

Semáforo para carros e pedestres

**Gabriel Vargas Azarias da Silva
Guilherme Alexandre de Souza Wilke
João Pedro Pires Ribeiro**

**Projeto Final
ELT72B - Sistemas Digitais
Prof. Adriano Ruseler
Julho, 2025**



Sumário

1. Introdução
2. Lista de materiais
3. Esquemáticos e hardware utilizado
4. Diagrama em blocos do projeto
5. Código utilizado no arduíno
6. Resultados obtidos (protótipo)

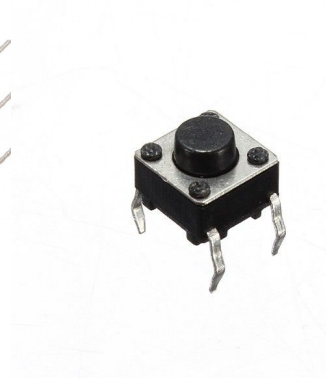
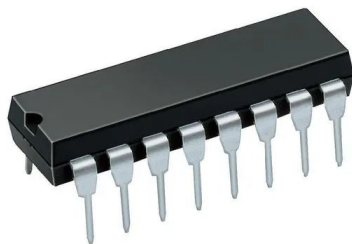
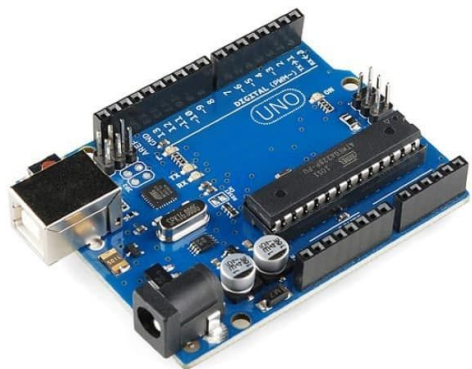
1. Introdução

O objetivo do projeto é fazer um semáforo para carros e um para pedestres, onde o semáforo para os pedestres têm a preferência. A ideia é que haja uma alternância contínua entre os semáforos, mas caso haja pedestres e estes apertem o botão solicitando a passagem, o sinal fique verde para eles.



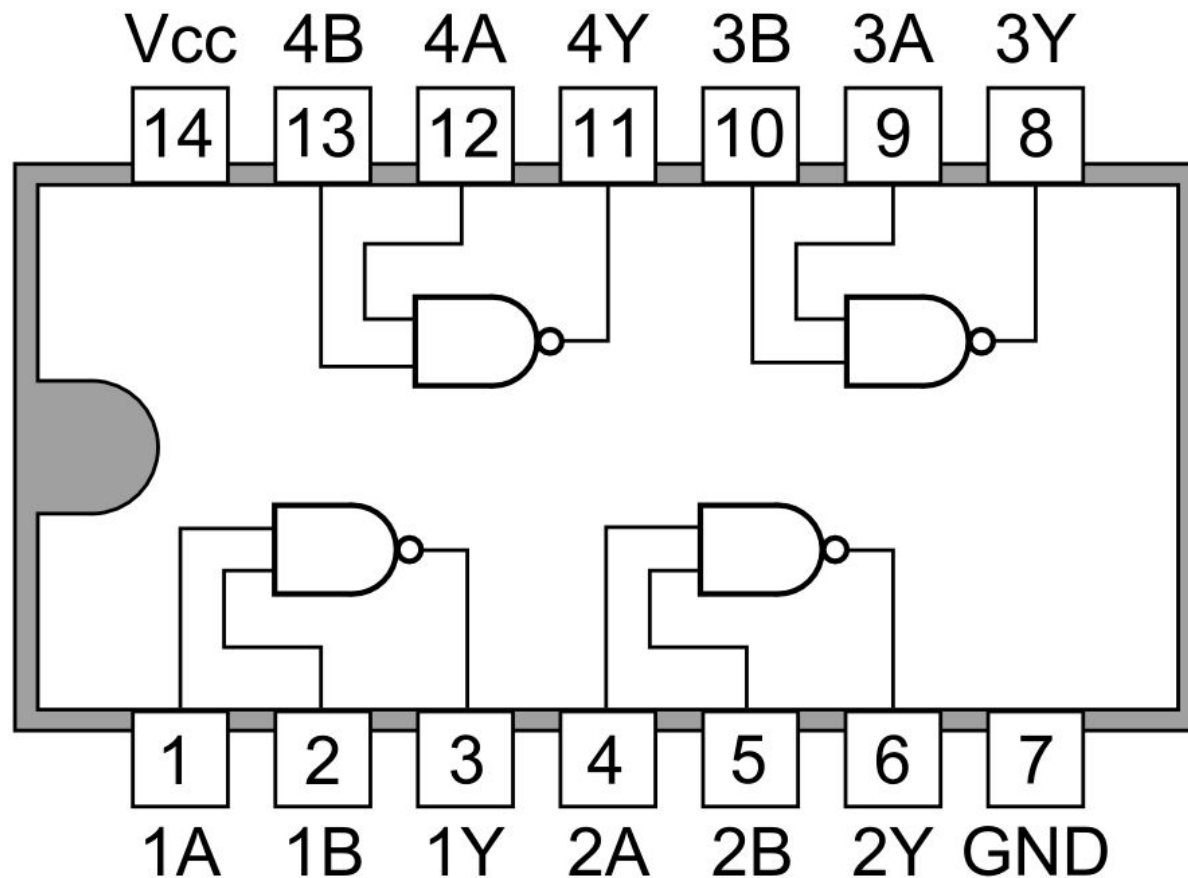
2. Lista de materias

- 1x Arduino Uno
- 2x CI 7400 (NAND)
- 5x LEDs (dois verdes, dois vermelhos e um amarelo)
- 1x Resistor de $10k\Omega$
- 5x Resistores de 220Ω
- 1x Botão de pressão (Push-button)

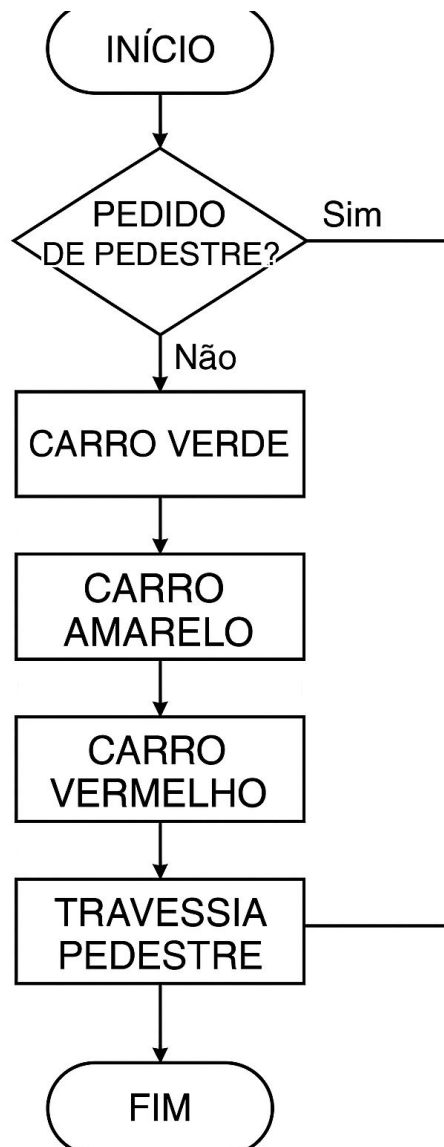


3. Esquemáticos e hardware utilizado

7400 Quad 2-input NAND Gates



4. Diagrama em blocos do projeto



5. Código utilizado no arduíno

```
1 // --- Definições dos Pinos ---
2 const int pinoS1 = 10; // Controla o bit 1 do estado
3 const int pinoS0 = 9;  // Controla o bit 0 do estado
4 const int pinoBotao = 2; // Lê o botão de pedestre
5
6 // --- Variável de Controle ---
7 // Guarda se o pedestre pediu para atravessar
8 bool pedidoPedestre = false;
9
10 // --- Setup: Roda apenas uma vez ---
11 void setup() {
12   pinMode(pinoS1, OUTPUT);
13   pinMode(pinoS0, OUTPUT);
14   pinMode(pinoBotao, INPUT);
15 }
16
17 // --- Função Auxiliar para Mudar o Estado ---
18 void definirEstado(int estado) {
19   // Envia os sinais para o circuito lógico
20   digitalWrite(pinoS1, (estado & 2) ? HIGH : LOW);
21   digitalWrite(pinoS0, (estado & 1) ? HIGH : LOW);
22 }
23
24 // --- Loop Principal: Roda para sempre ---
```

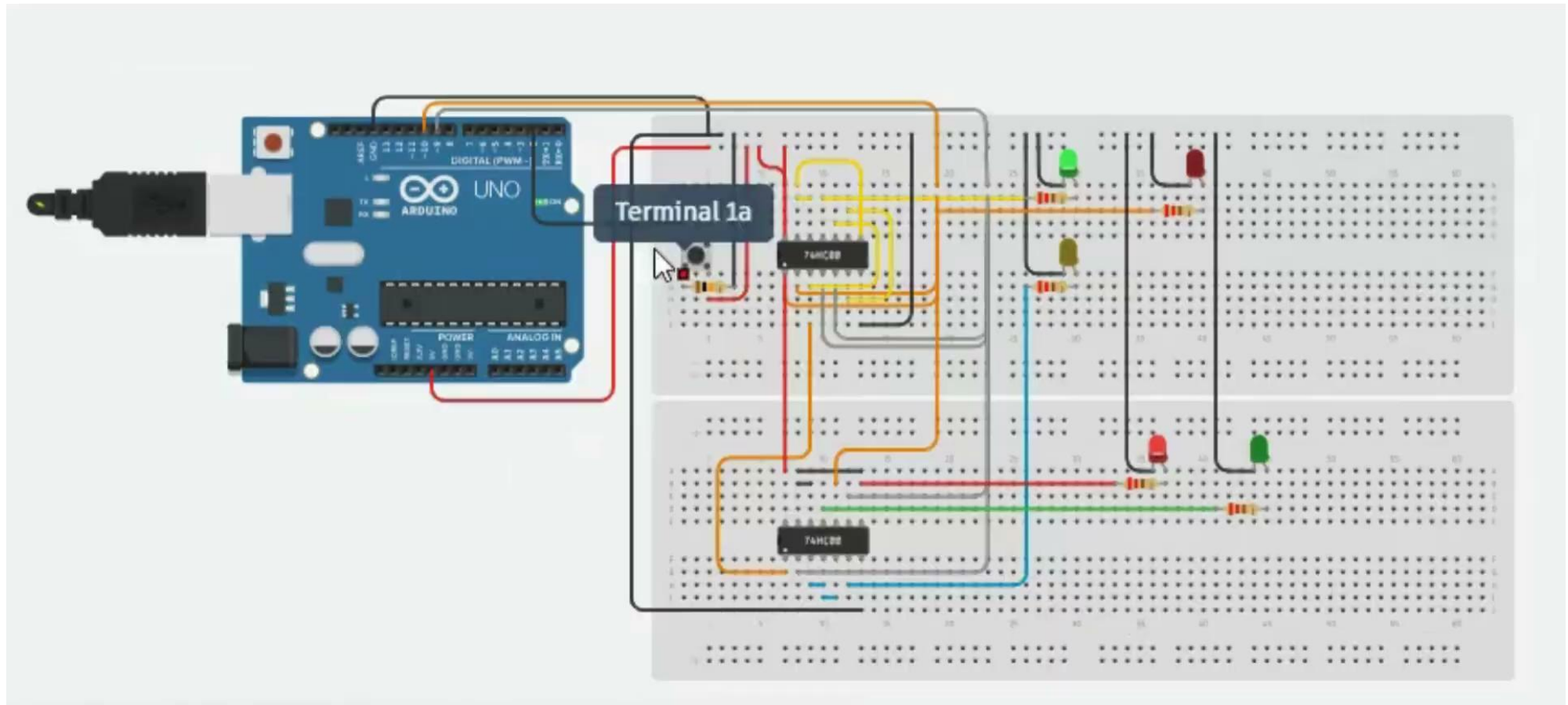

5. Código utilizado no arduíno

```
25 void loop() {
26   // A qualquer momento, se o botão for lido, a variável pedidoPedestre
27   // pode ser atualizada. A verificação principal acontece dentro do loop do verde.
28
29   // --- Ciclo do Semáforo ---
30
31   // Estado 0: Carro Verde (por até 15 segundos)
32   definirEstado(0);
33
34   // ===== A ALTERAÇÃO ESTÁ AQUI =====
35   // Este loop substitui o antigo 'delay(15000);'.
36   // Ele espera por 15 segundos, mas verifica o botão a cada 0.1 segundo.
37   for (int i = 0; i < 150; i++) {
38     if (digitalRead(pinoBotao) == HIGH) {
39       pedidoPedestre = true; // Anota que o pedido foi feito
40       break; // 'break' interrompe o loop do verde IMEDIATAMENTE!
41     }
42     delay(100); // Pausa por 0.1 segundo antes de verificar de novo
43   }
44   // =====
45
46   // Estado 1: Carro Amarelo (por 3 segundos)
47   // O programa segue para cá, seja após os 15s ou após o botão interromper o verde.
48   definirEstado(1);
```


5. Código utilizado no arduíno

```
49  delay(3000);
50
51  // Estado 2: Carro Vermelho
52  definirEstado(2);
53
54  // Ponto de Decisão: Verifica se há um pedido de pedestre para ser atendido.
55  if (pedidoPedestre == true) {
56      delay(1000); // Pausa de segurança
57
58      // Estado 3: Travessia de Pedestre (por 10 segundos)
59      definirEstado(3);
60      delay(10000);
61
62      pedidoPedestre = false; // Limpa o pedido, pois já foi atendido.
63
64      // Volta ao estado com tudo vermelho por segurança
65      definirEstado(2);
66      delay(2000);
67  } else {
68      // Se nenhum botão foi pressionado em nenhum momento, fica no vermelho por 5s
69      delay(5000);
70  }
71
72  // O ciclo recomeça
```

6. Resultados obtidos (protótipo)



6. Resultados obtidos (protótipo)

