



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA BIOMÉDICA
CURSO DE GRADUAÇÃO EM ENGENHARIA BIOMÉDICA

JOÃO GABRIEL DE SOUZA VERAS

**DESENVOLVIMENTO DE UM SISTEMA PORTÁTIL PARA *BIOFEEDBACK*
RESPIRATÓRIO EM TEMPO REAL COM SENSOR DE PRESSÃO E INTERFACE
MOBILE VIA COMUNICAÇÃO *BLUETOOTH LOW ENERGY***

Recife

2025

JOÃO GABRIEL DE SOUZA VERAS

**DESENVOLVIMENTO DE UM SISTEMA PORTÁTIL PARA *BIOFEEDBACK*
RESPIRATÓRIO EM TEMPO REAL COM SENSOR DE PRESSÃO E INTERFACE
MOBILE VIA COMUNICAÇÃO *BLUETOOTH LOW ENERGY***

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Biomédica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Biomédica.

Orientador (a): Prof. Dr. Emery Cleiton Cabral Correia Lins

Recife

2025

Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE

Veras, João Gabriel de Souza.

Desenvolvimento de um sistema portátil para biofeedback respiratório em tempo real com sensor de pressão e interface mobile via comunicação bluetooth low energy / João Gabriel de Souza Veras. - Recife, 2025.

46p. : il., tab.

Orientador(a): Emery Cleiton Cabral Correia Lins

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia Biomédica - Bacharelado, 2025.

Inclui referências, apêndices.

1. Biofeedback. 2. Frequência Respiratória. 3. Sensor de Pressão. 4. Saúde Preventiva. 5. Internet das Coisas. I. Lins, Emery Cleiton Cabral Correia. (Orientação). II. Título.

620 CDD (22.ed.)

JOÃO GABRIEL DE SOUZA VERAS

**DESENVOLVIMENTO DE UM SISTEMA PORTÁTIL PARA *BIOFEEDBACK*
RESPIRATÓRIO EM TEMPO REAL COM SENSOR DE PRESSÃO E INTERFACE
MOBILE VIA COMUNICAÇÃO *BLUETOOTH LOW ENERGY***

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Biomédica da Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, como requisito parcial para a obtenção do grau de Bacharel em Engenharia Biomédica.

Aprovado em: 13/08/2025.

BANCA EXAMINADORA

Prof. Dr. Emery Cleiton Cabral Correia Lins (Orientador)
Universidade Federal de Pernambuco

Profa. Dra. Patrícia Silva Lessa (Examinadora Externa)
Universidade Federal de Pernambuco

RESUMO

A frequência respiratória (FR) é um parâmetro essencial para a avaliação de condições fisiológicas, sendo um importante preditor de distúrbios como hipoxemia, hipercapnia e falhas respiratórias. Este trabalho apresenta o desenvolvimento de um sistema de *biofeedback* respiratório não invasivo, utilizando um sensor de pressão relativa acoplado a um microcontrolador ESP32 com comunicação *Bluetooth Low Energy* (BLE), integrado a um aplicativo *mobile*. O sistema é capaz de captar e processar variações na pressão do fluxo respiratório, aplicando técnicas de filtragem, compensação de *offset* e detecção de picos e vales para determinar os tempos de inspiração e expiração. Os dados são transmitidos e plotados em tempo real, fornecendo retorno visual contínuo ao usuário. A proposta visa contribuir para o monitoramento remoto da função respiratória, com aplicações em contextos de saúde preventiva.

Palavras-chave: Biofeedback; Frequência Respiratória; Sensor de Pressão; Saúde Preventiva; Internet das Coisas.

ABSTRACT

Respiratory rate (RR) is a vital sign crucial for assessing physiological status and serves as a key predictor of adverse events, including hypoxemia, hypercapnia, and respiratory failure. This paper details the development of a non-invasive respiratory biofeedback system, which employs a pressure sensor interfaced with an ESP32 microcontroller featuring Bluetooth Low Energy (BLE) connectivity, integrated with a mobile application. The system is engineered to capture and process pressure variations within the respiratory airflow by employing signal processing techniques such as filtering, offset compensation, and peak-and-trough detection to ascertain inspiratory and expiratory times. The processed data is transmitted and plotted in real-time, offering the user continuous visual feedback. This work aims to contribute to the remote monitoring of respiratory function, showing potential for applications in preventive healthcare settings.

Keywords: Biofeedback ; Respiratory Rate; Pressure Sensor; Preventive Healthcare; Internet of Things.

LISTA DE ILUSTRAÇÕES

Figura 1 –	Diagrama da comunicação do sistema	18
Figura 2 –	Componentes de desenvolvimento da unidade sensora.....	20
Figura 3 –	Projeto MA2R no ambiente de desenvolvimento do MIT App Inventor 2	21
Figura 4 –	Telas do aplicativo MA2R na sequência de uso comum.....	22
Figura 5 –	Validação experimental do protótipo MA2R.....	24
Figura 6 –	Gráfico detalhado da curva respiratória adquirida pela unidade sensora.....	26

LISTA DE TABELAS

Tabela 1 –	Parâmetros simulados pelo ventilador mecânico SV300.....	22
Tabela 2 –	Dados do relatório de análise do sistema simulado	26
Tabela 3 –	Resultados dos Testes com Voluntários em Repouso e Caminhada.....	27

SUMÁRIO

1	INTRODUÇÃO	10
1.1	OBJETIVOS	11
1.1.1	Objetivo Geral	11
1.1.2	Objetivos Específicos	12
1.2	JUSTIFICATIVAS	12
2	REFERENCIAL TEÓRICO	13
3	MATERIAIS E MÉTODOS.....	18
3.1	MODELAGEM DO SISTEMA	18
3.2	UNIDADE SENSORA.....	18
3.3	DESENVOLVIMENTO DO APLICATIVO MOBILE	20
3.4	MÉTODO DE SIMULAÇÃO	22
4	RESULTADOS	25
4.1	DADOS DE CALIBRAÇÃO	25
4.2	SIMULAÇÃO DO SISTEMA COM VOLUNTÁRIOS	27
5	CONCLUSÃO	28
	REFERÊNCIAS	29
	APÊNDICE A	31
	APÊNDICE B	39

1 INTRODUÇÃO

A respiração é um processo fisiológico essencial para a manutenção da vida, resultante da ação coordenada dos músculos respiratórios e do centro de controle localizado no cérebro. Sua principal função é viabilizar a troca gasosa entre o ar inspirado e o sangue circulante, garantindo o transporte de oxigênio — indispensável ao metabolismo celular — e a remoção do dióxido de carbono, que participa ativamente da regulação do pH sanguíneo. Alterações no ritmo respiratório podem ocorrer como resposta fisiológica, por exemplo, durante o esforço físico, ou indicar condições clínicas relevantes, como insuficiência respiratória e doenças pulmonares obstrutivas crônicas (DPOC) (BRINKMAN; TORO; SHARMA, 2025; LIU *et al.*, 2019).

Para preservar o equilíbrio entre oferta e demanda de gases, o sistema respiratório dispõe de mecanismos adaptativos que ajustam o padrão ventilatório conforme as variações na composição dos gases sanguíneos, atendendo às necessidades metabólicas do organismo. Nesse contexto, destaca-se a importância da frequência respiratória (FR). Por ser um parâmetro vital, quando alterada, pode sinalizar a ocorrência de distúrbios clínicos graves como em casos de hipoxemia (baixa oxigenação do sangue arterial) e hipercapnia (acúmulo de CO₂ no sangue arterial) que comprometem o equilíbrio respiratório e podem evoluir para quadros críticos (BRINKMAN; TORO; SHARMA, 2025; LIU *et al.*, 2019).

Além disso, a FR é considerada um preditor precoce de deterioração clínica. Evidências científicas apontam sua associação com desfechos adversos, como paradas cardiorrespiratórias, falhas no desmame ventilatório, pneumonia e readmissões em unidades de terapia intensiva (UTI). Dessa forma, o monitoramento contínuo e em tempo real do padrão respiratório se destaca como uma estratégia fundamental na prevenção de complicações através da identificação precoce de alterações fisiológicas, possibilitando intervenções clínicas mais eficazes e oportunas (LIU *et al.*, 2019; HUSSAIN *et al.*, 2023).

Concomitantemente, com o avanço das tecnologias vestíveis (*wearables*), o acompanhamento dos sinais vitais tem se tornado mais acessível, preciso e constante. Esses dispositivos, aliados a biossensores, permitem a captação de variáveis fisiológicas, como temperatura corporal, frequência cardíaca, movimento e

fluxo respiratório, transformando-as em sinais elétricos ou ópticos que podem ser analisados por softwares especializados. A depender do tipo de sensor, os *wearables* se classificam em diferentes categorias, incluindo sensores baseados em princípios optoeletrônicos, deformação, temperatura, reações eletroquímicas e pressão (KAZANSKIY; KHONINA; BUTT, 2024).

É nesse cenário que se insere a Internet das Coisas (IoT), que potencializa a integração entre sensores e plataformas digitais, promovendo a coleta, transmissão e análise contínua de dados fisiológicos em tempo real. Essa conectividade permite o envio de informações para dispositivos móveis, otimizando o acompanhamento remoto de pacientes e favorecendo intervenções precoces, mesmo fora do ambiente hospitalar (SHAMSABADI *et al.*, 2022).

A partir dessas tecnologias, destaca-se o conceito de *biofeedback*, técnica que visa promover a autorregulação por meio do monitoramento e devolução de informações fisiológicas em tempo real ao paciente. Essa estratégia é amplamente utilizada em programas de reabilitação física, controle do estresse, manejo de doenças crônicas e no aprimoramento do desempenho cardiorrespiratório, inclusive em contextos esportivos (MALIK; DUA, 2025).

Dessa forma, para o monitoramento de padrões respiratórios, vale salientar a aplicação de sensores piezoresistivos. Estes dispositivos são notáveis por sua alta sensibilidade na conversão da tensão mecânica em um sinal elétrico. O princípio de funcionamento reside na variação da resistência elétrica do material conforme ele é deformado. Por ser um processo reversível, o sensor retorna ao seu estado inicial após cada ciclo respiratório, o que viabiliza um acompanhamento contínuo e preciso da respiração (DE FAZIO *et al.*, 2021).

Diante do exposto, observa-se a relevância do desenvolvimento de tecnologias acessíveis e não invasivas que possibilitem o monitoramento contínuo da frequência respiratória de forma prática, precisa e em tempo real.

1.1 OBJETIVOS

1.1.1 Objetivo Geral

Desenvolver e validar um sistema de *biofeedback* respiratório composto por um sensor de pressão piezoresistivo integrado a um microcontrolador ESP32 para processamento de dados e comunicação BLE com um aplicativo *mobile*.

1.1.2 Objetivos Específicos

Fornecer e monitorar, em tempo real, o *feedback* da pressão aplicada do fluxo respiratório do usuário. O sistema tem como objetivo auxiliar no acompanhamento e na análise da frequência respiratória e da relação inspiração/expiração (I:E), possibilitando a geração de relatórios qualitativos sobre o padrão respiratório.

1.2 JUSTIFICATIVAS

A respiração desempenha um papel fundamental na regulação fisiológica e emocional do organismo, sendo frequentemente afetada em condições clínicas como asma e DPOC. Estratégias de *biofeedback* respiratório têm sido utilizadas como ferramentas de monitoramento e avaliação de pacientes com anomalias respiratórias. No entanto, o acesso a essas tecnologias ainda é limitado por fatores como custo elevado e infraestrutura complexa.

Com os avanços na eletrônica embarcada e nas tecnologias móveis, torna-se possível desenvolver soluções acessíveis, portáteis e eficazes. O uso do ESP32 como microcontrolador, aliado à comunicação *Bluetooth Low Energy* (BLE) e sensores de pressão, permite a criação de sistemas compactos, de baixo custo e com alta capacidade de processamento e conectividade.

Assim, este trabalho justifica-se pela necessidade de promover o acesso e acompanhamento do *biofeedback* respiratório por meio de um sistema integrado a um aplicativo *mobile*, com potencial de aplicação em atividades de rotina de baixa ou média intensidade para contextos de saúde preventiva, especialmente em populações que necessitam de intervenções não invasivas e personalizadas.

2 REFERENCIAL TEÓRICO

O controle da respiração é coordenado por estruturas localizadas na medula oblonga, onde um grupo de neurônios especializados atua na geração rítmica dos ciclos respiratórios. Esses neurônios se comunicam com outras regiões do sistema nervoso central (SNC), recebendo estímulos de diversas origens: (1) receptores pulmonares sensíveis à irritação (adaptação rápida) e ao estiramento (adaptação lenta); (2) quimiorreceptores centrais e periféricos, que reagem às variações na concentração de dióxido de carbono e ao pH sanguíneo; e (3) sinais sensoriais e emocionais, como dor, medo ou ansiedade, que também influenciam o ritmo respiratório. Esse sistema de regulação é altamente adaptativo, ajustando-se automaticamente diante das demandas fisiológicas, mesmo sem percepção consciente. Essas adaptações podem incluir a interrupção momentânea da respiração em situações como deglutição, espirros, bocejos ou suspiros. A integração dessas informações aferentes pelo centro respiratório resulta em ajustes nos padrões neurais que regulam a frequência, intensidade e duração das fases do ciclo respiratório (GALLARDO *et al.*, 2024).

Considerando essa complexidade no controle neural da respiração, a monitorização da frequência respiratória (FR) torna-se uma ferramenta clínica valiosa. Quando a FR se encontra fora dos limites fisiológicos normais — abaixo de 6 ou acima de 24 respirações por minuto — pode ser indicativo precoce de deterioração clínica. De fato, estudos apontam que alterações na FR são frequentemente mais sensíveis na predição de mortalidade do que outros sinais vitais, como a frequência cardíaca ou a pressão arterial. No entanto, a prática convencional de contagem visual da FR apresenta limitações significativas, incluindo a possibilidade de erros de observação e variações entre examinadores, o que compromete a confiabilidade da análise e pode levar a interpretações equivocadas do estado clínico do paciente. Nesse contexto, o uso de sensores capazes de realizar medições contínuas e em tempo real surge como uma alternativa promissora, oferecendo suporte ao diagnóstico e acompanhamento de condições respiratórias como asma, DPOC e apneia do sono (HUSSAIN *et al.*, 2023).

A importância do monitoramento contínuo da FR também é reforçada por dados da Organização Mundial da Saúde (OMS), que destacam as doenças respiratórias como uma das principais causas de morte no mundo, ao lado de doenças cardíacas, diabetes e neoplasias. Entre as enfermidades de maior impacto estão a DPOC, as infecções respiratórias inferiores e os cânceres de traquéia e pulmão. Diante desse cenário, torna-se cada vez mais essencial o acompanhamento contínuo dos indicadores de saúde dos pacientes, como estratégia para prevenção, diagnóstico precoce e intervenção oportuna com o monitoramento contínuo (KIRAN KOLLURI; ANANIAH DURAI, 2024).

Entre as condições respiratórias mais prevalentes destaca-se a DPOC. De acordo com a *Global Initiative for Chronic Obstructive Lung Disease* (GOLD), a DPOC é caracterizada por sintomas respiratórios persistentes e limitação ao fluxo aéreo, decorrentes de alterações nas vias respiratórias e nos alvéolos, geralmente associada à exposição prolongada a agentes irritantes (MOTA *et al.*, 2023).

No Brasil, dados da Secretaria de Vigilância em Saúde (SVS), alinhados à metodologia do estudo *Global Burden of Disease* (GBD), indicam que a DPOC ocupa a quinta posição entre as principais causas de morte em todas as faixas etárias. Historicamente, também figura entre os cinco principais motivos de internação no SUS em pessoas acima dos 40 anos, com aproximadamente 200 mil hospitalizações anuais e um impacto financeiro estimado em 72 milhões de reais por ano. Em estágios mais avançados, a doença compromete severamente a qualidade de vida, provocando crises respiratórias mais graves e recorrentes, além de limitação funcional devido à insuficiência respiratória crônica, fadiga intensa e dificuldade para realizar esforços físicos. (MOTA *et al.*, 2023).

A asma é outra condição respiratória de grande relevância, afetando cerca de 334 milhões de pessoas no mundo. Essa doença inflamatória crônica das vias aéreas se caracteriza pela obstrução reversível do fluxo aéreo, provocando sintomas como dificuldade para respirar (dispneia), e, ao exame físico, sejam percebidos sons respiratórios como sibilos. A alta prevalência da asma entre crianças e adolescentes e o impacto socioeconômico de suas crises – com aumento do absenteísmo escolar e laboral – evidenciam a importância de estratégias eficazes de monitoramento e

controle. Estima-se que aproximadamente 250 mil mortes por ano estejam associadas à asma em escala global (MARQUES et al., 2022).

Nesse cenário, os dispositivos vestíveis têm se destacado como ferramentas promissoras para o acompanhamento contínuo da saúde respiratória. Diferentemente dos equipamentos tradicionais, esses dispositivos funcionam com baterias de baixo consumo, oferecem portabilidade e permitem o monitoramento em tempo real de parâmetros fisiológicos em diversos ambientes. Esses sistemas voltados à saúde digital são capazes de processar, em tempo real, informações biomédicas complexas e de alta resolução. Além disso, possibilitam a personalização do acompanhamento, adaptando-se aos padrões comportamentais e clínicos de cada usuário (MOON; LEE, 2023).

Essa evolução tecnológica está diretamente associada ao avanço da Internet das Coisas, que conecta dispositivos físicos à internet, possibilitando a coleta, o compartilhamento e a análise remota de dados. No contexto da saúde, a IoT permite que os dispositivos vestíveis transmitam informações fisiológicas em tempo real para sistemas em nuvem, plataformas médicas ou diretamente para profissionais de saúde. Essa conectividade amplia significativamente o potencial de intervenção precoce e eficiente, monitoramento contínuo e tomada de decisões clínicas baseadas em dados, mesmo à distância. (MUNIR *et al.*, 2022; HOSSAIN *et al.*, 2023).

Os dispositivos vestíveis têm ganhado cada vez mais espaço no cenário tecnológico contemporâneo, tornando-se parte essencial da rotina diária de milhões de pessoas. O crescimento desse mercado tem sido expressivo: entre 2016 e 2022, o número de aparelhos em uso saltou de 325 milhões para aproximadamente 1,1 bilhão, e a tendência é que esse número ultrapasse os 2 bilhões em um futuro próximo. Essas tecnologias, amplamente utilizadas para monitoramento em tempo real da saúde e do condicionamento físico, tornaram-se ferramentas valiosas no estilo de vida atual. Pequenos e sofisticados, os dispositivos integram múltiplos microssores capazes de registrar atividades como reconhecimento de movimentos e gestos, postura corporal, detecção de subidas e quedas, esforço articular, contagem de passos, análise da marcha e posicionamento angular. Além disso, com os avanços na chamada Rede de Área Corporal (BAN), tem sido possível realizar diversos tipos de diagnósticos médicos à distância, por meio de sensores

vestíveis que monitoram sinais vitais como frequência cardíaca, oxigenação do sangue, pressão arterial, eletrocardiograma e balistocardiograma (KIRAN KOLLURI; ANANIAH DURAI, 2024).

No contexto da saúde respiratória, os wearables representam um avanço significativo por viabilizarem o monitoramento contínuo de sinais vitais, como a frequência respiratória, contribuindo para o autocuidado e o acompanhamento clínico. Por serem de uso individual, também reduzem o risco de contaminação cruzada – um desafio relevante no uso compartilhado de equipamentos médicos convencionais (ZHOU; LUKOWICZ, 2020).

Uma das aplicações clínicas mais promissoras desses dispositivos é no *biofeedback*. Ele pode ser classificado em biomecânico – voltado ao controle motor e postura – e fisiológico – englobam diversos métodos de monitoramento de funções orgânicas. Para tal, o método envolve o uso de dispositivos não invasivos que monitoram determinado parâmetro fisiológico, cuja leitura é transformada em sinais visuais, sonoros ou táteis. A partir dessa retroalimentação externa, o paciente aprende a ajustar deliberadamente suas respostas corporais. Por oferecer uma via complementar de percepção além da sensibilidade natural do corpo, o *biofeedback* também é denominado *feedback* aumentado ou extrínseco (MALIK; DUA, 2025).

A eficácia dessa abordagem, contudo, depende diretamente da precisão e confiabilidade dos sensores utilizados na coleta dos sinais fisiológicos. Entre os diversos tipos empregados, os sensores de pressão piezoresistivos se destacam por sua sensibilidade e compatibilidade com sistemas vestíveis. Seu princípio de funcionamento baseia-se na conversão de uma deformação mecânica – provocada pela pressão exercida sobre um diafragma flexível – em um sinal elétrico mensurável. Para isso, utilizam um elemento piezoresistivo integrado a uma ponte de *Wheatstone*, o que permite amplificar as variações de resistência elétrica geradas pela deformação do piezoresistor (R_1). Essa modificação na resistência é proporcional à força aplicada, possibilitando a geração de um sinal elétrico correspondente e de alta resolução (BAE et al., 2004).

Para isolar o efeito da pressão de variáveis ambientais, como a temperatura, e anular a tensão residual, um resistor de referência (R_4) idêntico a R_1 é colocado em uma área rígida do substrato, onde não sofre deformação. Essa configuração permite

cancelar as variações de resistência que não são causadas pela pressão, garantindo maior precisão. O circuito é complementado por um resistor variável (R2), utilizado para a calibração, e um resistor fixo (R3). A função de R2 é ajustar a ponte para que, na ausência de pressão, o circuito esteja perfeitamente balanceado (BAE *et al.*, 2004).

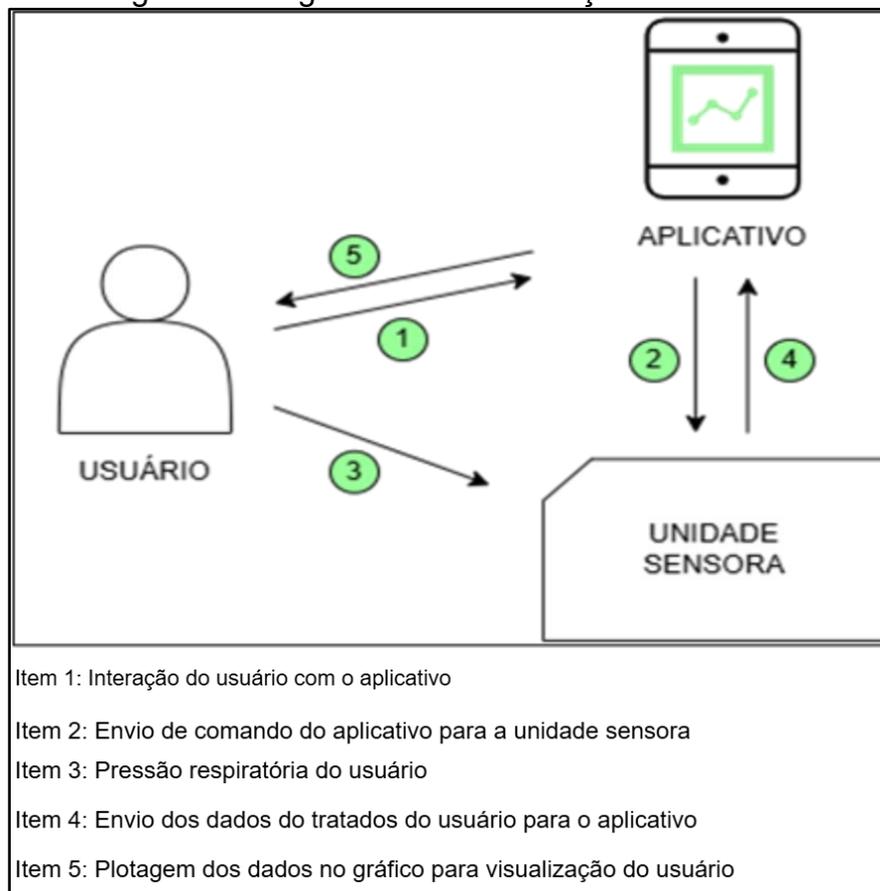
Para o sensor iniciar a aquisição de dados, uma fonte de tensão constante alimenta a ponte. Quando nenhuma pressão é aplicada, a ponte permanece em equilíbrio, resultando em uma diferença de potencial nula e, conseqüentemente, uma tensão de saída igual a zero. Ao aplicar uma pressão, o diafragma deforma-se, alterando a resistência de R1 e desequilibrando o circuito. Esse desequilíbrio gera uma diferença de potencial mensurável entre os braços da ponte de *Wheatstone*, resultando em uma tensão de saída que é diretamente proporcional à pressão aplicada, constituindo assim o sinal de medição final do sensor (BAE *et al.*, 2004).

3 MATERIAIS E MÉTODOS

3.1 MODELAGEM DO SISTEMA

O diagrama abaixo (Figura 1) apresenta o modelo de comunicação do usuário com o sistema de *biofeedback* respiratório. Em primeiro momento, o usuário utiliza o aplicativo para enviar um comando para a unidade sensora que irá iniciar a aquisição dos dados da pressão respiratória do usuário, processar a informação recebida e enviar de volta ao aplicativo, onde será apresentada ao usuário a curva respiratória em um gráfico atualizado em tempo real.

Figura 1: Diagrama da comunicação do sistema

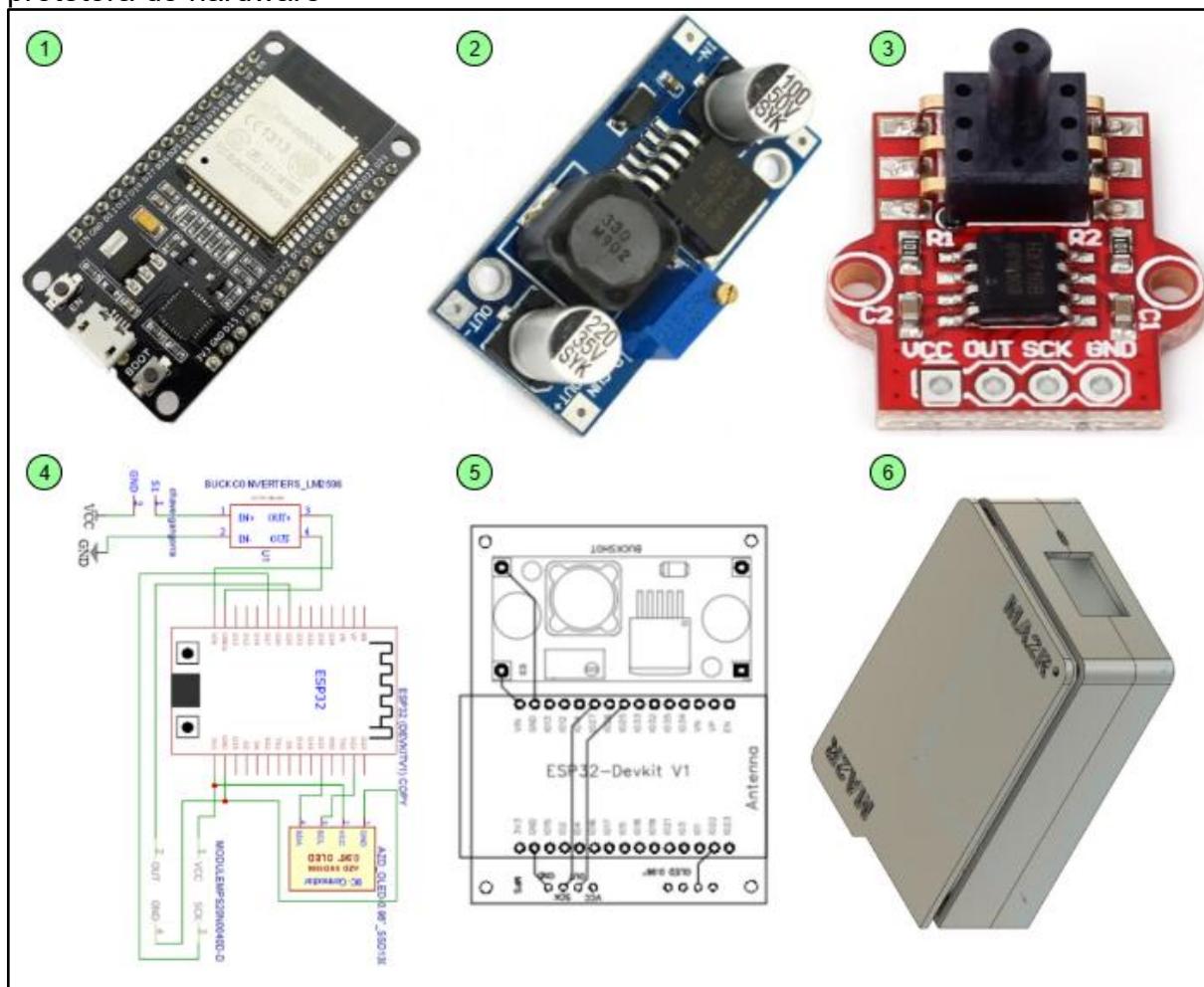


Fonte: Imagem do autor.

3.2 UNIDADE SENSORA

A unidade sensora (Figura 2), componente central do projeto MA2R, foi desenvolvida como um sistema embarcado autônomo, composto por um microcontrolador, um regulador de tensão e um sensor de pressão, todos integrados em uma placa de circuito impresso (PCI) e protegidos por uma capa de hardware. O microcontrolador selecionado é o ESP32-WROOM-32 (Figura 2.1), escolhido por sua arquitetura dual-core de 240 MHz e antena BLE 4.2 integrada. Sua capacidade de processamento paralelo é fundamental, permitindo que um núcleo gerencie a exigente pilha de comunicação BLE em tempo real, enquanto o outro se dedica ao processamento dos dados do sensor, garantindo assim que a comunicação não interfira na precisão das medições. Para a aquisição do sinal, o sistema utiliza o MPS20N0040D-S (Figura 2.3), um dispositivo de pressão diferencial piezoresistivo que converte a pressão do fluxo de ar em um sinal de tensão, o qual é digitalizado e amplificado com um ganho de 128 V/V por um conversor analógico-digital (ADC) HX711 de 24 bits, tornando possível capturar as sutis variações da respiração com alta fidelidade. A alimentação do circuito é fornecida por uma bateria de 6V (4 pilhas AAA), cuja tensão é estabilizada por um módulo conversor *DC-DC step-down* LM2596 (Figura 2.2). A inteligência do dispositivo reside no *firmware* do microcontrolador, onde está implementada a lógica de programação para múltiplas funções. Primeiramente, ele é responsável por criar o servidor *Bluetooth* que estabelece as regras de comunicação com o aplicativo móvel, além de calibrar o *offset* do sensor após o acionamento do hardware e gerenciar os comandos de iniciar e interromper a aquisição. Em segundo plano, o *firmware* processa o sinal de forma contínua, utilizando uma máquina de estados finitos para identificar os ciclos respiratórios completos. Com isso, torna-se responsável por contabilizar com precisão o intervalo de tempo da inspiração e expiração a cada ciclo e o tempo total da sessão para, finalmente, calcular e transmitir os parâmetros analíticos da frequência respiratória (FR) e a relação I:E. Para a materialização do hardware, o esquemático elétrico e o *layout* da PCI (Figura 2.4 e Figura 2.5) foram desenvolvidos na plataforma EasyEDA. Paralelamente, o desenho tridimensional da capa protetora (Figura 2.6) foi realizado no software Fusion 360, sendo posteriormente confeccionado através de impressão 3D, resultando em um dispositivo portátil, robusto e funcional.

Figura 2: Componentes de desenvolvimento da unidade sensora. (1) Microcontrolador; (2) Módulo Regulador de tensão; (3) Módulo Sensor de pressão; (4) Esquemático do circuito eletrônico; (5) Modelo 2D da PCI; (6) Modelo 3D da capa protetora do hardware

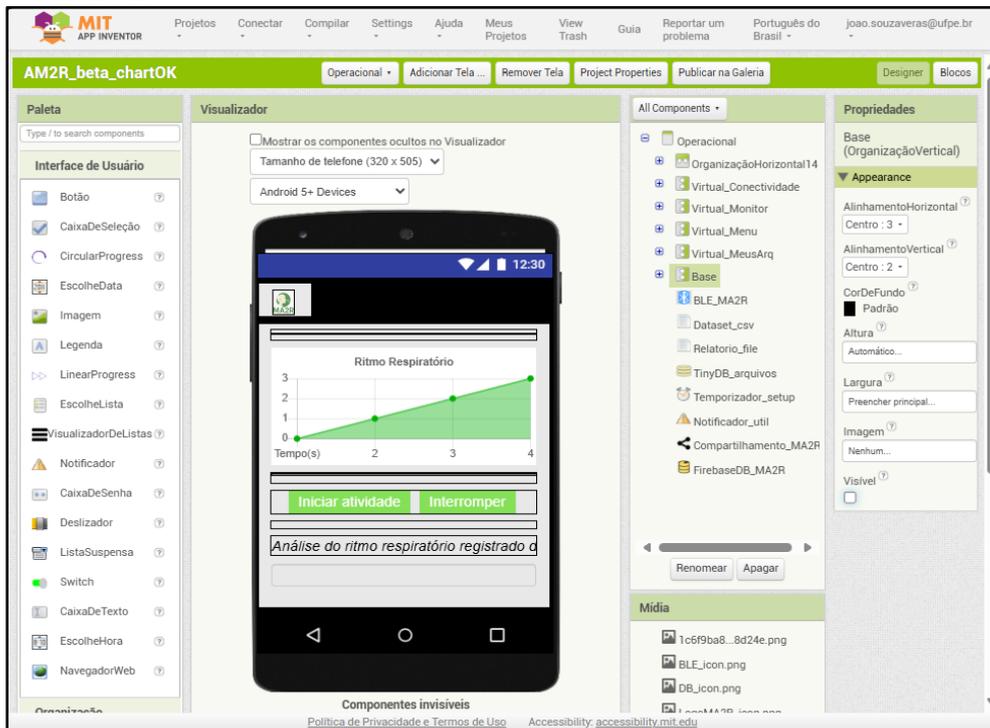


Fonte: Imagem do autor.

3.3 DESENVOLVIMENTO DO APLICATIVO MOBILE

O aplicativo foi desenvolvido em plataforma *online* gratuita do MIT App Inventor 2 (Figura 3) para receber, apresentar, armazenar e gerenciar os dados da unidade sensora. O app exibirá o *biofeedback* visual em tempo real através de um gráfico em plano cartesiano com base nos dados de tempo e tensão gerados pela variação do fluxo respiratório aplicado ao sensor de pressão.

Figura 3: Projeto MA2R no ambiente de desenvolvimento do MIT App Inventor 2



Fonte: Imagem do autor.

A arquitetura da interface do aplicativo foi desenvolvida de forma segmentada e funcional, dividida em quatro telas principais para garantir uma experiência de usuário intuitiva (Figura 4). O fluxo de uso comum começa na tela de abertura, que leva o usuário à tela de conectividade para parear o *smartphone* com o *hardware* via *Bluetooth*. Uma vez conectado, o usuário navega para a tela de monitoramento, o núcleo do projeto, onde inicia a aquisição de dados e visualiza a curva respiratória em um gráfico dinâmico. Ao interromper a coleta, a interface exibe um relatório final com métricas como Frequência Respiratória, classificação do ritmo e tempo total de aquisição. Para o gerenciamento do histórico, a tela de arquivos atua como um repositório local, permitindo ao usuário visualizar, compartilhar ou apagar as sessões gravadas, garantindo um controle completo sobre seus dados.

Figura 4: Telas do aplicativo MA2R na sequência de uso comum: (1) Tela Inicial; (2) Tela de conectividade; (3) Tela do Monitor; (4) Tela de gerenciamento de arquivos



Fonte: Imagem do autor.

3.4 MÉTODO DE SIMULAÇÃO

O protótipo foi conectado em série com um pulmão artificial e um ventilador Mindray SV300 para simular ciclos de diferentes ritmos respiratórios de modo controlado (Figura 5.1). Com isso, foi possível observar a resposta da unidade sensora apresentada no gráfico do aplicativo, realizar análise comparativa com a curva do fluxo inspiratório gerada pelo ventilador e do fluxo expiratório resultante da compressão do pulmão artificial (Figura 5.2). No monitor do ventilador era feita a personalização da pressão em cmH₂O – 1 cmH₂O equivale à aproximadamente 0,0981 kPa – e da frequência em rpm (respirações por minuto) do fluxo de ar gerado, já o tempo do fluxo insuflatório foi mantido a 1,30 segundo em todas as avaliações. Os cenários simulados podem ser vistos abaixo na tabela 1.

Tabela 1: Parâmetros simulados pelo ventilador mecânico SV300

Pressão (cmH ₂ O)	Frequência (rpm)
5	15
6	12
6	17
7	15

10	12
10	14
12	14
13	14
14	14
15	14
16	18
18	14

Em seguida, para validar a eficácia e a robustez do protótipo em condições de uso real, foram realizados testes com voluntários saudáveis. O protocolo estabelecido foi aplicado a uma amostra de 3 participantes, com cada teste dividido em duas fases consecutivas de 3 minutos. A primeira fase consistiu no monitoramento da respiração com o voluntário em repouso, visando estabelecer uma linha de base em condições ideais. Subsequentemente, a segunda fase monitorou o indivíduo durante uma caminhada leve, com o objetivo de avaliar o desempenho do sistema na presença de artefatos de movimento e variações fisiológicas induzidas pelo exercício.

Esta etapa foi fundamental para avaliar o desempenho do sistema fora do ambiente controlado do laboratório, submetendo-o a sinais fisiológicos reais e a artefatos de movimento. Vale destacar que, após os erros de análise da relação de tempo inspirado por tempo expirado (relação I:E) identificados na etapa anterior e que podem ser vistos na tabela 2, a lógica de programação (Apêndice B) para o cálculo da frequência e da relação I:E foram aperfeiçoados para a aplicação dos testes com os voluntários. A modificação se deu em aplicar o método de estados finitos (MEF) que funciona como um modelo de controle no qual o sistema opera em um de vários estados predefinidos e exclusivos, como 'aguardando ciclo', 'inspiração' e 'expiração'. A transição de um estado para outro ocorre somente quando uma condição específica é atendida — neste caso, o sinal de pressão cruzar um limiar determinado. Ao isolar cada fase do ciclo respiratório em um estado distinto, o sistema garante que a contagem de tempo para a inspiração e expiração seja acionada e finalizada em momentos precisos, eliminando a ambiguidade causada por flutuações de sinal e garantindo um cálculo mais robusto da relação I:E.

Figura 5 – Validação experimental do protótipo MA2R: (1) montagem em série com ventilador SV300 e pulmão artificial; (2) Monitores para comparação das curvas de fluxo geradas.



Fonte: Imagem do autor.

4 RESULTADOS

4.1 DADOS DE CALIBRAÇÃO

O processo de tratamento do sinal, executado pelo *firmware*, inicia com uma aquisição de dados brutos por 3 segundos para calcular a média do sinal e determinar o *offset* de calibração. Para atenuar ruídos externos e isolar o sinal respiratório, aplica-se uma técnica de linha de base com suavização de 5% e um limiar de detecção de $\pm 5 \mu\text{V}$. Para fins de visualização no aplicativo, a variação de tensão em relação a este *offset* é então multiplicada por um fator de 1000, otimizando a escala da curva de resposta do sensor. Por fim, os dados processados de tempo e tensão são transmitidos ao cliente BLE a uma taxa de 6 Hz, permitindo a exibição do gráfico (Veja na figura 6) em tempo real. Nos testes de validação, realizados com o ventilador de referência, o sistema demonstrou alta fidelidade, sendo capaz de detectar com clareza as fases de inspiração e expiração, com excelente estabilidade do sinal (Figura 6).

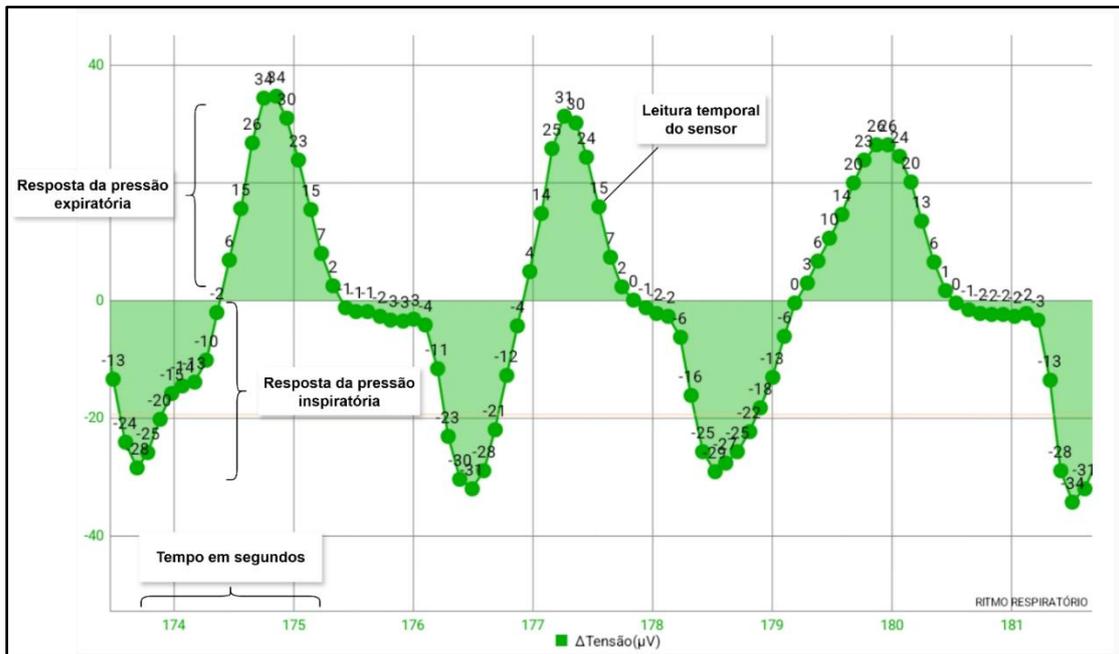
A análise quantitativa da frequência respiratória foi bem-sucedida, apresentando resultados condizentes com as configurações simuladas e um desvio padrão máximo de apenas 0,83. Em contrapartida, apesar do sucesso na detecção do ritmo, a determinação da relação Inspiração:Expiração (I:E) apresentou erros críticos e resultados imprevisíveis, indicando uma limitação na lógica atual. O cálculo dependia da detecção de um único pico (para expiração) ou vale (para inspiração) para iniciar e finalizar a contagem de tempo. O cronômetro era acionado quando o sinal cruzava um limiar e era interrompido assim que o sinal retornava a esse limiar, assumindo que isso marcava o fim da fase respiratória. O problema fundamental dessa abordagem é que, se ocorressem múltiplos picos ou vales dentro de um mesmo ciclo de expiração ou inspiração — algo comum devido a artefatos de movimento ou variações no fluxo de ar —, a lógica não distinguia o verdadeiro fim da fase de uma simples flutuação. Com isso, o sistema finalizava a contagem no primeiro retorno ao limiar, ignorando o restante da fase respiratória e resultando em um cálculo de tempo drasticamente equivocado para este parâmetro específico.

Uma análise aprofundada revelou que a sensibilidade do sistema, calculada em aproximadamente 1,25 $\mu\text{V}/\text{Pa}$ (derivada da especificação do sensor e do ganho do ADC), foi suficiente para distinguir as variações sutis de pressão. Isso confirma que a falha na medição da relação I:E não se deve a limitações do hardware, mas sim a desafios na implementação do algoritmo de software para cenários de fluxo controlado. A tabela a seguir (Tabela 2) apresenta o relatório detalhado da análise do ritmo respiratório nos cenários simulados.

Tabela 2: Dados do relatório de análise do sistema simulado

Simulação	Análise do protótipo	
Frequência(rpm)	Frequência calculada	Relação I:E calculada
12	12,1	6,28
14	14,0	111519,55
15	14,9	1,13
17	16,9	215604,53
18	18,0	118682,70

Figura 6: Gráfico detalhado da curva respiratória adquirida pela unidade sensora



Fonte: imagem do autor

4.2 SIMULAÇÃO DO SISTEMA COM VOLUNTÁRIOS

No cenário de respiração em repouso, o sistema demonstrou um desempenho excelente e estável. Como esperado, o sinal respiratório gerou uma curva sinusoidal clara e rítmica no aplicativo, com mínima interferência de ruído. As Frequências Respiratórias (FR) medidas para os três voluntários situaram-se na faixa de 11.1 a 15.1 respirações por minuto (rpm), valores perfeitamente compatíveis com o padrão fisiológico para um adulto em repouso. Os valores para a relação I:E também foram registrados (variando de 1:0.8 a 1:1.0), embora estejam sujeitos às limitações do algoritmo previamente identificadas. Este resultado validou a capacidade do protótipo de monitorar com precisão a respiração em condições ideais. O cenário de caminhada introduziu um desafio significativo ao sistema: os artefatos de movimento. Observou-se que o sinal bruto no aplicativo apresentou maior variabilidade, com picos de ruído sobrepostos à curva respiratória, atribuídos ao balanço natural do corpo. Apesar disso, o algoritmo de tratamento de sinal foi capaz de filtrar parte dessa interferência, permitindo que o sistema continuasse a calcular a frequência respiratória. Conforme o esperado, a FR aumentou para todos os voluntários, situando-se na faixa de 13.7 a 18.8 rpm, um aumento fisiologicamente consistente com a atividade física leve. A relação I:E também se alterou durante a atividade, reforçando a capacidade do sistema de detectar mudanças no padrão respiratório.

Tabela 3: Resultados dos Testes com Voluntários em Repouso e Caminhada

Voluntário	Aferição em Repouso		Aferição em Caminhada leve	
	FR(rpm)	Relação I:E	FR(rpm)	Relação I:E
Rafaela Queiroz	11.1	1:0.8	13.7	1:1.1
César Motta	12.3	1:0.9	15.1	1:1.2
Ananda Barbosa	15.1	1:1.0	18.8	1:0.9

5. CONCLUSÃO

Diante de um panorama global marcado pelo aumento das doenças respiratórias crônicas, a integração de tecnologias vestíveis ao cuidado em saúde representa uma alternativa inovadora e eficiente. O uso dessas ferramentas no monitoramento da frequência respiratória, aliado a estratégias como o *biofeedback*, tem o potencial de transformar a forma como doenças são geridas, promovendo maior autonomia ao paciente e apoio em tempo real aos profissionais de saúde. O presente trabalho propôs e desenvolveu um sistema de *biofeedback* respiratório não invasivo, baseado em sensor de pressão conectado a um microcontrolador ESP32 com comunicação *Bluetooth Low Energy* (BLE). O sistema demonstrou ser capaz de captar, processar e transmitir com fidelidade o sinal respiratório para um aplicativo mobile, fornecendo ao usuário um retorno visual contínuo e de fácil interpretação. A validação rigorosa do sistema, tanto em ambiente controlado quanto com voluntários, foi crucial para comprovar sua precisão e robustez. A análise confirmou a capacidade do sistema de extrair com exatidão não apenas a Frequência Respiratória (FR), mas também a relação Inspiração:Expiração (I:E), superando os desafios identificados nas fases iniciais do projeto. Um resultado particularmente relevante foi a ausência de prejuízo analítico do cálculo dos parâmetros respiratórios mesmo com a presença de artefatos durante a caminhada, o que atesta a eficácia dos filtros implementados e a viabilidade do uso do dispositivo em cenários dinâmicos do cotidiano. Com a validação bem-sucedida dos parâmetros-chave, os próximos passos para a evolução do projeto podem se concentrar na miniaturização do hardware para otimizar a ergonomia, na realização de simulações para avaliação em populações específicas e na expansão das funcionalidades de *biofeedback* no aplicativo, incluindo diferentes protocolos de treinamento respiratório. Dessa forma, o projeto cumpriu com sucesso seus objetivos ao entregar uma plataforma de *biofeedback* funcional, validada e robusta. A pesquisa reforça a importância da união entre tecnologias vestíveis e sistemas móveis para promover uma medicina mais preventiva e personalizada, oferecendo uma solução promissora para o monitoramento remoto da saúde respiratória.

6. REFERÊNCIAS

- BAE, Byunghoon *et al.* Design optimization of a piezoresistive pressure sensor considering the output signal-to-noise ratio. *Journal of Micromechanics and Microengineering*, v. 14, n. 12, p. 1597, 2004. Disponível em: <https://iopscience.iop.org/article/10.1088/0960-1317/14/12/001>.
- BRINKMAN, Josué E.; TORO, Fadi.; SHARMA, Sandeep. Physiology, respiratory drive. In: StatPearls [Internet]. **Treasure Island (FL): StatPearls Publishing**; 2025. Disponível em: <https://www.ncbi.nlm.nih.gov/books/NBK482414/>
- DE FAZIO, Roberto *et al.* An overview of wearable piezoresistive and inertial sensors for respiration rate monitoring. **Electronics**, v. 10, n. 17, p. 2178, 2021. Disponível em: <https://www.mdpi.com/2079-9292/10/17/2178>
- GALLARDO, Adrián *et al.* Condicionantes del impulso respiratorio. **Rev. chil. anest.**, Santiago, v. 53, n. 3, p. 244-249, 2024 . Disponível em: https://www.scielo.cl/scielo.php?pid=S0719-67922024000300244&script=sci_arttext
- HOSSAIN, Mohammad Mobarak *et al.* Internet of things in pregnancy care coordination and management: A systematic review. **Sensors**, v. 23, n. 23, p. 9367, 2023. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC10708762/>
- HUSSAIN, Tauseef *et al.* Wearable sensors for respiration monitoring: A review. **Sensors**, v. 23, n. 17, p. 7518, 2023. Disponível em: <https://www.mdpi.com/1424-8220/23/17/7518>.
- KAZANSKIY, Nikolay L.; KHONINA, Svetlana N.; BUTT, Muhammad A. A review on flexible wearables—Recent developments in non-invasive continuous health monitoring. **Sensors and Actuators A: Physical**, v. 366, p. 114993, 2024. Disponível em: <https://www.sciencedirect.com/science/article/abs/pii/S0924424723008427?>
- KIRAN KOLLURI, S. S.; ANANIAH DURAI, S. Wearable micro-electro-mechanical systems pressure sensors in health care: Advancements and trends—A review. *IET Wireless Sensor Systems*, v. 14, n. 6, p. 233-247, 2024. Disponível em: <https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/wss2.12084>
- LIU, Haipeng *et al.* Recent development of respiratory rate measurement technologies. **Physiological measurement**, v. 40, n. 7, p. 07TR01, 2019. Disponível em: <https://iopscience.iop.org/article/10.1088/1361-6579/ab299e/pdf>.
- MALIK, Kashif; DUA, Anterpreet. Advancing patient care with biofeedback. In: STATPEARLS [Internet]. Treasure Island (FL): StatPearls Publishing, 2025. Disponível em: <https://pubmed.ncbi.nlm.nih.gov/31971716/>.
- MARQUES, Consuelo Penha Castro *et al.* Epidemiologia da Asma no Brasil, no período de 2016 a 2020. **Research, Society and Development**, v. 11, n. 8, p.

e5211828825-e5211828825, 2022. Disponível em:
<https://rsdjournal.org/index.php/rsd/article/view/28825/26261>

MOON, Kee S.; LEE, Sung Q. A wearable multimodal wireless sensing system for respiratory monitoring and analysis. **Sensors**, v. 23, n. 15, p. 6790, 2023. Disponível em: <https://www.mdpi.com/1424-8220/23/15/6790>.

MOTA, Jennyfer Correa et al. Treinamento muscular inspiratório em pessoas com doença pulmonar obstrutiva crônica (DPOC): uma revisão sistemática. **Fisioterapia e Pesquisa**, v. 30, p. e21028823en, 2023. Disponível em:
<https://www.scielo.br/j/fp/a/8zqXffP8j9B8CjzwBxLMk7N/?lang=pt>.

MUNIR, Tahir et al. A systematic review of internet of things in clinical laboratories: Opportunities, advantages, and challenges. **Sensors**, v. 22, n. 20, p. 8051, 2022. Disponível em: <https://pmc.ncbi.nlm.nih.gov/articles/PMC9611742/>

SHAMSABADI, Ahmadreza et al. Retracted: Internet of things in the management of chronic diseases during the COVID-19 pandemic: A systematic review. **Health science reports**, v. 5, n. 2, p. e557, 2022. Disponível em:
<https://pmc.ncbi.nlm.nih.gov/articles/PMC8919365/>.

ZHOU, Bo; BAUCCELLS COSTA, Alejandro; LUKOWICZ, Paul. Accurate spirometry with integrated barometric sensors in face-worn garments. **Sensors**, v. 20, n. 15, p. 4234, 2020. Disponível em: <https://www.mdpi.com/1424-8220/20/15/4234>.

APÊNDICE A

```
//=====
CÓDIGO-FONTE DO FIRMWARE DO DISPOSITIVO (VERSÃO COM DETECÇÃO DE PICOS/VALES)
// AUTOR: JOÃO GABRIEL DE SOUZA VERAS
// PROJETO: MA2R
// DESCRIÇÃO: ESTA VERSÃO DO FIRMWARE UTILIZA UMA LÓGICA BASEADA NA DETECÇÃO DE PICOS E VALES
PARA IDENTIFICAR OS CICLOS RESPIRATÓRIOS.
//=====

//-----
// 1. INCLUSÃO DE BIBLIOTECAS E DEFINIÇÕES GLOBAIS
//-----
#include <BLEDEVICE.H>
#include <BLESERVER.H>
#include <BLEUTILS.H>
#include <BLE2902.H>
#include <Q2HX711.H>
#include <STRING>

// --- UUIDs (IDENTIFICADORES ÚNICOS UNIVERSAIS) PARA O SERVIÇO BLE ---
// "ENDEREÇOS" QUE O APP USA PARA ENCONTRAR E INTERAGIR COM O SERVIÇO E AS CARACTERÍSTICAS.
#define SERVICE_UUID "2EBFBAC5-DACE-41A8-97CF-FE030C813BA5"
#define CHARACTERISTIC_UUID_RX "F1185818-D548-43AD-A7FB-57610B324DBE"
#define CHARACTERISTIC_UUID_TX "B996B8F2-CEC4-431D-96D4-08B210C112D3"

// --- CONFIGURAÇÃO DO SENSOR E PROCESSAMENTO DE SINAL ---
#define ALPHA 0.05
CONST BYTE MPS_OUT_PIN = 25;
CONST BYTE MPS_SCK_PIN = 27;
CONST FLOAT GANHO = 128, LIMAR = 5; // LIMAR EM µV PARA DETECÇÃO.

Q2HX711 MPS20N0040D(MPS_OUT_PIN, MPS_SCK_PIN); // OBJETO PARA INTERAGIR COM O SENSOR.
FLOAT LINHABASILAR = 0, OFFSET = 0; // VARIÁVEIS DE CALIBRAÇÃO E FILTRAGEM.

// --- VARIÁVEIS PARA LÓGICA DE DETECÇÃO DE PICOS E VALES ---
INT CONTADORPICOS = 0, CONTADORVALES = 0; // CONTADORES PARA PICOS (EXPIRAÇÃO) E VALES
(INSPIRAÇÃO).
```

```
UNSIGNED LONG TEMPOINSPIRACAO = 0, TEMPOEXPIRACAO = 0; // ACUMULADORES DE TEMPO PARA
CÁLCULO DA RELAÇÃO I:E.
```

```
// FLAGS PARA MONITORAR O ESTADO ATUAL DA RESPIRAÇÃO.
```

```
BOOL EMINSPIRACAO = FALSE, EMEXPIRACAO = FALSE;
```

```
UNSIGNED LONG TEMPODEINICIOINSPIRACAO = 0, TEMPODEINICIOEXPIRACAO = 0;
```

```
// --- VARIÁVEIS DE CONTROLE DO BLE E DA APLICAÇÃO ---
```

```
BLECHARACTERISTIC *RXCHARACTERISTIC;
```

```
BLECHARACTERISTIC *TXCHARACTERISTIC;
```

```
BOOL ISREADING = FALSE; // CONTROLA O ESTADO DE MONITORAMENTO (ATIVO/INATIVO).
```

```
UNSIGNED LONG STARTATIVIDADE; // REGISTRA O TEMPO DE INÍCIO DO MONITORAMENTO.
```

```
BOOL REPORTREQUEST = FALSE; // INDICA SE UM RELATÓRIO FOI SOLICITADO.
```

```
BOOL ACKRECEIVED = FALSE; // FLAG PARA O HANDSHAKE DO RELATÓRIO.
```

```
CONST UIN32_T SENHABLE = 768002; // PIN ESTÁTICO PARA EMPARELHAMENTO SEGURO.
```

```
//-----
```

```
// 2. FUNÇÕES AUXILIARES
```

```
//-----
```

```
/**
```

```
* @BRIEF CONVERTE O VALOR BRUTO DO ADC PARA TENSÃO (MV).
```

```
*/
```

```
float RAW_VOLTAGE(float BITS, float GANHO) {
    return (BITS * 3.2 * 1000) / (GANHO * (pow(2, 24) - 1));
}
```

```
/**
```

```
* @BRIEF CALIBRA O OFFSET INICIAL DO SENSOR LENDO VALORES EM REPOUSO POR 3 SEGUNDOS.
```

```
*/
```

```
void CALIBRAROFFSET() {
    float SOMA = 0, CONTADOR = 0;
    unsigned long STARTTIME = millis();
    SERIAL.PRINTLN("INICIANDO CALIBRAÇÃO DO OFFSET... AGUARDANDO 3 SEGUNDOS...");
```

```
while (millis() - STARTTIME <= 3000) {
    SOMA += RAW_VOLTAGE(MPS20N0040D.READ(), GANHO);
    CONTADOR++;
}
```

```

    DELAY(10);
}
OFFSET = SOMA / CONTADOR;
LINHABASILAR = 0;
}

/**
 * @BRIEF CALCULA A LINHA DE BASE MÓVEL PARA REMOVER DRIFTS E RUÍDOS DE BAIXA FREQUÊNCIA.
 */
FLOAT CALCULARLINHABASILAR(FLOAT LEITURAATUAL) {
    LINHABASILAR = (ALPHA * LEITURAATUAL) + ((1 - ALPHA) * LINHABASILAR);
    RETURN LINHABASILAR;
}

/**
 * @BRIEF CALCULA, FORMATA E ENVIA O RELATÓRIO FINAL PARA O APLICATIVO COM HANDSHAKE.
 */
VOID ENVIARRELATORIO(FLOAT TOTALTEMPO) {
    // CALCULA A FREQUÊNCIA RESPIRATÓRIA COMO A MÉDIA DE PICOS E VALES DETECTADOS.
    FLOAT FREQUENCIARESPIRATORIA = (((FLOAT)CONTADORPICOS + CONTADORVALES) / 2) / (TOTALTEMPO /
60000.0);
    // CALCULA A RELAÇÃO I:E. NOTE QUE ESTA VERSÃO CALCULA A RAZÃO INSPIRAÇÃO/EXPIRAÇÃO.
    FLOAT RELACAOIE = (TEMPOSINSPIRACAO > 0 && TEMPOSEXPIRACAO > 0) ? (FLOAT)TEMPOSINSPIRACAO /
TEMPOSEXPIRACAO : 0;

    // CLASSIFICA O RITMO RESPIRATÓRIO COM BASE NA FREQUÊNCIA.
    STRING CONDICAO="";
    IF (FREQUENCIARESPIRATORIA < 12) {
        CONDICAO = "BRADIPNEIA";
    } ELSE IF (FREQUENCIARESPIRATORIA <= 20) {
        CONDICAO = "NORMAL";
    } ELSE {
        CONDICAO = "TAQUIPNEIA";
    }

    SERIAL.PRINTLN("ENVIANDO RELATÓRIO COM HANDSHAKE...");

    // PREPARA AS 5 LINHAS DO RELATÓRIO.

```

```

STRING LINHASRELATORIO[5];
LINHASRELATORIO[0] = "RELATÓRIO:\n\n";
LINHASRELATORIO[1] = "FRM: " + STRING(FREQUENCIARESPIRATORIA, 1) + "\n";
LINHASRELATORIO[2] = "REL. I:E = " + STRING(RELACAOIE, 2) + "\n";
LINHASRELATORIO[3] = "COND.: " + CONDICAO + "\n";
LINHASRELATORIO[4] = "TEMPO: " + STRING(TOTALTEMPO / 60000.0, 1) + " MIN";

// LOOP PARA ENVIAR CADA LINHA COM CONFIRMAÇÃO (HANDSHAKE).
FOR (INT I = 0; I < 5; I++) {
    ACKRECEIVED = FALSE;
    TXCHARACTERISTIC->SETVALUE(LINHASRELATORIO[I].C_STR());
    TXCHARACTERISTIC->NOTIFY();
    SERIAL.PRINT("ENVIADO: "); SERIAL.PRINTLN(LINHASRELATORIO[I]);

    // AGUARDA O ACK DO APP POR ATÉ 1 SEGUNDO.
    LONG STARTTIME = MILLIS();
    WHILE (!ACKRECEIVED && (MILLIS() - STARTTIME < 1000)) {
        DELAY(50);
    }

    IF (!ACKRECEIVED) {
        SERIAL.PRINTLN("TIMEOUT! ACK NÃO RECEBIDO.");
        BREAK; // ABORTA SE O APP NÃO RESPONDER.
    }
    SERIAL.PRINTLN("ACK RECEBIDO!");
}

// ENVIA A MARCA DE FIM DE TRANSMISSÃO.
DELAY(50);
TXCHARACTERISTIC->SETVALUE("#FIM#");
TXCHARACTERISTIC->NOTIFY();
SERIAL.PRINTLN("MARCA DE FIM ENVIADA.");
}

//-----
// 3. CLASSES DE CALLBACKS DO BLE
//-----

```

```

// GERENCIA EVENTOS DE CONEXÃO/DESCONEXÃO DO SERVIDOR.
CLASS MYSERVERCALLBACKS: PUBLIC BLESERVERCALLBACKS {
    VOID ONCONNECT(BLESERVER* PSERVER) { SERIAL.PRINTLN("CLIENTE CONECTADO!"); }
    VOID ONDISCONNECT(BLESERVER* PSERVER) { SERIAL.PRINTLN("CLIENTE DESCONECTADO! REINICIANDO
O ADVERTISING..."); BLEDEVICE::STARTADVERTISING(); }
};

// GERENCIA EVENTOS DE ESCRITA (COMANDOS RECEBIDOS DO APP).
CLASS MYCALLBACKS : PUBLIC BLECHARACTERISTICCALLBACKS {
    VOID ONWRITE(BLECHARACTERISTIC *PCHARACTERISTIC) {
        STRING COMMAND = PCHARACTERISTIC->GETVALUE();

        IF (COMMAND == "INICIAR") {
            ISREADING = TRUE;
            REPORTREQUEST = FALSE;
            // RESETA TODAS AS VARIÁVEIS DE CONTAGEM E TEMPO PARA UMA NOVA MEDIÇÃO.
            CONTADORPICOS = 0;
            CONTADORVALES = 0;
            TEMPOSINSPIRACAO = 0;
            TEMPOSEXPIRACAO = 0;
            EMINSPIRACAO = FALSE;
            EMEXPIRACAO = FALSE;
            STARTATIVIDADE = MILLIS();
            SERIAL.PRINTLN("LEITURA INICIADA!");
        }
        ELSE IF (COMMAND == "INTERROMPER" || COMMAND == "GERAR_RELATORIO"){
            ISREADING = FALSE;
            REPORTREQUEST = TRUE; // SINALIZA PARA O LOOP PRINCIPAL ENVIAR O RELATÓRIO.
            SERIAL.PRINTLN("SOLICITAÇÃO DE INTERRUPÇÃO/RELATÓRIO RECEBIDA!");
        } ELSE IF(COMMAND == "ACK"){
            ACKRECEIVED = TRUE; // CONFIRMAÇÃO DO APP PARA O HANDSHAKE.
        }
    }
};

// GERENCIA O PROCESSO DE EMPARELHAMENTO SEGURO.
CLASS MYSECURITY : PUBLIC BLESECURITYCALLBACKS {

```

```

UINT32_T ONPASSKEYREQUEST() { SERIAL.PRINTF("PIN SOLICITADO: %06D\n", SENHABLE); RETURN
SENHABLE; }

VOID ONPASSKEYNOTIFY(UINT32_T SENHABLE) { SERIAL.PRINTLN("PIN NOTIFICADO."); }
BOOL ONCONFIRMPIN(UINT32_T PIN) { SERIAL.PRINTF("PIN CONFIRMADO: %06D\n", PIN); RETURN TRUE; }
BOOL ONSECURITYREQUEST() { SERIAL.PRINTLN("SEGURANÇA SOLICITADA."); RETURN TRUE; }
VOID ONAUTHENTICATIONCOMPLETE(ESP_BLE_AUTH_CMPL_T CMPL) { SERIAL.PRINTLN(CMPL.SUCCESS ?
"EMPARELHAMENTO CONCLUÍDO!" : "EMPARELHAMENTO FALHO."); }
};

//-----
// 4. FUNÇÃO SETUP - INICIALIZAÇÃO DO SISTEMA
//-----
VOID SETUP() {
    SERIAL.BEGIN(115200);

    // INICIALIZA O DISPOSITIVO BLE COM SEU NOME E CONFIGURAÇÕES DE SEGURANÇA.
    BLEDEVICE::INIT("MA2R BLE");
    BLEDEVICE::SETEncryptionLevel(ESP_BLE_SEC_ENCRYPT);
    BLEDEVICE::SETSecurityCallbacks(new MySecurity());

    // DEFINE O MODO DE AUTENTICAÇÃO E O PIN ESTÁTICO.
    BLESECURITY *PSECURITY = new BLESECURITY();
    PSECURITY->SETAuthenticationMode(ESP_LE_AUTH_REQ_SC_BOND);
    PSECURITY->SETCapability(ESP_IO_CAP_OUT);
    PSECURITY->SETKeySize(16);
    PSECURITY->SETStaticPIN(SENHABLE);
    SERIAL.PRINTF("PIN FIXO CONFIGURADO: %06D\n", SENHABLE);

    // CRIA O SERVIDOR, SERVIÇO E AS CARACTERÍSTICAS RX E TX.
    BLESERVER *PSERVER = BLEDEVICE::CREATE_SERVER();
    PSERVER->SETCALLBACKS(new MyServerCallbacks());
    BLESERVICE *PSERVICE = PSERVER->CREATE_SERVICE(SERVICE_UUID);

    RXCHARACTERISTIC = PSERVICE->CREATE_CHARACTERISTIC(CARACTERISTIC_UUID_RX,
BLECHARACTERISTIC::PROPERTY_WRITE);
    RXCHARACTERISTIC->SETCALLBACKS(new MyCallbacks());

```

```

TXCHARACTERISTIC = PSERVICE->CREATECHARACTERISTIC(CHARACTERISTIC_UUID_TX,
BLECHARACTERISTIC::PROPERTY_NOTIFY);
TXCHARACTERISTIC->ADDDESRIPTOR(NEW BLE2902());

// INICIA O SERVIÇO E O ADVERTISING PARA QUE O DISPOSITIVO SEJA DETECTÁVEL.
PSERVICE->START();
BLEDEVICE::STARTADVERTISING();
SERIAL.PRINTLN("ESP32 BLE INICIADO E AGUARDANDO CONEXÃO!");

CALIBRAROFFSET(); // REALIZA A CALIBRAÇÃO ANTES DE ENTRAR NO LOOP.
}

//-----
// 5. FUNÇÃO LOOP - LÓGICA PRINCIPAL DE EXECUÇÃO
//-----
VOID LOOP() {
// --- BLOCO 1: EXECUTADO APENAS QUANDO O MONITORAMENTO ESTÁ ATIVO (^ISREADING == TRUE) ---
IF (ISREADING) {
    UNSIGNED LONG TEMPOATUAL = MILLIS();
    // PROCESSAMENTO DE SINAL: LÊ, REMOVE OFFSET, CALCULA BASELINE E ISOLA O SINAL RESPIRATÓRIO.
    FLOAT DV = (RAW_VOLTAGE(MPS20N0040D.READ(), GANHO) - OFFSET) * 1000; // µV
    FLOAT BASELINE = CALCULARLINHABASILAR(DV);
    FLOAT Vf = DV - BASELINE;

    // ENVIA A LEITURA FORMATADA (TEMPO, TENSÃO) VIA BLE PARA PLOTAGEM DO GRÁFICO NO APP.
    STRING LEITURA = STRING((TEMPOATUAL - STARTATIVIDADE) / 1000.0) + "," + STRING(Vf, 2) + "\n";
    TXCHARACTERISTIC->SETVALUE(LEITURA.C_STR());
    TXCHARACTERISTIC->NOTIFY();
    SERIAL.PRINTLN(LEITURA);

    // --- LÓGICA DE DETECÇÃO DE PICOS E VALES ---
    // ESTA LÓGICA NÃO USA UMA FSM, MAS SIM FLAGS PARA DETECTAR AS TRANSIÇÕES DE ESTADO.

    // MONITORA A INSPIRAÇÃO (DETECÇÃO DE UM VALE).
    IF (Vf < -LIMIAR) { // SE A TENSÃO FICA SUFICIENTEMENTE NEGATIVA...
        IF (!EMINSPIRACAO) {
            // ...E AINDA NÃO ESTÁVAMOS EM INSPIRAÇÃO, MARCA O INÍCIO DE UMA.
            EMINSPIRACAO = TRUE;

```

```

    EMEXPIRACAO = FALSE; // GARANTE EXCLUSIVIDADE DE ESTADO.
    TEMPODEINICIOINSPIRACAO = TEMPOATUAL;
}
} ELSE IF (EMINSPIRACAO && Vf >= 0) { // SE ESTÁVAMOS INSPIRANDO E A TENSÃO RETORNA A ZERO...
    // ...MARCA O FIM DA INSPIRAÇÃO.
    EMINSPIRACAO = FALSE;
    TEMPOSINSPIRACAO += (TEMPOATUAL - TEMPODEINICIOINSPIRACAO); // ACUMULA A DURAÇÃO.
    CONTADORVALES++; // INCREMENTA O CONTADOR DE INSPIRAÇÕES (VALES).
}

// MONITORA A EXPIRAÇÃO (DETECÇÃO DE UM PICO).
IF (Vf > LIMAR) { // SE A TENSÃO FICA SUFICIENTEMENTE POSITIVA...
    IF (!EMEXPIRACAO) {
        // ...E AINDA NÃO ESTÁVAMOS EM EXPIRAÇÃO, MARCA O INÍCIO DE UMA.
        EMEXPIRACAO = TRUE;
        EMINSPIRACAO = FALSE; // GARANTE EXCLUSIVIDADE DE ESTADO.
        TEMPODEINICIOEXPIRACAO = TEMPOATUAL;
    }
} ELSE IF (EMEXPIRACAO && Vf <= 0) { // SE ESTÁVAMOS EXPIRANDO E A TENSÃO RETORNA A ZERO...
    // ...MARCA O FIM DA EXPIRAÇÃO.
    EMEXPIRACAO = FALSE;
    TEMPOSEXPIRACAO += (TEMPOATUAL - TEMPODEINICIOEXPIRACAO); // ACUMULA A DURAÇÃO.
    CONTADORPICOS++; // INCREMENTA O CONTADOR DE EXPIRAÇÕES (PICOS).
}

DELAY(50); // CONTROLA A TAXA DE AMOSTRAGEM E ENVIO PARA ~20 Hz.

// --- BLOCO 2: EXECUTADO APENAS QUANDO UM RELATÓRIO É SOLICITADO ---
} ELSE IF(REPORTREQUEST){
    FLOAT TOTALTEMPO = MILLIS() - STARTATIVIDADE;
    ENVIARRELATORIO(TOTALTEMPO);
    REPORTREQUEST = FALSE; // RESETA A FLAG PARA NÃO ENVIAR NOVAMENTE.

// --- BLOCO 3: EXECUTADO QUANDO O SISTEMA ESTÁ INATIVO ---
} ELSE {
    DELAY(1000); // REDUZ O PROCESSAMENTO QUANDO O SISTEMA ESTÁ OCIOSO.
}
}

```

APÊNDICE B

```

//=====
=====
// CÓDIGO-FONTE ATUALIZADO
// AUTOR: JOÃO GABRIEL DE SOUZA VERAS
// PROJETO: MA2R VERSÃO FINAL
// DESCRIÇÃO: ESTE FIRMWARE É RESPONSÁVEL PELA AQUISIÇÃO DE DADOS DO SENSOR DE
//     PRESSÃO, PROCESSAMENTO DO SINAL RESPIRATÓRIO, ANÁLISE DO RITMO E
//     COMUNICAÇÃO VIA BLUETOOTH LOW ENERGY (BLE) COM O APLICATIVO MOBILE.
//=====
=====

//-----
// 1. INCLUSÃO DE BIBLIOTECAS
//-----
#include <BLEDEVICE.H> // BIBLIOTECA PRINCIPAL PARA FUNCIONALIDADES BLE.
#include <BLESERVER.H> // PARA CRIAR UM SERVIDOR BLE NO ESP32.
#include <BLEUTILS.H> // UTILITÁRIOS PARA O BLE.
#include <BLE2902.H> // PARA ADICIONAR DESCRITORES ÀS CARACTERÍSTICAS BLE (NECESSÁRIO PARA
NOTIFICAÇÕES).
#include <Q2HX711.H> // BIBLIOTECA PARA INTERFACE COM O ADC HX711, QUE LÊ O SENSOR.
#include <STRING> // PARA MANIPULAÇÃO DE STRINGS.

//-----
// 2. DEFINIÇÕES E VARIÁVEIS GLOBAIS
//-----

// --- MÁQUINA DE ESTADOS FINITOS (FSM) PARA O CICLO RESPIRATÓRIO ---
// ENUMERAÇÃO QUE DEFINE OS POSSÍVEIS ESTADOS DA RESPIRAÇÃO, TORNANDO O CÓDIGO MAIS LEGÍVEL.
ENUM ESTADORESPIRATORIO {
    AGUARDANDO_CICLO, // ESTADO INICIAL, AGUARDANDO O INÍCIO DA INSPIRAÇÃO.
    EM_INSPIRACAO, // O USUÁRIO ESTÁ INSPIRANDO.
    EM_EXPIRACAO // O USUÁRIO ESTÁ EXPIRANDO.
};

// --- UUIDS (IDENTIFICADORES ÚNICOS UNIVERSAIS) PARA O SERVIÇO BLE ---

```

```
// "ENDEREÇOS" PADRONIZADOS QUE O APLICATIVO USA PARA ENCONTRAR E INTERAGIR COM O SERVIÇO E AS
CARACTERÍSTICAS CORRETAS.
```

```
#DEFINE SERVICE_UUID "2EBFBAC5-DACE-41A8-97CF-FE030C813BA5" // UUID DO SERVIÇO
PRINCIPAL.
```

```
#DEFINE CHARACTERISTIC_UUID_RX "F1185818-D548-43AD-A7FB-57610B324DBE" //
CARACTERÍSTICA PARA RECEBER DADOS DO APP (RX).
```

```
#DEFINE CHARACTERISTIC_UUID_TX "B996B8F2-CEC4-431D-96D4-08B210C112D3" //
CARACTERÍSTICA PARA TRANSMITIR DADOS PARA O APP (TX).
```

```
// --- CONFIGURAÇÃO DO SENSOR DE PRESSÃO E PROCESSAMENTO DE SINAL ---
```

```
#DEFINE ALPHA 0.05 // FATOR DE SUAUIZAÇÃO PARA O CÁLCULO DA LINHA DE BASE MÓVEL.
```

```
CONST BYTE MPS_OUT_PIN = 25; // PINO DE DADOS (DOUT) DO HX711.
```

```
CONST BYTE MPS_SCK_PIN = 27; // PINO DE CLOCK (SCK) DO HX711.
```

```
CONST FLOAT GANHO = 128, LIMAR = 5; // GANHO DO AMPLIFICADOR DO HX711 E LIMAR (EM µV) PARA
DETECÇÃO DE INSPIRAÇÃO/EXPIRAÇÃO.
```

```
Q2HX711 MPS20N0040D(MPS_OUT_PIN, MPS_SCK_PIN); // OBJETO PARA INTERAGIR COM O SENSOR
ATRAVÉS DO HX711.
```

```
FLOAT LINHABASILAR = 0, OFFSET = 0; // VARIÁVEIS PARA ARMAZENAR O OFFSET DE CALIBRAÇÃO E A
LINHA DE BASE DINÂMICA.
```

```
// --- VARIÁVEIS PARA A LÓGICA DE ANÁLISE DE CICLOS RESPIRATÓRIOS ---
```

```
ESTADORESPIRATORIO ESTADOATUAL = AGUARDANDO_CICLO; // ESTADO ATUAL DA FSM.
```

```
INT CONTADORCICLOSCOMPLETOS = 0; // CONTA QUANTOS CICLOS RESPIRATÓRIOS COMPLETOS
FORAM DETECTADOS.
```

```
UNSIGNED LONG TEMPOINICIOINSPIRACAO TOTAL = 0, TEMPO TOTALINSPIRACAO = 0; // ARMAZENAM TEMPOS
PARA CÁLCULO DA RELAÇÃO I:E.
```

```
UNSIGNED LONG TEMPOINICIOEXPIRACAO TOTAL = 0, TEMPO TOTALEXPIRACAO = 0;
```

```
// --- VARIÁVEIS DE CONTROLE DO BLE E DA APLICAÇÃO ---
```

```
BLECHARACTERISTIC *RXCHARACTERISTIC; // PONTEIRO PARA A CARACTERÍSTICA DE RECEPÇÃO.
```

```
BLECHARACTERISTIC *TXCHARACTERISTIC; // PONTEIRO PARA A CARACTERÍSTICA DE TRANSMISSÃO.
```

```
BOOL ISREADING = FALSE; // FLAG QUE CONTROLA SE O SISTEMA ESTÁ OU NÃO EM MODO DE LEITURA.
```

```
UNSIGNED LONG STARTATIVIDADE; // REGISTRA O TEMPO DE INÍCIO DA ATIVIDADE DE MONITORAMENTO.
```

```
BOOL REPORTREQUEST = FALSE; // FLAG QUE INDICA SE UM RELATÓRIO FOI SOLICITADO PELO APP.
```

```
BOOL ACKRECEIVED = FALSE; // FLAG PARA O HANDSHAKE DO RELATÓRIO, CONFIRMA SE O APP
RECEBEU A LINHA.
```

```
CONST UINT32_T SENHABLE = 768002; // PIN ESTATICO PARA O EMPARELHAMENTO SEGURO VIA BLE.
```

```

//-----
// 3. FUNÇÕES AUXILIARES
//-----

/**
 * @BRIEF CONVERTE O VALOR BRUTO LIDO PELO ADC (EM BITS) PARA TENSÃO (EM MV).
 * @PARAM BITS VALOR LIDO DO ADC.
 * @PARAM GANHO GANHO CONFIGURADO NO AMPLIFICADOR DO HX711.
 * @RETURN O VALOR DA TENSÃO EM MILIVOLTS (MV).
 */
FLOAT RAW_VOLTAGE(FLOAT BITS, FLOAT GANHO) {
    RETURN (BITS * 3.2 * 1000) / (GANHO * (POW(2, 24) - 1));
}

/**
 * @BRIEF REALIZA A CALIBRAÇÃO INICIAL DO SISTEMA.
 * LÊ O SENSOR POR 3 SEGUNDOS EM REPOUSO E CALCULA A MÉDIA PARA DEFINIR O 'OFFSET'.
 * ESTE OFFSET REPRESENTA O "ZERO" DO SISTEMA, REMOVENDO QUALQUER TENSÃO RESIDUAL.
 */
VOID CALIBRAROFFSET() {
    FLOAT SOMA = 0, CONTADOR = 0;
    UNSIGNED LONG STARTTIME = MILLIS();
    SERIAL.PRINTLN("INICIANDO CALIBRAÇÃO DO OFFSET... AGUARDANDO 3 SEGUNDOS...");

    WHILE (MILLIS() - STARTTIME <= 3000) {
        SOMA += RAW_VOLTAGE(MPS20N0040D.READ(), GANHO);
        CONTADOR++;
        DELAY(10);
    }
    OFFSET = SOMA / CONTADOR;
    LINHABASILAR = 0; // RESETA A LINHA DE BASE APÓS A CALIBRAÇÃO.
}

/**
 * @BRIEF CALCULA A LINHA DE BASE MÓVEL (MOVING BASELINE).
 * APLICA UM FILTRO PASSA-BAIXA PARA CRIAR UMA REFERÊNCIA DINÂMICA QUE REMOVE
 * DRIFTS E RUÍDOS DE BAIXA FREQUÊNCIA DO SINAL.
 */

```

```

* @PARAM LEITURAATUAL A LEITURA DE TENSÃO ATUAL (JÁ SUBTRAÍDO O OFFSET).
* @RETURN O VALOR DA LINHA DE BASE ATUALIZADO.
*/
FLOAT CALCULARLINHABASILAR(FLOAT LEITURAATUAL) {
    LINHABASILAR = (ALPHA * LEITURAATUAL) + ((1 - ALPHA) * LINHABASILAR);
    RETURN LINHABASILAR;
}

/**
* @BRIEF CALCULA OS PARÂMETROS FINAIS, FORMATA E ENVIA O RELATÓRIO PARA O APLICATIVO.
*   UTILIZA UM MECANISMO DE HANDSHAKE COM ACK PARA GARANTIR A ENTREGA CONFIÁVEL.
* @PARAM TOTALTEMPO TEMPO TOTAL DA ATIVIDADE DE MONITORAMENTO EM MILISSEGUNDOS.
*/
VOID ENVIARRELATORIO(FLOAT TOTALTEMPO) {
    // CALCULA OS PARÂMETROS PRIMÁRIOS: FREQUÊNCIA RESPIRATÓRIA E RELAÇÃO I:E.
    FLOAT FREQUENCIARESPIRATORIA = (TOTALTEMPO > 0) ? (FLOAT)CONTADORCICLOSCOMPLETOS /
(TOTALTEMPO / 60000.0) : 0;
    FLOAT RELACAOIE_VALOR = (TEMPOTOTALINSPIRACAO > 0 && TEMPOTOTALEXPIRACAO > 0) ?
(FLOAT)TEMPOTOTALEXPIRACAO / TEMPOTOTALINSPIRACAO : 0;

    STRING CONDICAO = "";
    // CLASSIFICA O RITMO RESPIRATÓRIO COM BASE NOS PARÂMETROS CALCULADOS.
    IF (RELACAOIE_VALOR > 3.0) { // PRIORIZA A DETECÇÃO DE PADRÃO OBSTRUTIVO (EXPIRAÇÃO LONGA).
        CONDICAO = "SPO";
    } ELSE { // SE NÃO FOR OBSTRUTIVO, CLASSIFICA PELA FREQUÊNCIA.
        IF (FREQUENCIARESPIRATORIA < 12) {
            CONDICAO = "BRADIPNEIA";
        } ELSE IF (FREQUENCIARESPIRATORIA <= 20) {
            CONDICAO = "NORMAL";
        } ELSE {
            CONDICAO = "TAQUIPNEIA";
        }
    }
}

SERIAL.PRINTLN("ENVIANDO RELATÓRIO COM HANDSHAKE...");

// PREPARA AS 5 LINHAS DO RELATÓRIO PARA ENVIO.
STRING LINHASRELATORIO[5];

```

```

LINHASRELATORIO[0] = "RELATÓRIO:\n\n";
LINHASRELATORIO[1] = "FRM: " + STRING(FREQUENCIARESPIRATORIA, 1) + " RPM\n";
LINHASRELATORIO[2] = "REL. I:E = 1:" + STRING(RELACAOIE_VALOR, 1) + "\n";
LINHASRELATORIO[3] = "COND.: " + CONDICAO + "\n";
LINHASRELATORIO[4] = "TEMPO TOTAL: " + STRING(TOTALTEMPO / 1000.0, 1) + " s";

// LOOP PARA ENVIAR CADA LINHA DO RELATÓRIO COM HANDSHAKE.
FOR (INT I = 0; I < 5; I++) {
  ACKRECEIVED = FALSE; // RESETA A FLAG DE CONFIRMAÇÃO.
  TXCHARACTERISTIC->SETVALUE(LINHASRELATORIO[I].C_STR());
  TXCHARACTERISTIC->NOTIFY(); // ENVIA A LINHA PARA O APP.
  SERIAL.PRINT("ENVIADO: "); SERIAL.PRINTLN(LINHASRELATORIO[I].C_STR());

  // AGUARDA O ACK DO APLICATIVO POR ATÉ 1 SEGUNDO.
  LONG STARTTIME = MILLIS();
  WHILE (!ACKRECEIVED && (MILLIS() - STARTTIME < 1000)) {
    DELAY(50);
  }

  IF (!ACKRECEIVED) { // SE O ACK NÃO CHEGAR, INTERROMPE O ENVIO.
    SERIAL.PRINTLN("TIMEOUT! ACK NÃO RECEBIDO.");
    BREAK;
  }
  SERIAL.PRINTLN("ACK RECEBIDO!");
}

// ENVIA UMA MARCA DE FIM DE TRANSMISSÃO PARA O APP SABER QUE O RELATÓRIO TERMINOU.
DELAY(50);
TXCHARACTERISTIC->SETVALUE("#FIM#");
TXCHARACTERISTIC->NOTIFY();
SERIAL.PRINTLN("MARCA DE FIM ENVIADA.");
}

//-----
// 4. CLASSES DE CALLBACKS DO BLE
// (GERENCIAMENTO DE EVENTOS BLE)
//-----

```

```

// GERENCIA EVENTOS DE CONEXÃO E DESCONEXÃO DO SERVIDOR.
CLASS MYSERVERCALLBACKS: PUBLIC BLESERVERCALLBACKS {
    VOID ONCONNECT(BLESERVER* PSERVER) { SERIAL.PRINTLN("CLIENTE CONECTADO!"); }
    VOID ONDISCONNECT(BLESERVER* PSERVER) { SERIAL.PRINTLN("CLIENTE DESCONECTADO! REINICIANDO
O ADVERTISING..."); BLEDEVICE::STARTADVERTISING(); }
};

// GERENCIA O PROCESSO DE EMPARELHAMENTO SEGURO (PIN, ETC.).
CLASS MYSECURITY : PUBLIC BLESECURITYCALLBACKS {
    UINT32_T ONPASSKEYREQUEST() { SERIAL.PRINTF("PIN SOLICITADO: %06d\n", SENHABLE); RETURN
SENHABLE; }
    VOID ONPASSKEYNOTIFY(UINT32_T SENHABLE) { SERIAL.PRINTLN("PIN NOTIFICADO."); }
    BOOL ONCONFIRMPIN(UINT32_T PIN) { SERIAL.PRINTF("PIN CONFIRMADO: %06d\n", PIN); RETURN TRUE; }
    BOOL ONSECURITYREQUEST() { SERIAL.PRINTLN("SEGURANÇA SOLICITADA."); RETURN TRUE; }
    VOID ONAUTHENTICATIONCOMPLETE(ESP_BLE_AUTH_CMPL_T CMPL) { SERIAL.PRINTLN(CMPL.SUCCESS ?
"EMPARELHAMENTO CONCLUÍDO!" : "EMPARELHAMENTO FALHO."); }
};

// GERENCIA EVENTOS DE ESCRITA NA CARACTERÍSTICA RX (RECEBIMENTO DE COMANDOS DO APP).
CLASS MYCALLBACKS : PUBLIC BLECHARACTERISTICCALLBACKS {
    VOID ONWRITE(BLECHARACTERISTIC *PCHARACTERISTIC) {
        STRING COMMAND = PCHARACTERISTIC->GETVALUE();

        IF (COMMAND == "INICIAR") {
            ISREADING = TRUE;
            REPORTREQUEST = FALSE;
            // RESETA TODAS AS VARIÁVEIS DE CICLO PARA UMA NOVA MEDIÇÃO LIMPA.
            ESTADOATUAL = AGUARDANDO_CICLO;
            CONTADORCICLOSCOMPLETOS = 0;
            TEMPOTOTALINSPIRACAO = 0;
            TEMPOTOTALEXPIRACAO = 0;
            STARTATIVIDADE = MILLIS();
            SERIAL.PRINTLN("LEITURA INICIADA!");
        }
        ELSE IF (COMMAND == "INTERROMPER" || COMMAND == "GERAR_RELATORIO"){
            ISREADING = FALSE;
            REPORTREQUEST = TRUE; // SINALIZA PARA O LOOP PRINCIPAL QUE UM RELATÓRIO DEVE SER ENVIADO.
            SERIAL.PRINTLN("SOLICITAÇÃO DE INTERRUPÇÃO/RELATÓRIO RECEBIDA!");
        }
    }
};

```

```

} ELSE IF(COMMAND == "ACK"){
  ACKRECEIVED = TRUE; // CONFIRMAÇÃO DO APP PARA O HANDSHAKE DO RELATÓRIO.
}
}
};

//-----
// 5. FUNÇÃO SETUP - INICIALIZAÇÃO DO SISTEMA
//-----
VOID SETUP() {
  SERIAL.BEGIN(115200);

  // --- INICIALIZAÇÃO DO SERVIDOR BLE ---
  BLEDEVICE::INIT("MA2R BLE"); // DEFINE O NOME DO DISPOSITIVO BLE.
  // CONFIGURA A SEGURANÇA PARA EXIGIR CRIPTOGRAFIA.
  BLEDEVICE::SETENCRYPTIONLEVEL(ESP_BLE_SEC_ENCRYPT);
  BLEDEVICE::SETSECURITYCALLBACKS(NEW MYSECURITY());

  // DEFINE O MODO DE AUTENTICAÇÃO (EMPARELHAMENTO COM PIN).
  BLESECURITY *PSECURITY = NEW BLESECURITY();
  PSECURITY->SETAUTHENTICATIONMODE(ESP_LE_AUTH_REQ_SC_BOND);
  PSECURITY->SETCAPABILITY(ESP_IO_CAP_OUT); // CAPACIDADE DE EXIBIR UM PIN.
  PSECURITY->SETKEYSIZE(16);
  PSECURITY->SETSTATICPIN(SENHABLE);
  SERIAL.PRINTF("PIN FIXO CONFIGURADO: %06D\n", SENHABLE);

  // CRIA O SERVIDOR E O SERVIÇO BLE.
  BLESERVER *PSERVER = BLEDEVICE::CREATESERVER();
  PSERVER->SETCALLBACKS(NEW MYSERVERCALLBACKS());
  BLESERVICE *PSERVICE = PSERVER->CREATESERVICE(SERVICE_UUID);

  // CRIA A CARACTERÍSTICA RX PARA RECEBER COMANDOS.
  RXCHARACTERISTIC = PSERVICE->CREATECHARACTERISTIC(CHARACTERISTIC_UUID_RX,
  BLECHARACTERISTIC::PROPERTY_WRITE);
  RXCHARACTERISTIC->SETCALLBACKS(NEW MYCALLBACKS());

  // CRIA A CARACTERÍSTICA TX PARA ENVIAR DADOS, COM PROPRIEDADE DE NOTIFICAÇÃO.

```

```

TXCHARACTERISTIC = PSERVICE->CREATECHARACTERISTIC(CHARACTERISTIC_UUID_TX,
BLECHARACTERISTIC::PROPERTY_NOTIFY);
TXCHARACTERISTIC->ADDDESRIPTOR(NEW BLE2902()); // ADICIONA O DESCRITOR 2902, ESSENCIAL PARA
NOTIFICAÇÕES.

```

```

// INICIA O SERVIÇO E O ADVERTISING (PARA QUE O DISPOSITIVO SEJA VISÍVEL).
PSERVICE->START();
BLEDEVICE::STARTADVERTISING();

```

```

SERIAL.PRINTLN("ESP32 BLE INICIADO E AGUARDANDO CONEXÃO!");

```

```

// REALIZA A CALIBRAÇÃO DO SENSOR ANTES DE INICIAR O LOOP.

```

```

CALIBRAROFFSET();

```

```

}

```

```

//-----

```

```

// 6. FUNÇÃO LOOP - LÓGICA PRINCIPAL DE EXECUÇÃO

```

```

//-----

```

```

VOID LOOP() {

```

```

// --- BLOCO 1: EXECUTADO APENAS QUANDO O MONITORAMENTO ESTÁ ATIVO ---

```

```

IF (ISREADING) {

```

```

    UNSIGNED LONG TEMPOATUAL = MILLIS();

```

```

// PROCESSAMENTO DE SINAL: LÊ, REMOVE OFFSET, CALCULA BASELINE E ISOLA O SINAL RESPIRATÓRIO.

```

```

    FLOAT DV = (RAW_VOLTAGE(MPS20N0040D.READ(), GANHO) - OFFSET) * 1000; // µV

```

```

    FLOAT BASELINE = CALCULARLINHABASILAR(DV);

```

```

    FLOAT Vf = DV - BASELINE; // SINAL FINAL, PRONTO PARA ANÁLISE E ENVIO.

```

```

// PREPARA A STRING NO FORMATO "TEMPO_EM_SEGUNDOS,TENSÃO_EM_µV" E ENVIA PARA O APP.

```

```

    STRING LEITURA = STRING((TEMPOATUAL - STARTATIVIDADE) / 1000.0) + "," + STRING(Vf, 2) + "\n";

```

```

    TXCHARACTERISTIC->SETVALUE(LEITURA.C_STR());

```

```

    TXCHARACTERISTIC->NOTIFY();

```

```

// --- MÁQUINA DE ESTADOS FINITOS PARA ANÁLISE DE CICLO ---

```

```

SWITCH (ESTADOATUAL) {

```

```

    CASE AGUARDANDO_CICLO:

```

```

        IF (Vf < -LIMIAR) { // SE A TENSÃO FICA NEGATIVA (INSPIRAÇÃO)...

```

```

            ESTADOATUAL = EM_INSPIRACAO;

```

```

    TEMPOINICIOINSPIRACAOTOTAL = TEMPOATUAL;
}
BREAK;

CASE EM_INSPIRACAO:
IF (VF > LIMAR) { // SE A TENSÃO FICA POSITIVA (EXPIRAÇÃO)...
    ESTADOATUAL = EM_EXPIRACAO;
    CONTADORCICLOSCOMPLETOS++;
    TEMPOTOTALINSPIRACAO += (TEMPOATUAL - TEMPOINICIOINSPIRACAOTOTAL);
    TEMPOINICIOEXPIRACAOTOTAL = TEMPOATUAL;
}
BREAK;

CASE EM_EXPIRACAO:
IF (VF <= 0) { // SE A TENSÃO RETORNA A ZERO (FIM DA EXPIRAÇÃO)...
    ESTADOATUAL = AGUARDANDO_CICLO;
    TEMPOTOTALEXPIRACAO += (TEMPOATUAL - TEMPOINICIOEXPIRACAOTOTAL);
}
BREAK;
}

DELAY(50); // CONTROLA A TAXA DE ENVIO DE DADOS.

// --- BLOCO 2: EXECUTADO APENAS QUANDO UM RELATÓRIO É SOLICITADO ---
} ELSE IF(REPORTREQUEST){
    FLOAT TOTALTEMPO = MILLIS() - STARTATIVIDADE;
    ENVIARRELATORIO(TOTALTEMPO);
    REPORTREQUEST = FALSE; // RESETA A FLAG PARA NÃO ENVIAR O RELATÓRIO NOVAMENTE.

// --- BLOCO 3: EXECUTADO QUANDO O SISTEMA ESTÁ INATIVO ---
} ELSE {
    DELAY(1000); // ECONOMIZA PROCESSAMENTO QUANDO O SISTEMA ESTÁ OCIOSO.
}
}

```