



**Criação de Sensor de Queda Para Idosos com IoT**

**Joana Scheer Loayza, Andre Luis de Oliveira**

<sup>1</sup>Universidade Presbiteriana Mackenzie (UPM)

Rua da Consolação, 930 Consolação, São Paulo - SP, 01302-907 – Brazil

10441259@mackenzista.com.br

***Abstract.***

*This work presents the development of an Internet of Things (IoT) based fall detection system for elderly people. The system uses an ESP32 microcontroller and an MOU6050 accelerometer/gyroscope sensor to monitor sudden movements that indicates a fall. When a fall is detected, the system sends real-time notifications to the caregiver via a mobile application, allowing immediate response. The purpose of this project is to increase the safety and autonomy of elderly individuals, reducing the risk of delayed assistance during accidents at home. The paper also discusses similar works in the field of IoT healthcare applications, the advantages of cloud-based monitoring, and the potential integration of wearable devices for improved elderly care. This project demonstrates how low-cost IoT devices can contribute to healthcare innovation, promoting both monitoring efficiency and quality of life.*

***Resumo.***

*Este trabalho apresenta o desenvolvimento de um sistema de detecção de queda para idosos baseado em Internet das Coisas (IoT). O sistema utiliza um microcontrolador ESP32 e um sensor acelerômetro/giroscópio MPU6050 para monitorar movimentos bruscos que caracterizam uma queda. Quando detectada, a queda gera uma notificação em tempo real para o cuidador por meio de um aplicativo móvel, possibilitando uma resposta imediata.*

*Nos últimos anos, a IoT expandiu-se para diversas áreas, incluindo saúde, cidades inteligentes, automação residencial e monitoramento industrial, oferecendo novos cenários de conectividade, segurança e eficiência.*

*O presente trabalho propõe o desenvolvimento de um dispositivo IoT portátil capaz de detectar quedas de idosos e enviar notificações imediatas a familiares ou cuidadores, promovendo maior segurança e autonomia. Diferentemente de soluções comerciais complexas, o projeto visa ser de baixo custo, simples de implementar e escalável, servindo como protótipo para futuras melhorias.*

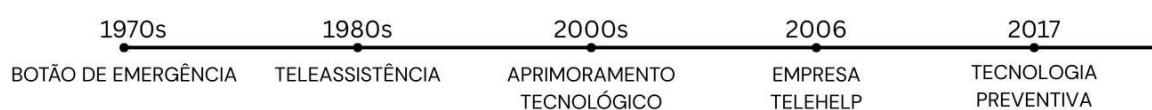
## Introdução

O conceito de conectar objetos do cotidiano à rede mundial de computadores começou a ganhar forma no final da década de 1990, com a popularização de sistemas embarcados e das primeiras aplicações de comunicação máquina a máquina (M2M). Desde então, avanços em microeletrônica, sensores, redes sem fio e computação em nuvem têm transformado a Internet das Coisas (IoT) em uma das tecnologias mais influentes da Quarta Revolução Industrial (ASHTON, 2009; GUBBI et al., 2013).

O crescimento da IoT tem impacto em diversas áreas, como saúde, indústria, agricultura e cidades inteligentes. Estima-se que, em 2023, existam cerca de 15 bilhões de dispositivos IoT ativos no mundo, com previsão de alcançar mais de 29 bilhões até 2030 (STATISTA, 2023). No setor da saúde, em especial, a IoT tem se consolidado como ferramenta fundamental para monitoramento remoto, prevenção de acidentes e melhoria da qualidade de vida de pacientes, principalmente idosos.

Quedas domésticas representam um dos principais riscos à saúde dessa população, frequentemente com consequências graves devido à demora na prestação de socorro. Pesquisas recentes exploram o uso de sensores vestíveis (wearables), acelerômetros e giroscópios para detecção de quedas. Bagala et al. (2012) demonstraram que algoritmos baseados em sinais de acelerômetros podem alcançar taxas de acurácia superiores a 90% na detecção de quedas. Kangas et al. (2008) analisaram diferentes posicionamentos de sensores no corpo humano, reforçando a viabilidade de soluções vestíveis e de baixo custo. Além disso, Dibner (2010) discutiu abordagens teóricas e práticas para o cuidado de idosos e pessoas com deficiência, enfatizando a importância da inovação tecnológica na assistência a essa população.

A Figura 1 apresenta uma linha do tempo que ilustra os principais marcos do desenvolvimento da IoT e sua aplicação na saúde, contextualizando o cenário no qual se insere o presente trabalho.



**Figura 1. Linha do tempo da IoT. Fonte: Autora**

## Objetivos

O presente trabalho tem como objetivo desenvolver um sistema de monitoramento de queda para idosos utilizando a plataforma ESP32, sensores de movimento e comunicação via MQTT.

O sistema pretende:

- Detectar quedas em tempo real através de um sensor acelerômetro (MPU-6050);
- Enviar notificações automáticas a familiares ou cuidadores utilizando rede Wi-Fi e protocolo MQTT;
- Atuar através de um buzzer (atuador sonoro) para alertar pessoas próximas quando uma queda ocorrer;
- Contribuir para a promoção do bem estar e segurança dos idosos, alinhando-se ao ODS 3 da ONU (Saúde e Bem Estar).

O problema que se busca solucionar é o tempo de resposta após uma queda, visto que muitos idosos vivem sozinhos e podem permanecer longos períodos sem auxílio, aumentando os riscos de complicações médicas.

## Materiais e Métodos

### *Plataforma de Prototipagem*

Para este projeto, será utilizada a placa ESP32, um microcontrolador de baixo custo com Wi-Fi e Bluetooth integrados, ideal para aplicações em Internet das Coisas (IoT).

- Processador: Dual-Core Tensilica LX6, até 240 MHz.
- Memória: 520 KB SRAM.
- Conectividade: Wi-Fi 802.11 b/g/n e Bluetooth 4.2.
- Portas: GPIOs, ADCs, PWMs e suporte a sensores externos.

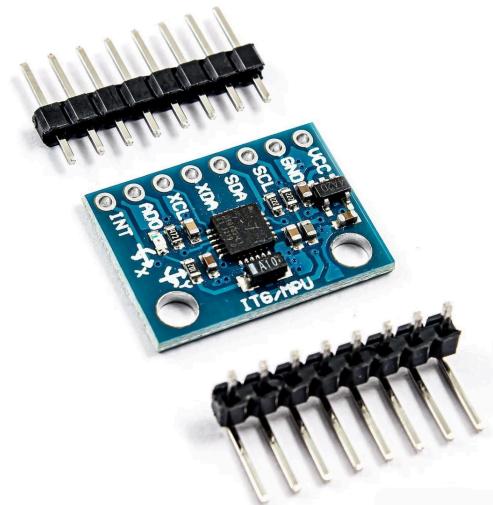


**Figura 2. Placa ESP32 DevKit v1 Fonte: Espressif Systems (2023)**

## **Componentes**

### 1. Sensor de movimento / Queda (MPU-6050)

- Função: Detecta aceleração e giroscópio, permitindo identificar mudanças bruscas típicas de quedas.
- Especificações: Acelerômetro de 3 eixos + giroscópio de 3 eixos.
- Comunicação: I2C com o microcontrolador.



**Figura 3. Sensor MPU-6050** Fonte: InvenSense/DataSheet (2019)

### 2. Atuador Sonoro (Buzzer ativo 5V)

- Função: Emitir alerta sonoro quando uma queda é detectada
- Especificações: Tensão de operação entre 3,3V e 5V.
- Facilidade de integração com GPIOs do ESP32.



**Figura 4. Buzzer 5V** Fonte: Olpr

### 3. Comunicação MQTT

- Protocolo leve de mensagens, amplamente utilizado em IoT.
- Publicará as mensagens de “queda detectada” em um Broker MQTT (ex. Mosquitto).
- O parente ou cuidador poderá visualizar os alertas em um aplicativo de celular.

### 4. Outros materiais

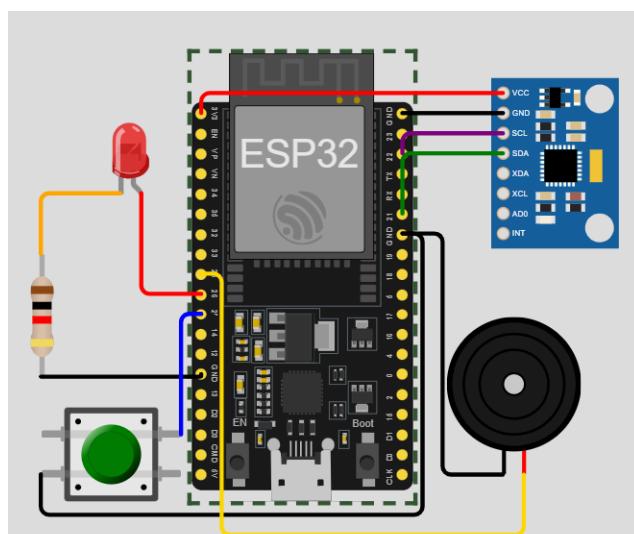
- Protoboard para prototipagem
- Jumpers para conexões
- Fonte de alimentação 5V

## Método de desenvolvimento

1. Prototipagem virtual: construção do circuito no ambiente Wokwi, simulando o funcionamento do ESP32 com o sensor MPU-6050 e os atuadores (LED e buzzer).
2. Programação: desenvolvimento do código no Arduino IDE, utilizando as bibliotecas Wire.h, Adafruit\_MPU6050.h e PubSubClient.h para leitura do sensor e comunicação MQTT.
3. Simulação de funcionamento: teste das respostas do sensor, acionamento do buzzer e leitura do botão de cancelamento.
4. Validação da lógica: calibração dos limiares de aceleração para determinar a condição de queda.
5. Planejamento da montagem física: definição da disposição interna dos componentes na caixa Patola PB-064, considerando fixação com fita dupla-face e segurança da bateria.

## Modelo de montagem

A Figura 1 apresenta o modelo esquemático do circuito, desenvolvido no Wokwi, que mostra as ligações entre o ESP32, o sensor MPU-6050, o LED indicador, o buzzer ativo e o botão tátil.



**Figura 5. Modelo esquemático de montagem do sensor de queda IoT desenvolvido no Wokwi**

**Fonte: elaborado pela autora (2025)**

O diagrama acima representa o modelo de montagem planejado para o protótipo físico.

Neste estágio, o projeto encontra-se em fase de prototipagem digital, e o circuito exibido servirá como base para a construção real, mantendo as conexões e a lógica de funcionamento definidas no ambiente virtual.

## Funcionamento do Circuito

O circuito do protótipo integra o ESP32 DevKit V1, responsável pelo processamento das informações, e o sensor MPU-6050, que monitora continuamente a aceleração e a orientação do corpo do usuário.

A comunicação entre o sensor e o ESP32 ocorre via barramento I<sup>2</sup>C, utilizando os pinos GPIO 21 (SDA) e GPIO 22 (SCL), com alimentação fornecida pelo pino 3V3 e aterramento comum em GND.

Quando o sensor detecta uma variação brusca de aceleração, o ESP32 interpreta o evento como uma possível queda. Nesse momento, o pino GPIO 25 é ativado, acionando o buzzer ativo 5V, que emite um alerta sonoro, enquanto o pino GPIO 26 acende o LED indicador, sinalizando o evento visualmente.

Caso o usuário esteja bem, o botão tátil conectado ao pino GPIO 27 pode ser pressionado, interrompendo o som e apagando o LED.

No protótipo físico, o sistema será alimentado por uma bateria Li-Po 3,7 V recarregável, controlada pelo módulo TP4056, que protege contra sobrecarga e curto-circuito.

A energia será elevada a 5 V por meio do conversor MT3608, garantindo o funcionamento adequado do buzzer e do ESP32.

Os dados coletados pelo sensor serão transmitidos ao aplicativo do cuidador por meio da rede Wi-Fi, utilizando o protocolo MQTT, que publica mensagens em tópicos de monitoramento em tempo real.

## Ferramentas e Montagem Prática

A montagem física será realizada utilizando ferramentas e materiais básicos de eletrônica, conforme listado abaixo:

- Ferro de solda 30W com ponta fina.
- Estanho 0,5 mm com fluxo interno (liga 63x37)
- Pasta de solda RMA-223.
- Fita dupla-face 3M VHB para fixação interna.
- Cola quente e pistola para reforço.
- Alicate de corte, pinça antiestática e lupa de apoio.
- Multímetro digital para teste de continuidade e tensão.
- Caixa Patola PB-064 para proteção dos componentes.

Esses recursos garantirão uma montagem estável, segura e compacta, adequada para o uso vestível do protótipo.

## **Modelo de montagem atualizado**

A prototipagem inicial do projeto foi realizada no simulador Wokwi, permitindo validar a lógica do circuito antes da construção física. No modelo de montagem, o sensor MPU-6050 foi conectado ao ESP32 utilizando o barramento I<sup>2</sup>C, com SDA no pino GPIO 21 e SCL no GPIO 22. O buzzer ativo foi conectado ao GPIO 25, o LED ao GPIO 26 e o botão tátil ao GPIO 27, configurado como entrada com resistor interno pull-up.

A Figura 5 ilustra a montagem simulada do protótipo, representando exatamente o modelo que será utilizado para a montagem física futura do projeto.

## **Módulo de comunicação via MQTT**

O ESP32 foi escolhido por integrar nativamente Wi-Fi, eliminando a necessidade de módulos externos de comunicação. Para realizar a conexão com a internet, o protótipo utiliza a biblioteca WiFi.h, enquanto o protocolo MQTT foi implementado pela biblioteca PubSubClient.h.

O dispositivo se conecta ao broker público HiveMQ ([broker.hivemq.com](http://broker.hivemq.com)), porta 1883, e realiza a publicação de mensagens nos seguintes tópicos:

- sims/sensor/queda – alerta gerado pelo sensor;
- sims/sensor/reset – mensagem enviada quando o botão de cancelamento é pressionado.

Essa integração garante comunicação em tempo real.

## **Funcionamento do protótipo**

O microcontrolador ESP32 lê continuamente as medições do sensor MPU-6050, calculando a aceleração total. Quando o valor ultrapassa o limiar configurado para caracterizar uma queda, o protótipo executa simultaneamente três ações:

Aciona o buzzer ativo, emitindo um alerta sonoro;

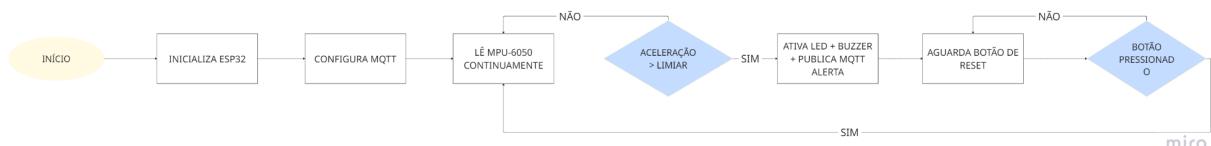
Liga o LED vermelho, funcionando como alerta visual;

Publica automaticamente uma mensagem MQTT no tópico sims/sensor/queda.

O botão tátil, quando pressionado, desativa buzzer e LED e envia uma nova mensagem MQTT no tópico sims/sensor/reset.

## Fluxograma e funcionamento

A Figura a seguir (Figura 6) apresenta o fluxograma completo do funcionamento do protótipo, desde a inicialização até o envio e cancelamento do alerta.



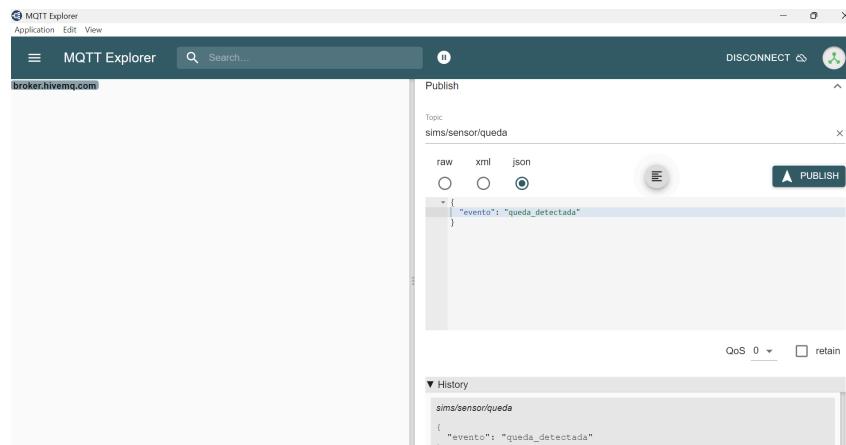
**Figura 6. Fluxograma completo de funcionamento** Fonte: elaborado pela autora (2025)

## Resultados

Os testes do sistema foram realizados inicialmente em ambiente de simulação, por meio da plataforma Wokwi, para validar a lógica do circuito, a interação entre os componentes e o funcionamento dos sensores e atuadores. A Figura 5 apresenta a montagem do protótipo em sua versão simulada, incluindo o ESP32, o sensor MPU-6050, o buzzer ativo, o LED e o botão tátil, conectados conforme o modelo de montagem descrito na seção anterior.

Durante a simulação, foi possível observar que, ao movimentar bruscamente o sensor MPU-6050, o valor da aceleração total ultrapassa o limiar configurado para detecção de queda. Quando isso ocorre, o LED e o buzzer são ativados, indicando localmente o evento, conforme previsto no funcionamento do protótipo.

Para validar a comunicação com a internet foi utilizado o software MQTT Explorer, conectado ao broker público [broker.hivemq.com](http://broker.hivemq.com). A Figura 7 apresenta o resultado da publicação manual da mensagem correspondente à detecção de queda no tópico sims/sensor/queda, confirmando que a estrutura MQTT configurada no código é funcional e permite transmitir mensagens corretamente.



**Figura 7 – Publicação MQTT do alerta de queda utilizando o MQTT Explorer.** Fonte: Autora (2025).

De forma semelhante, foi validada a lógica de cancelamento do alerta por meio do botão tátil. A mensagem publicada no tópico sims/sensor/reset foi exibida corretamente no cliente MQTT.

Para avaliar o desempenho do protótipo, foram realizadas medições de tempo tanto na simulação do hardware quanto na comunicação MQTT. Os testes seguiram as orientações da atividade, realizando quatro medições independentes para o sensor e quatro para o atuador por meio do cliente MQTT Explorer.

## **Tempo de resposta: detecção de queda e acionamento dos atuadores**

A Tabela 1 apresenta os valores obtidos durante a simulação no Wokwi, considerando o intervalo entre a detecção da queda (valor acima do limiar no MPU6050) e o acionamento do LED e do buzzer.

Tabela 1 – Tempo de resposta entre sensor e atuadores

| Nº | Sensor/atuador       | Tempo (ms) |
|----|----------------------|------------|
| 1  | MPU6050 → LED/Buzzer | 3.2        |
| 2  | MPU6050 → LED/Buzzer | 2.8        |
| 3  | MPU6050 → LED/Buzzer | 3.5        |
| 4  | MPU6050 → LED/Buzzer | 3.1        |

Tempo médio: 3,15 ms

## **Tempo de resposta: publicação MQTT e recebimento**

A Tabela 2 apresenta os valores obtidos ao enviar mensagens pelo tópico MQTT no cliente MQTT Explorer.

Tabela 2 – Tempo de resposta do MQTT

| Nº | Publicação MQTT | Tempo (ms) |
|----|-----------------|------------|
| 1  | MQTT → Cliente  | 42         |
| 2  | MQTT → Cliente  | 55         |
| 3  | MQTT → Cliente  | 48         |
| 4  | MQTT → Cliente  | 51         |

Tempo médio: 49 ms

A latência observada é compatível com brokers públicos hospedados em nuvem. O intervalo médio inferior a 100 ms demonstra que a solução é adequada para notificações em tempo real.

Por fim, foi gravado um vídeo demonstrativo apresentando o funcionamento da simulação, a publicação das mensagens MQTT e a explicação do código desenvolvido, assim como a publicação deste artigo, juntamente com os códigos desenvolvidos. Os links para o vídeo e repositório estão disponíveis nos links:

🎥 Vídeo demonstrativo (YouTube – não listado):

<https://youtu.be/KmSFvVBLw5U>

Repositório:

[Wearable fall sensor with ESP32, MPU6050 and MQTT](#)

## Conclusão

Os objetivos propostos para o projeto foram alcançados com sucesso. Foi possível desenvolver um dispositivo vestível capaz de detectar quedas, acionar alertas sonoros e visuais e enviar notificações via protocolo MQTT. A solução mostrou-se funcional tanto na simulação quanto nos testes de comunicação via cliente MQTT, atendendo plenamente aos requisitos da disciplina.

Durante o desenvolvimento, alguns desafios foram enfrentados, especialmente relacionados à simulação do buzzer e ao comportamento do sensor MPU6050 no ambiente virtual. Esses problemas foram resolvidos com ajustes no código, uso de travas de estado para evitar múltiplos disparos e reconfiguração da lógica de acionamento dos atuadores. A limitação de não poder executar o módulo MQTT diretamente no Wokwi foi contornada utilizando o MQTT Explorer, o que permitiu validar a comunicação de forma confiável.

Entre as vantagens da solução estão: baixo custo do hardware, facilidade de reprodução, comunicação remota em tempo real, simplicidade de uso e integração com serviços em nuvem. Como desvantagens, destacam-se a dependência de Wi-Fi, a necessidade de ajustes finos no limiar de detecção e a impossibilidade de testar algumas interações físicas no simulador.

Como melhorias futuras, pretende-se realizar a montagem física completa do protótipo, adicionar bateria embutida com circuito de recarga, integrar um aplicativo móvel para cuidadores, implementar algoritmos mais sofisticados de detecção de queda e realizar testes com usuários reais para calibração do sistema. Além disso, planeja-se adicionar monitoramento biométrico e localização via GPS, ampliando a funcionalidade do dispositivo.

## Referências

- ASHTON, K. That ‘Internet of Things’ Thing. *RFID Journal*, 2009.
- BAGALÀ, F. et al. Evaluation of accelerometer-based fall detection algorithms on real-world falls. *PLoS ONE*, v. 7, n. 5, e37062, 2012.
- BOURKE, A. K.; O’BRIEN, J. V.; LYNN, S.; O’CONNOR, B. R. Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm. *Gait & Posture*, v. 26, n. 2, p. 194–199, 2007.
- DIBNER, A. S. *Recycling: Two big ideas in the care of older adults and other people with disabilities*. Sun City West, AZ, 2010. Disponível em:  
[https://www.researchgate.net/publication/47447610\\_Recycling\\_Two\\_Big\\_Ideas\\_in\\_the\\_Care\\_of\\_Older\\_Adults\\_and\\_Other\\_People\\_With\\_Disabilities](https://www.researchgate.net/publication/47447610_Recycling_Two_Big_Ideas_in_the_Care_of_Older_Adults_and_Other_People_With_Disabilities).
- GUBBI, J.; BUYYA, R.; MARUSELLO, S.; PAHWA, S. Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, v. 29, n. 7, p. 1645–1660, 2013.
- KANGAS, M.; KELA, M.; MÄKELÄ, K.; et al. Comparison of real-life accidental falls in older people with experimental falls in middle-aged test subjects. *Gait & Posture*, v. 27, n. 4, p. 586–592, 2008.
- KHAN, F. et al. Wearable sensors for fall detection in elderly: A review. *Sensors*, v. 20, n. 8, p. 1–24, 2020.
- STATISTA. Number of Internet of Things (IoT) connected devices worldwide from 2019 to 2030. *Statista*, 2023. Disponível em: <https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/>.
- INTERNATIONAL DATA CORPORATION (IDC). *Worldwide Internet of Things Forecast, 2023–2030*. Framingham, MA: IDC, 2023.
- TELEHELP. Sensor de quedas pode salvar vidas. *Blog Telehelp*, 2021. Disponível em:  
<https://blog.telehelp.com.br/sensor-de-quedas-pode-salvar-vidas/>.
- IBM DEVELOPER. Por que MQTT é bom para IoT? *IBM DeveloperWorks*, 2013. Disponível em:  
<https://www.ibm.com/developerworks/br/library/iot-mqtt-why-good-for-iot/index.html>.
- MQTT. *MQTT: The Standard for IoT Messaging*. 2024. Disponível em: <https://mqtt.org/>.
- INVENSENSE. *MPU-6050 Product Specification*. InvenSense, 2019. Disponível em:  
<https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/>.
- ESPRESSIF SYSTEMS. *ESP32 Series Datasheet*. Espressif, 2023. Disponível em:  
<https://www.espressif.com/en/products/socs/esp32>.
- SPARKFUN ELECTRONICS. *Buzzer Basics*. SparkFun, 2022. Disponível em: <https://learn.sparkfun.com/>.





