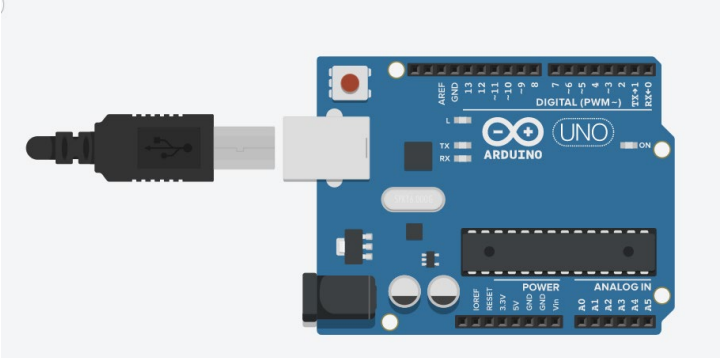
1. Resumo

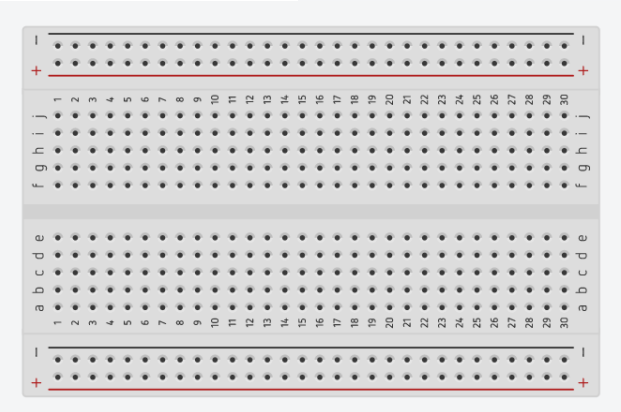
O projeto se trata da construção de um circuito elétrico utilizando as ferramentas do Tinkercad. O circuito irá simular um semáforo de veículos (vermelho, amarelo, verde) e um semáforo de pedestre (vermelho e verde). Além disso, instalaremos tanto um botão para a abertura imediata do sinal, quanto um dispositivo de alerta sonoro.

2. Setup

A peças utilizadas no circuito foram um Arduino, cinco lâmpadas LED, três resistores, uma placa de ensaio, um botão e um piezo.







3. Montagem

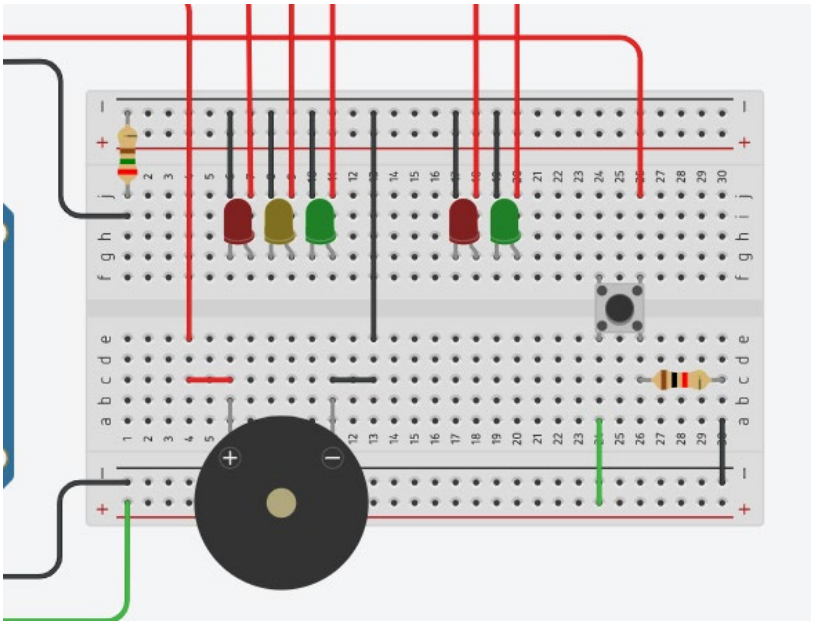
Estarei utilizando uma placa de ensaio para encaixar todos os componentes:

Leds de veículos: Conectados nas portas digitais (10,9,8) e porta ground.

Leds de pedestres: Conectados nas portas digitais (4, 2) e porta ground.

Botão: Conectado na porta 5v, porta digital 12 e ground.

Piezo: Conectado na porta digital 13 e porta ground.



Problema 1: O maior problema neste projeto foi a organização. Precisei desmontar várias vezes o circuito devido a péssima disposição dos fios. O resultado final, o qual também não fiquei totalmente satisfeito, está representado acima.

4. Código

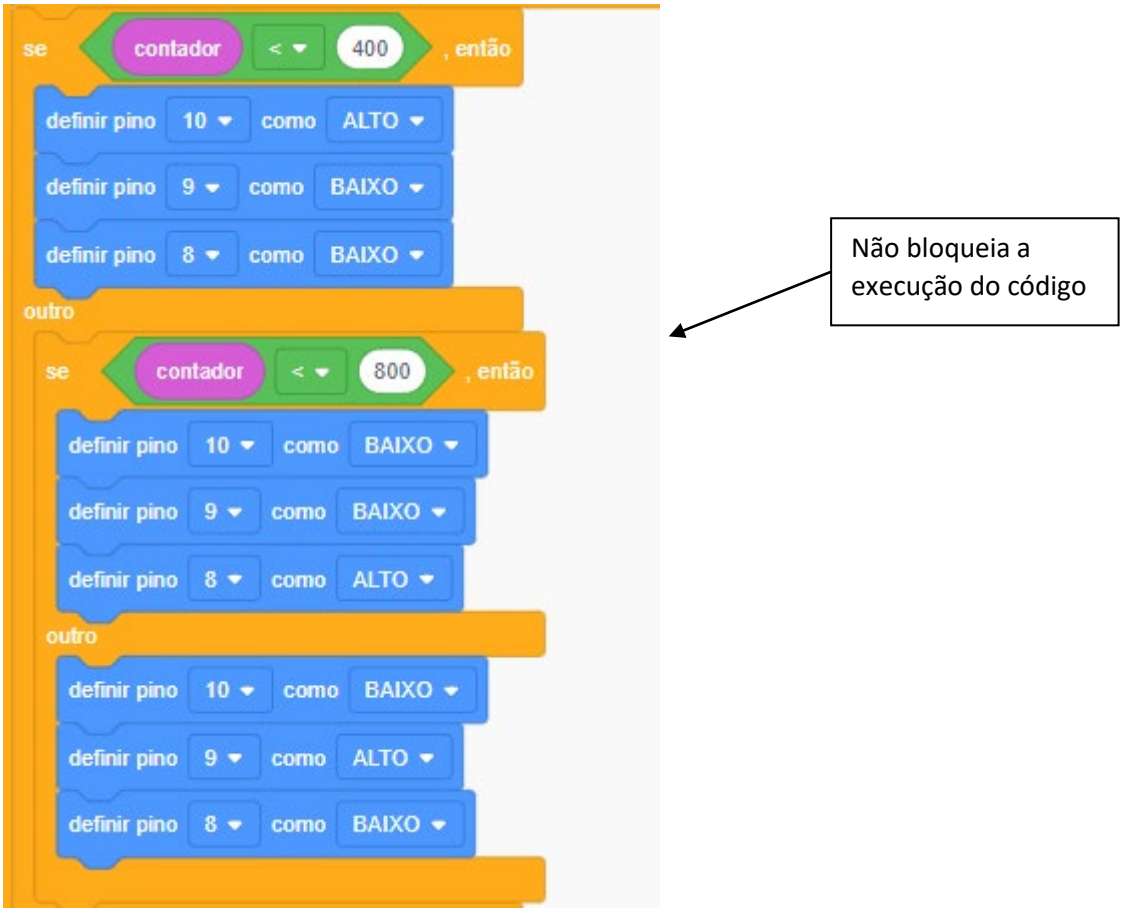
A minha primeira tentativa de programar os sinais foi reutilizando a lógica do trabalho anterior: o código inicia com o sinal dos veículos fechado e o dos pedestres aberto. Quando o tempo finaliza, invertemos e esperamos (3s para pedestres, 7s para carros).

Visualmente o resultado era o esperado, contudo foi quando tentei implementar o botão que percebi a falha. O código exemplificado acima utiliza a função wait(), uma função que trava a execução do código por um determinado tempo. Por causa dessa função, era impossível checar o tempo todo se o botão foi pressionado ou não.

Era necessário implementar uma lógica que não trave a execução do código.

Solução: utilizar variáveis para comparar o estado do sinal e determinar quais leds ativar.

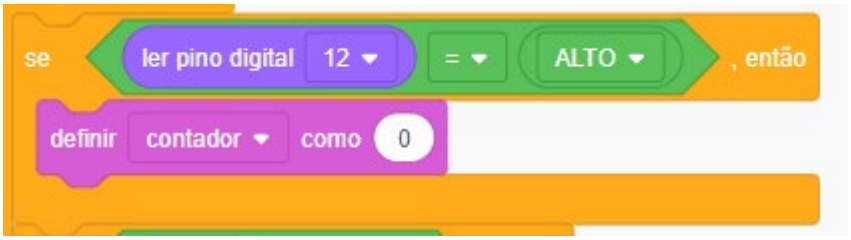
Nova estrutura do código:



Obs: Contador incrementa em um a cada iteração do código, por isso para ajustar a velocidade dos sinais preciso aumentar o valor da checagem (ex: contador < 500).

Quando o contador chega no valor limite ele reseta.

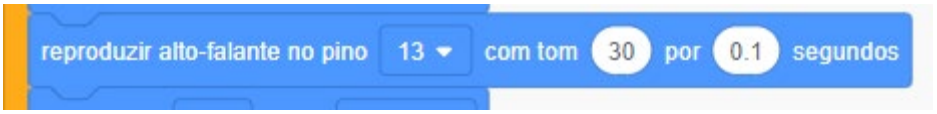
Checa se o botão foi pressionado (agora roda várias vezes por segundo):



Checa se o contador chegou no valor limite (caso sim, zere o contador).

No final da execução, SEMPRE incrementar o contador:

Função para reproduzir o alerta sonoro:



Problema 2: O piezo sempre emite um ruído baixo mesmo quando não acionado

(Não consegui solucionar o problema)

6. Resultado

Circuito para análise: https://www.tinkercad.com/things/clcgwj6LtIu?sharecode=MGGTH76zmFoMIwpDCx1EoRFbsbeTaJ968XtDFxbCZqQ