



**UNIVERSIDADE DA INTEGRAÇÃO INTERNACIONAL
DA LUSOFONIA AFRO-BRASILEIRA**

INSTITUTO DE ENGENHARIAS E DESENVOLVIMENTO SUSTENTÁVEL

BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO

**GABRIEL PAIXÃO DE OLIVEIRA
ANTONIO GERLE TEÓFILO DA SILVA
ELÊNCIO CALADO CUSTUME ZIVANE**

MICROCONTROLADORES E DSPS

ESTACIONAMENTO INTELIGENTE COM ARDUINO UNO

REDENÇÃO

2024

SUMÁRIO

1 INTRODUÇÃO	3
2 OBJETIVO	3
3 MATERIAIS	3
4 PROCEDIMENTOS DE MONTAGEM	4
4.1 Simulação	4
4.2 Prática	6
5 FUNCIONAMENTO LÓGICO	7
6 RESULTADOS E DISCUSSÃO	10
6 CONCLUSÃO	10
7 REFERÊNCIAS	11

1 INTRODUÇÃO

Uma tarefa muito comum para quem sai de casa com um automóvel próprio é a de procurar uma vaga para estacioná-lo quando se chega ao seu destino. Ainda que seja um evento bastante recorrente na vida de diversas pessoas, poucos são os estacionamentos que possuem artifícios para facilitar essa tarefa. Tal cenário, somado ao estresse que causam os engarrafamentos, agrava ainda mais a experiência dos condutores automobilísticos, tendo-se em vista que ainda poderá ser necessário dar várias voltas pelos quarteirões do seu destino até que se visualize um local para parar seu veículo. Nesse contexto, o presente trabalho visa à construção de um sistema eletrônico, baseado no Arduino Uno, capaz de sinalizar de forma múltipla e direta os espaços vagos e ocupados dentro de uma área determinada, a fim de ser o protótipo de uma tecnologia apta a instaurar um estacionamento inteligente.

2 OBJETIVO

- Construir de um sistema eletrônico, baseado no Arduino Uno, capaz de sinalizar de forma múltipla e direta os espaços vagos e ocupados dentro de uma área determinada, a fim de ser o protótipo de uma tecnologia apta a instaurar um estacionamento inteligente.

3 MATERIAIS

- 1 Notebook.
- 1 Arduino Uno.
- 1 LCD 16x2.
- 1 Protoboard.
- 2 Sensores ultrassônicos.
- 27 Jumpers.
- 2 LEDs vermelhos.
- 2 LEDs verdes.
- 4 Resistores de 330 ohms.
- 1 Resistor de 220 ohms.

4 PROCEDIMENTOS DE MONTAGEM

O projeto foi realizado em duas partes. A primeira utilizou-se da simulação dos equipamentos fornecida pelo site tinkercad.com para planejar as conexões e o funcionamento do sistema. A segunda procurou reproduzir, com os materiais do Laboratório de Robótica, o que já havia sido planejado computacionalmente na etapa anterior, a fim de tornar o sistema sensível ao mundo físico.

4.1 Simulação

A montagem do circuito no tinkercad.com foi realizada utilizando o seguinte esquema:

Conexões do Display LCD:

VSS (Pino 1 do LCD): Conectado ao GND (terra) na protoboard.

VDD (Pino 2 do LCD): Conectado ao VCC (5V) na protoboard.

V0 (Pino 3 do LCD): Conectado ao pino digital 6 do Arduino.

RS (Pino 4 do LCD): Conectado ao pino digital 12 do Arduino.

RW (Pino 5 do LCD): Conectado ao GND.

E (Pino 6 do LCD): Conectado ao pino digital 11 do Arduino.

D4, D5, D6, D7 (Pinos 11, 12, 13, 14 do LCD): Conectados aos pinos digitais 5, 4, 3, 2 do Arduino, respectivamente.

A (Anodo do backlight): Conectado ao VCC via um resistor de 200 ohms.

K (Catodo do backlight): Conectado ao GND.

Conexões dos Sensores Ultrassônicos:

VCC (Pino 1): Conectado ao VCC (5V) na protoboard.

Trig (Pino 2): Conectado aos pinos analógicos A1 e A5 do Arduino (um para cada sensor).

Echo (Pino 3): Conectado aos pinos analógicos A0 e A4 do Arduino (um para cada sensor).

GND (Pino 4): Conectado ao GND na protoboard.

Conexões dos LEDs:

Anodo (lado mais longo) dos LEDs Vermelhos: Conectado ao pino analógico A2 e o outro ao pino digital 10 do Arduino.

Catodo (lado mais curto) dos LEDs Vermelhos: Conectado ao GND na protoboard.

Anodo (lado mais longo) dos LEDs Verdes: Conectado ao pino analógico A3 e o outro ao pino digital 9 do Arduino.

Catodo (lado mais curto) dos LEDs Verdes: Conectado ao GND na protoboard.

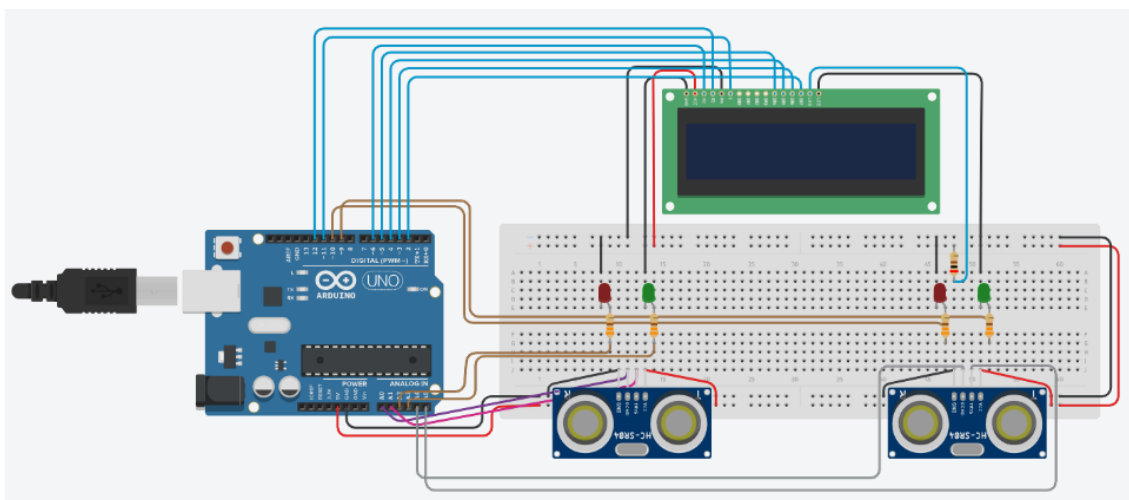


Figura 1. Representação esquemática do sistema projetado computacionalmente.

4.2 Prática

Conforme explicado anteriormente, a montagem prática decorreu da utilização dos materiais do Laboratório de Robótica e do esquema projetado na última etapa. Dessa maneira, todas as conexões permaneceram muito similares na protoboard, sendo as entradas no Arduino idênticas ao esperado.

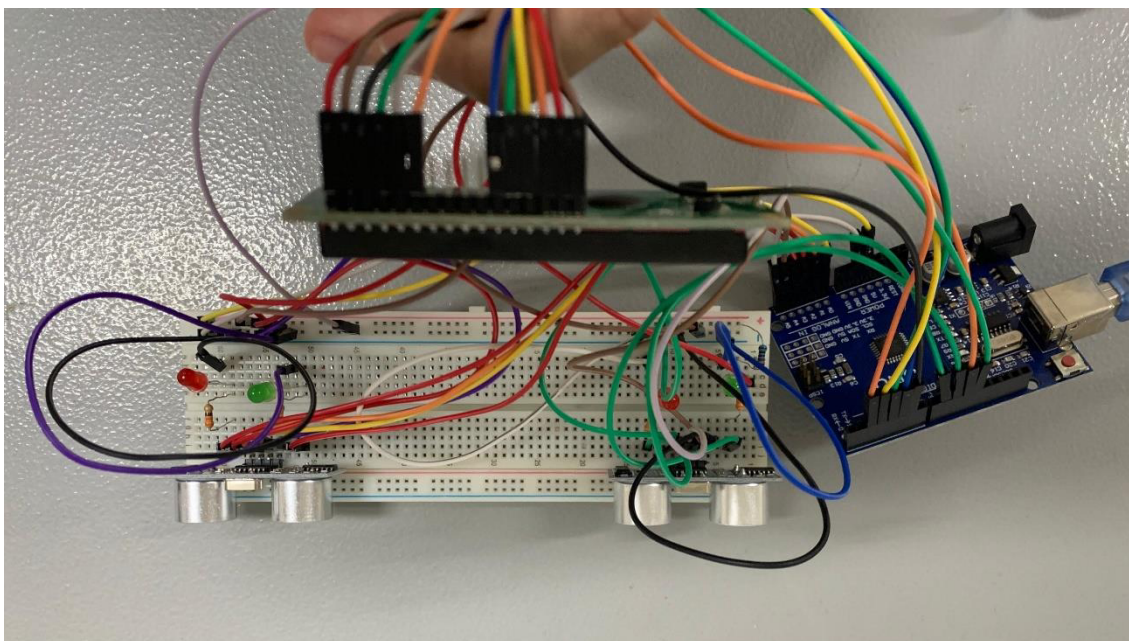


Figura 2. Visão superior do sistema montado na prática.

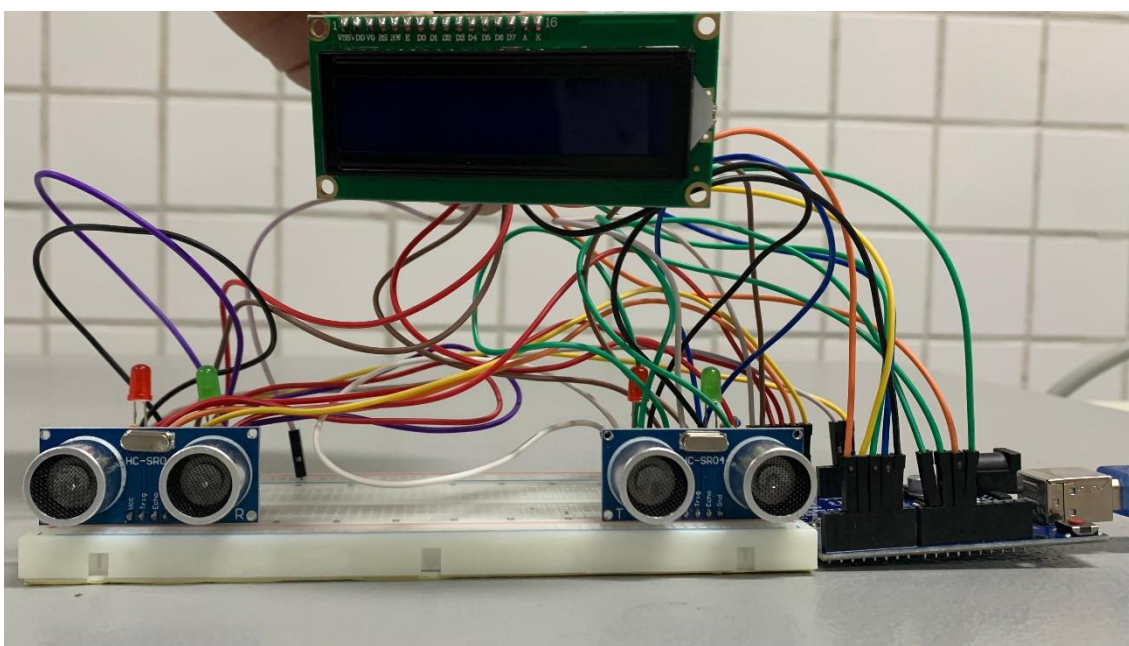


Figura 3. Visão frontal do sistema montado na prática.

5 FUNCIONAMENTO LÓGICO

Para explicar o funcionamento lógico do projeto, é de maior necessidade que seja analisado o programa gravado no Arduino Uno que opera o sistema construído de maneira a atender o objetivo do trabalho.

Nesse sentido, o código faz uso da biblioteca LiquidCrystal para controlar o display LCD e mede a distância utilizando os sensores ultrassônicos. Com base na distância medida, o código acende LEDs vermelhos e verdes para indicar se a vaga está ocupada ou livre.

Lógica de Funcionamento

1. Setup Inicial:

- Configura os pinos de TRIG, ECHO, LEDs e LCD.
- Inicializa o LCD.
- Define o contraste do LCD.
- Exibe no LCD os títulos para cada vaga: "VAGA 1" e "VAGA 2".

2. Loop Principal:

- Medição da Distância:
 - Para cada sensor, a distância é medida enviando um pulso de 10 microsegundos pelo pino TRIG e lendo o tempo de retorno no pino ECHO.
 - A distância é calculada utilizando a fórmula: $distance = (duration * 0,0343) / 2$
- Indicação de Vaga:
 - Sensor 1:
 - Se a distância medida for menor ou igual a 10 cm, considera a vaga ocupada e:
 - Acende o LED vermelho.
 - Apaga o LED verde.
 - Atualiza o LCD para exibir "CHEIO".
 - Se a distância for maior que 10 cm, considera a vaga livre e:
 - Apaga o LED vermelho.
 - Acende o LED verde.

- Atualiza o LCD para exibir "LIVRE".
- Sensor 2:
 - Segue a mesma lógica do Sensor 1 para determinar o estado da vaga.
- Atraso de 500ms:
 - Insere um atraso de 500 milissegundos antes de realizar a próxima medição.

Fluxo de Operação

1. Inicialização:

- O Arduino configura os pinos e inicializa o LCD para exibir as vagas.

2. Medição Contínua:

- Os sensores ultrassônicos continuamente medem a distância dos objetos (possíveis carros) na frente deles.
- A cada 500ms, o Arduino processa essas medições para determinar se as vagas estão ocupadas ou livres.

3. Atualização dos Indicadores:

- Dependendo da distância medida, o Arduino atualiza os LEDs e o LCD para refletir o estado atual das vagas de estacionamento.

Este sistema permite monitorar em tempo real a ocupação das vagas, facilitando a gestão do estacionamento e ajudando os motoristas a encontrar vagas disponíveis de maneira eficiente.

Figura 4. Parte 1 do código.

```

1  #include <LiquidCrystal.h>
2
3  int Contrast = 100;
4  LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
5  // Sensor 1
6  const int trigPin1 = A1; // Pino TRIG do sensor ultrassônico 1
7  const int echoPin1 = A0; // Pino ECHO do sensor ultrassônico 1
8  const int ledRed1 = A2; // Pino do LED vermelho para sensor 1
9  const int ledGreen1 = A3; // Pino do LED verde para sensor 1
10
11 // Sensor 2
12 const int trigPin2 = A5; // Pino TRIG do sensor ultrassônico 2
13 const int echoPin2 = A4; // Pino ECHO do sensor ultrassônico 2
14 const int ledRed2 = 10; // Pino do LED vermelho para sensor 2
15 const int ledGreen2 = 9; // Pino do LED verde para sensor 2
16
17 void setup() {
18   // Configurações do sensor 1
19   pinMode(trigPin1, OUTPUT); // Define o pino TRIG como saída
20   pinMode(echoPin1, INPUT); // Define o pino ECHO como entrada
21   pinMode(ledRed1, OUTPUT); // Define o pino do LED vermelho como saída
22   pinMode(ledGreen1, OUTPUT); // Define o pino do LED verde como saída
23
24   // Configurações do sensor 2
25   pinMode(trigPin2, OUTPUT); // Define o pino TRIG como saída
26   pinMode(echoPin2, INPUT); // Define o pino ECHO como entrada
  
```


Figura 5. Parte 2 do código utilizado.

```

26 pinMode(echoPin2, INPUT); // Define o pino ECHO como entrada
27 pinMode(ledRed2, OUTPUT); // Define o pino do LED vermelho como saída
28 pinMode(ledGreen2, OUTPUT); // Define o pino do LED verde como saída
29
30 analogWrite(6, Contrast);
31 lcd.begin(16, 2);
32 delay(100);
33 lcd.setCursor(0, 0);
34 lcd.print("|VAGA 1||VAGA 2|");
35 lcd.setCursor(0, 1);
36 lcd.print("|      ||      |");
37
38 }
39
40
41 void loop() {
42   // Leitura do sensor 1
43   long duration1;
44   float distance1;
45
46
47   digitalWrite(trigPin1, LOW);
48   delayMicroseconds(2);
49   digitalWrite(trigPin1, HIGH);
50   delayMicroseconds(10);
51   digitalWrite(trigPin1, LOW);

```

Figura 6. Parte 3 do código utilizado.

```

51 digitalWrite(trigPin1, LOW);
52 duration1 = pulseIn(echoPin1, HIGH);
53 distance1 = (duration1 * 0.0343) / 2;
54
55 Serial.print("Sensor 1 - Distância: ");
56 Serial.print(distance1);
57 Serial.println(" cm");
58
59 if (distance1 > 0 && distance1 <= 10) {
60   digitalWrite(ledRed1, HIGH); // Acende o LED vermelho
61   digitalWrite(ledGreen1, LOW); // Apaga o LED verde
62   lcd.setCursor(1, 1);
63   lcd.print("CHEIO");
64 } else {
65   digitalWrite(ledRed1, LOW); // Apaga o LED vermelho
66   digitalWrite(ledGreen1, HIGH); // Acende o LED verde
67   lcd.setCursor(1, 1);
68   lcd.print("LIVRE");
69 }
70
71 // Leitura do sensor 2
72 long duration2;
73 float distance2;
74
75 digitalWrite(trigPin2, LOW);
76 delayMicroseconds(2);

```

Figura 7. Parte 4 do código utilizado.

```

76 delayMicroseconds(2);
77 digitalWrite(trigPin2, HIGH);
78 delayMicroseconds(10);
79 digitalWrite(trigPin2, LOW);
80 duration2 = pulseIn(echoPin2, HIGH);
81 distance2 = (duration2 * 0.0343) / 2;
82
83 Serial.print("Sensor 2 - Distância: ");
84 Serial.print(distance2);
85 Serial.println(" cm");
86
87 if (distance2 > 0 && distance2 <= 10) {
88   digitalWrite(ledRed2, HIGH); // Acende o LED vermelho
89   digitalWrite(ledGreen2, LOW); // Apaga o LED verde
90   lcd.setCursor(9, 1);
91   lcd.print("CHEIO");
92 } else {
93   digitalWrite(ledRed2, LOW); // Apaga o LED vermelho
94   digitalWrite(ledGreen2, HIGH); // Acende o LED verde
95   lcd.setCursor(9, 1);
96   lcd.print("LIVRE");
97 }
98
99 delay(500); // Aguarda meio segundo antes de medir novamente
100
101

```

6 RESULTADOS E DISCUSSÃO

O projeto foi finalizado alcançando pleno funcionamento. O teste foi gravado e pode ser acessado para visualização por meio deste link: <https://youtube.com/shorts/NRfeo6ia5ls?feature=share>.

Pela gravação, é possível observar o funcionamento esperado de um protótipo simples para estacionamento inteligente, onde o led verde aceso indica a vaga desocupada, o led vermelho indica a vaga ocupada, e o LCD atualiza a situação da vaga, sendo o sensor ultrassônico o meio pelo qual os estados da vaga são alterados.

Nesse sentido, a proposta é que as informações do LCD estejam disponíveis antes que o motorista precise procurar exaustivamente uma vaga, e que o LED fortaleça ainda mais a percepção rápida de espaços disponíveis num estacionamento.

Ainda assim, mesmo que o protótipo tenha sido concluído com êxito em seu objetivo, esses foram apenas os recursos possíveis para ele dentro do prazo estabelecido e das condições materiais disponíveis. Dados melhores recursos, o projeto poderia também conter um aplicativo que tornasse disponíveis informações sobre a situação de vagas de forma mais remota e cômoda para os clientes do estacionamento.

6 CONCLUSÃO

O projeto atingiu o seu objetivo satisfatoriamente. O protótipo simples, mas eficaz, valida a viabilidade da proposta de fornecer informações em tempo real sobre a ocupação das vagas, facilitando a vida dos motoristas ao evitar a busca exaustiva por um espaço disponível. A combinação dos LEDs e do display LCD garante uma percepção rápida e clara do estado das vagas, melhorando a eficiência e a experiência do usuário no estacionamento. A elaboração do protótipo seguiu a etapa de simulação e prática, bem como a do sequenciamento lógico fornecida pela escrita do código. Embora o projeto tenha atingido seu objetivo com os recursos e o prazo disponíveis, há potencial para melhorias significativas.

7 REFERÊNCIAS

BANDEIRA, Thayanne Barros; FONSECA, Wellington; FERNANDES, Filipe Cavalcanti; et al. **Protótipo de estacionamento automatizado utilizando modelo computacional matricial e microcontrolador Arduino**. In: Congresso Nacional de Matemática Aplicada à Indústria, novembro de 2014. DOI: 10.5151/mathpro-cnmai-0151.

Educational Engineering Team. **How to add any Arduino Library to Tinkercad for Simulation**. 2021. Disponível em: <<https://www.youtube.com/watch?v=id2Q-N3YmKU>>. Acesso em: 8 jul. 2024.

JuliaLabs. **Sensor de Distância Ultrassônico – Como Usar no Arduino**. 2020. Disponível em: <https://www.youtube.com/watch?v=J_bUAs-VXA8>. Acesso em: 8 jul. 2024.

Young Mind Creations. **How to interface LCD display to Arduino without potentiometer /Arduino Project**. 2022. Disponível em: <<https://www.youtube.com/watch?v=IM3d0dlAOJk>>.