

Implementação de sensor de distância com sinal sonoro

Elias de Almeida Sombra Neto¹

¹Instituto Federal de Educação, Ciência e Tecnologia do Ceará (IFCE) - Campus Maracanaú
Av. Parque Central, 1315 - Distrito Industrial I, Maracanaú-CE, Brasil

elias.almeida09@aluno.ifce.edu.br

Abstract. *This work presents the implementation of an embedded system using an HC-SR04 distance sensor and a buzzer, controlled by an ESP32. The goal was to emit sound signals with frequency based on object proximity. Measurements are based on the median of 51 sorted readings. The system was successfully built and responded effectively to distance changes.*

Resumo. *Este trabalho apresenta a implementação de um sistema embarcado com sensor de distância HC-SR04 e buzzer, usando um ESP32. O objetivo foi emitir sinais sonoros com frequência proporcional à proximidade de um objeto. A medição é baseada na mediana de 51 leituras ordenadas. O sistema foi montado e programado com sucesso, respondendo de forma eficaz às variações de distância.*

1. Introdução

Este relatório descreve uma atividade prática realizada no Laboratório de Eletroeletrônica e Sistema Embarcados (LAESE) durante a disciplina de Microcontroladores no Instituto Federal do Ceará (IFCE).

O objetivo da atividade é implementar um sensor de distância em um circuito que emite sinal sonoro com maior frequência conforme menor o espaço entre o dispositivo e o objeto. A medição da distância será feita em centímetros (cm) e a medição do tempo será feito em milissegundos (ms).

A leitura do sensor será feita em um laço de repetição com número de iterações fixada em 51. As distâncias serão armazenadas em um array, o array será ordenado e a mediana atribuída a uma variável. Essa mediana será a medida de distância considerada para apitar o buzzer com determinada frequência. A partir de 100 cm o circuito sinaliza um bip de 500 ms a cada 100 ms, a partir de 50 cm o bip dura 200 ms a cada 50 ms, a partir de 20 cm o bip é emitido de forma constante.

2. Materiais utilizados

Os materiais para a construção e acionamento do circuito incluem:

- 1 ESP32 30 pinos;
- 1 cabo micro USB;
- 1 protoboard;
- 1 sensor de distância HC-SR04;
- 1 buzzer;
- 6 cabos jumper macho-fêmea.

3. Montagem do circuito

Para realizar a montagem do circuito, é preciso seguir os seguintes passos:

- Ligar o ESP32 em uma fonte de alimentação com o cabo micro USB.
- Posicionar o sensor de distância na região de componentes da protoboard.
- Ligar o VIN do ESP32 com o VCC do sensor de distância através de um jumper macho-fêmea.
- Ligar o GPIO5 do ESP32 com o pino trigger do sensor de distância através de um jumper macho-fêmea.
- Ligar o GPIO18 do ESP32 com o pino echo do sensor de distância através de um jumper macho-fêmea.
- Ligar o GND do ESP32 com o GND do sensor de distância através de um jumper macho-fêmea.
- Posicionar o buzzer na região de componentes da protoboard.
- Ligar o GPIO15 do ESP32 no pino positivo do buzzer através de um jumper macho-fêmea.
- Ligar o GND do ESP32 no pino negativo do buzzer através de um jumper macho-fêmea.

4. Implementação do código

O código responsável por medir a distância com o sensor HC-SR04 e emitir sinal sonoro com o buzzer foi desenvolvido com a linguagem C++ no editor de código Visual Studio Code juntamente com sua extensão PlatformIO.

```
1 #include <Arduino.h>
2
3 #define SOUND_SPEED 0.034
4 #define NUM_MEASUREMENTS 51
5
6 const int trigPin = 5;
7 const int echoPin = 18;
8 const int buzzerPin = 15;
9
10 long duration;
11 float distanceCm;
12 float distances[NUM_MEASUREMENTS];
13
14 void setup() {
15     Serial.begin(9600);
16     pinMode(trigPin, OUTPUT);
17     pinMode(echoPin, INPUT);
18     pinMode(buzzerPin, OUTPUT);
19 }
20
21 float measureDistance() {
22     digitalWrite(trigPin, LOW);
23     delayMicroseconds(2);
24     digitalWrite(trigPin, HIGH);
```

```
25     delayMicroseconds(10);
26     digitalWrite(trigPin, LOW);
27     duration = pulseIn(echoPin, HIGH);
28     return duration * SOUND_SPEED / 2;
29 }
30
31 void sortArray(float arr[], int size) {
32     for (int i = 0; i < size - 1; i++) {
33         for (int j = 0; j < size - i - 1; j++) {
34             if (arr[j] > arr[j + 1]) {
35                 float temp = arr[j];
36                 arr[j] = arr[j + 1];
37                 arr[j + 1] = temp;
38             }
39         }
40     }
41 }
42
43 void loop() {
44     for (int i = 0; i < NUM_MEASUREMENTS; i++) {
45         distances[i] = measureDistance();
46         delay(5);
47     }
48
49     sortArray(distances, NUM_MEASUREMENTS);
50     float median = distances[NUM_MEASUREMENTS / 2];
51
52     Serial.print("Mediana da distancia (cm): ");
53     Serial.println(median);
54
55     if (median < 20) {
56         digitalWrite(buzzerPin, HIGH);
57     }
58     else if (median < 50) {
59         digitalWrite(buzzerPin, HIGH);
60         delay(200);
61         digitalWrite(buzzerPin, LOW);
62         delay(50);
63     }
64     else if (median < 100) {
65         digitalWrite(buzzerPin, HIGH);
66         delay(500);
67         digitalWrite(buzzerPin, LOW);
68         delay(100);
69     }
70 }
```

5. Resultados

A montagem do circuito e implementação do código responsável por emitir bips com frequências diferentes conforme a distância foi concluída de forma bem sucedida. A seguir, pode-se visualizar a forma do circuito no final da prática, assim como os testes realizados e uma demonstração do seu funcionamento.

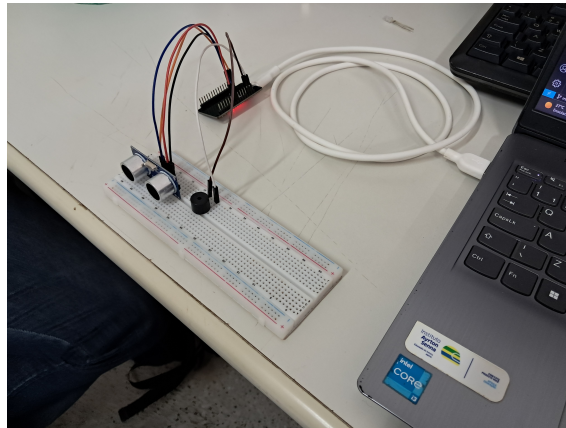


Figure 1. Circuito montado com um buzzer que sinaliza conforme a distância lida pelo sensor HC-SR04

```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  POSTGRESOL QUERY RESULTS
Mediana da distância (cm): 28.95
Mediana da distância (cm): 31.82
Mediana da distância (cm): 32.08
Mediana da distância (cm): 30.53
Mediana da distância (cm): 30.00
Mediana da distância (cm): 29.82
Mediana da distância (cm): 28.97
Mediana da distância (cm): 30.77
Mediana da distância (cm): 15.61
Mediana da distância (cm): 9.83
Mediana da distância (cm): 8.81
Mediana da distância (cm): 7.95
```

Figure 2. Resultados obtidos no terminal durante a leitura das distâncias

O vídeo que demonstra o circuito em funcionamento está disponível em: [Resultado da atividade prática 15](#)

6. Conclusão

Após a finalização dessa atividade prática em laboratório, foi possível projetar o circuito e desenvolver o código para emitir bips com frequências diferentes para um determinado intervalo de distância. Isso foi possível com o uso de um microcontrolador ESP32 e alguns componentes eletrônicos como uma protoboard, um sensor de distância, um buzzer e alguns cabos jumper. Assim, o objetivo proposto foi atingido com sucesso.