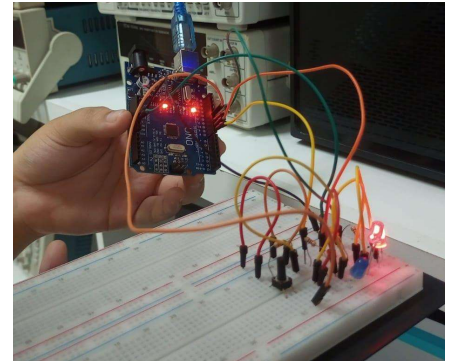
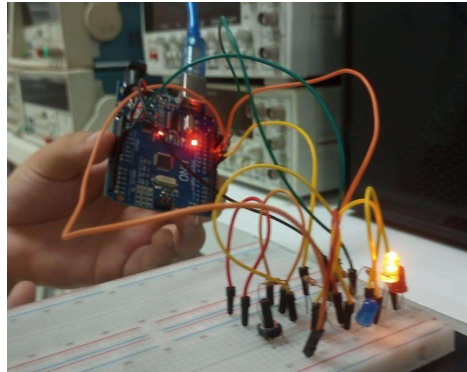
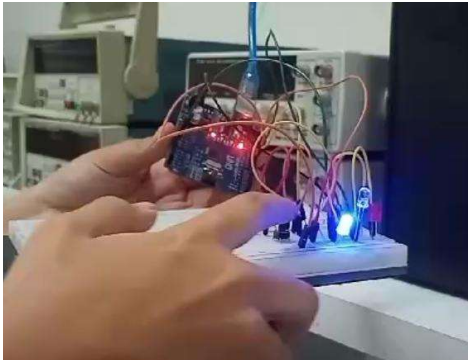


Além disso, o código foi aprimorado com a introdução de uma abordagem modular. Essa melhoria segue boas práticas de programação ao separar a lógica específica do semáforo em uma função própria. Essa organização não apenas torna o código mais legível, mas também facilita a reutilização em outros projetos, contribuindo para a eficiência do desenvolvimento.

Dessa forma, o projeto vai além de simular um semáforo comum ao integrar uma funcionalidade específica para a travessia de pedestres. Essa adição torna o sistema mais versátil e alinhado com as exigências contemporâneas de segurança no trânsito, representando um avanço prático e educativo na aplicação de tecnologia em situações do cotidiano.



Material Utilizado:

- Arduino Uno
- 1 x Resistor 10k Ω
- 6 x Resistores 220 Ω
- 1 x Chave Táctil Push-button
- 1 x LED Vermelho
- 1 x LED Amarelo
- 1 x LED Verde(azul)
- Protoboard
- Jumpers

Significado de Cada Parte do Código:

1. Declaração de Portas:
 - ``int vermelho = 10;``
 - ``int amarelo = 9;``
 - ``int verde(azul) = 8;``
 - ``int botao = 12;``

Associação de variáveis às portas específicas do Arduino.

2. Configuração Inicial:
 - ``pinMode``: Configuração das portas como saída (LEDs) ou entrada (botão).
 - Inicialização do semáforo com o verde(azul) ligado e os demais desligados.
3. Função ``loop``:
 - ``digitalRead(botao) == HIGH``: Verifica se o botão foi pressionado.
 - Se verdadeiro, aguarda um curto período e chama a função ``semaforo``.
4. Função ``semaforo``:
 - Alterna o estado dos LEDs, simulando a lógica de um semáforo.
 - ``delay``: Introduce pausas para representar os tempos de espera entre transições.

Ao pressionar o botão, o semáforo passa por um ciclo controlado de verde(azul) para amarelo, depois vermelho, e finalmente volta a verde(azul), permitindo uma travessia segura de pedestres.