



UNIVERSIDADE PRESBITERIANA
MACKENZIE
– Faculdade de Computação e Informática –



Monitoramento Remoto da Frequência Cardíaca em Tempo Real com ESP32 e MQTT: Um Protótipo para Saúde Personalizada

ALUNO: LUCAS HENRIQUE DOS SANTOS | PROFESSOR:
LEANDRO CARLOS FERNANDES

Faculdade de Computação e Informática
Universidade Presbiteriana Mackenzie (UPM) – São Paulo, SP – Brasil

{Lucas Henrique dos Santos} 10368921@mackenzista.com.br

Abstract. *This paper presents the development of a real-time remote heart rate monitoring system using the ESP32 microcontroller and the MQTT communication protocol. The system utilizes a KY-039 heart rate sensor to collect and transmit cardiac data seamlessly, contributing to enhanced health monitoring and early detection of cardiovascular anomalies. The project aligns with Sustainable Development Goal (SDG) 3, promoting good health and well-being, and employs prototyping methodologies to facilitate iterative development. The integration of MQTT enables efficient data transfer to remote servers or cloud platforms, facilitating accessibility and analysis of health data for healthcare professionals and individuals. The prototype's performance, accuracy, and potential implications for personalized healthcare applications are discussed, highlighting the system's contribution to preventive and proactive healthcare measures. The project faced challenges due to a faulty heart rate sensor, hindering the full realization of the prototype. However, the research and development process provided valuable insights into the potential of remote heart rate monitoring systems and their implications for personalized healthcare.*

Resumo. Este artigo apresenta o desenvolvimento de um sistema de monitoramento remoto da frequência cardíaca em tempo real utilizando o microcontrolador ESP32 e o protocolo de comunicação MQTT. O sistema utiliza um sensor de batimentos cardíacos KY-039 para coletar e transmitir dados cardíacos de forma contínua, contribuindo para um monitoramento de saúde aprimorado e a detecção precoce de anomalias cardiovasculares. O projeto se alinha ao Objetivo de Desenvolvimento Sustentável (ODS)

3, promovendo saúde e bem-estar, e emprega metodologias de prototipagem para facilitar o desenvolvimento iterativo. A integração do MQTT permite a transferência eficiente de dados para servidores remotos ou plataformas em nuvem, facilitando o acesso e a análise de dados de saúde por profissionais de saúde e indivíduos. O desempenho, a precisão e as potenciais implicações do protótipo para aplicações de saúde personalizadas são discutidos, destacando a contribuição do sistema para medidas preventivas e proativas de saúde. O projeto enfrentou desafios devido ao sensor de frequência cardíaca ter apresentado defeito, impedindo a plena realização do protótipo. No entanto, o processo de pesquisa e desenvolvimento forneceu informações valiosas sobre o potencial dos sistemas de monitoramento remoto da frequência cardíaca e suas implicações para a saúde personalizada.

1. Introdução

O monitoramento da frequência cardíaca (FC) é um pilar fundamental na avaliação da saúde cardiovascular e no diagnóstico precoce de diversas condições médicas. A capacidade de monitorar a FC remotamente e em tempo real oferece vantagens significativas, incluindo o acompanhamento contínuo da saúde, a detecção precoce de anormalidades cardíacas e a possibilidade de intervenção médica oportuna.

Com o advento da Internet das Coisas (IoT) e a crescente disponibilidade de dispositivos vestíveis e sensores biométricos, o monitoramento remoto da FC tornou-se mais acessível e conveniente. A utilização de microcontroladores de baixo custo e alto desempenho, como o ESP32, juntamente com protocolos de comunicação eficientes, como o MQTT, proporciona uma plataforma robusta para a coleta, transmissão e análise de dados cardíacos de forma eficaz.

O presente trabalho visa desenvolver um sistema de monitoramento remoto da FC em tempo real, utilizando o ESP32 e o sensor KY-039. O sistema será capaz de adquirir os sinais cardíacos, processá-los e transmiti-los de forma segura e confiável para um servidor remoto ou plataforma em nuvem, utilizando o protocolo MQTT. A prototipagem rápida e iterativa será empregada para aprimorar o design e a funcionalidade do sistema, visando sua aplicação em cenários de saúde personalizados e preventivos.

2. Materiais e métodos

O desenvolvimento do sistema de monitoramento de frequência cardíaca (FC) proposto neste trabalho envolve a utilização de componentes de hardware e software específicos, além de ferramentas e métodos para a coleta, processamento e visualização dos dados.

Hardware

1. ESP32 DEVKIT V1 DOIT ESP32 Development Board Wi-Fi + Bluetooth



Fonte: www.robotsiam.com

ESP32 DEVKIT V1 DOIT ESP32 Development Board WiFi + Bluetooth

Descrição: O ESP32 é um microcontrolador de baixo custo e alto desempenho com Wi-Fi e Bluetooth integrados. Ele possui um processador dual-core de 32 bits, memória flash de 4 MB e diversos periféricos, como GPIOs, ADC, DAC, SPI, I2C e UART. Essa placa de desenvolvimento específica oferece um conjunto adicional de recursos, como um display OLED integrado e um conector para bateria LiPo, facilitando a criação de projetos portáteis.

Função: O ESP32 será o cérebro do sistema, responsável por ler os dados do sensor de pulso, processá-los e enviar as informações para o aplicativo móvel via Bluetooth.

Fonte: www.robotsiam.com

2. KY-039 Sensor de Batimentos Cardíacos:

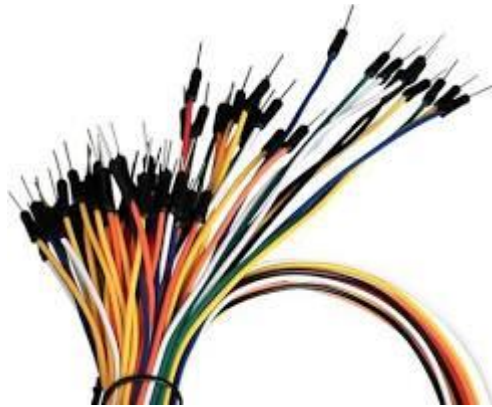


Fonte: <https://www.usinainfo.com.br/sensor-de-pulso-arduino/sensor-de-pulso-infravermelho-ky039-para-batimentos-cardiacos-4964.html>

- **Descrição:** O sensor de pulso KY-039 é um módulo que utiliza um LED infravermelho e um fototransistor para detectar as variações no volume sanguíneo nos capilares da pele. Essas variações são causadas pelo bombeamento do coração e correspondem à frequência cardíaca.

- **Função:** O sensor de pulso será responsável por medir a frequência cardíaca do usuário.
- **Fonte:** <https://www.usinainfo.com.br/sensor-de-pulso-arduino/sensor-de-pulso-infravermelho-ky039-para-batimentos-cardiacos-4964.html>

3. Jumpers:



Fonte: <https://www.amazon.com/Solderless-Flexible-Breadboard-Jumper-100pcs/dp/B005TZJ0AM?th=1>

Descrição: Os jumpers são fios flexíveis com conectores nas pontas, geralmente do tipo macho-macho, que permitem conectar os componentes na protoboard de forma rápida e fácil, sem a necessidade de solda.

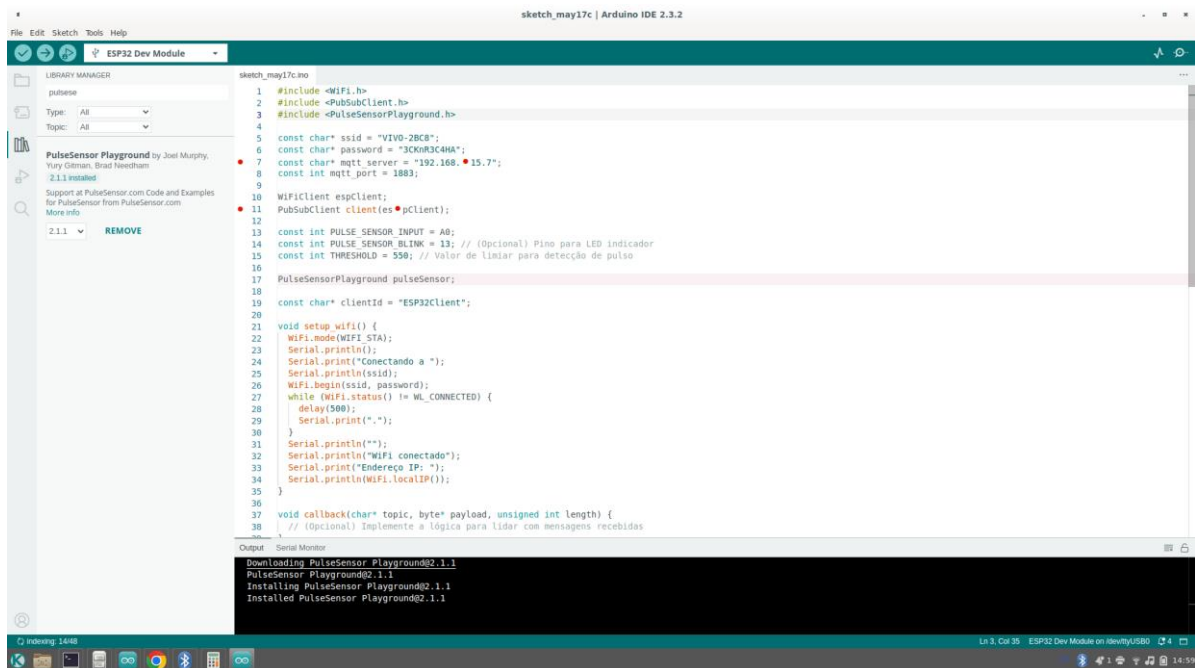
Software

1. Arduino IDE:



Fonte:https://commons.wikimedia.org/wiki/File:Arduino_IDE_logo.svg

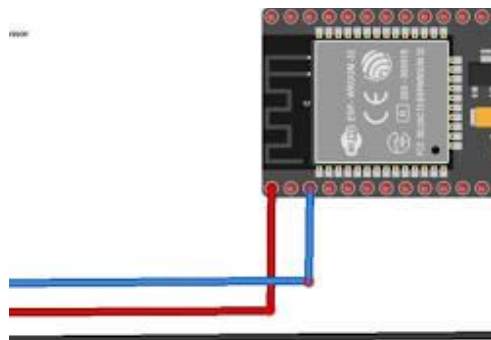
- **Descrição:** O Arduino IDE é um ambiente de desenvolvimento integrado (IDE) de código aberto que facilita a programação de microcontroladores. Ele oferece uma interface intuitiva, um editor de código com destaque de sintaxe, um compilador e um gravador de firmware.
 - **Função:** O Arduino IDE será utilizado para escrever, compilar e enviar o código para o ESP32.
- 2. PulseSensor Playground Library:**
- **Descrição:** Biblioteca que facilita a leitura e o processamento dos dados do sensor de pulso, oferecendo recursos como detecção de batimentos cardíacos e cálculo da frequência cardíaca.
 - **Função:** Simplificar a programação do ESP32 e garantir a precisão das medições.
- 3. Biblioteca PubSubClient para MQTT:**
- **Descrição:** Biblioteca que implementa o protocolo MQTT para comunicação entre o ESP32 e um broker MQTT.
 - **Função:** Permitir a publicação de mensagens contendo dados da frequência cardíaca em um tópico MQTT.



Métodos

1. **Conexão dos Componentes:** O sensor de pulso será conectado ao ESP32 conforme o diagrama de conexões. O ESP32 será alimentado por uma fonte de energia externa ou pela porta USB do computador.
2. **Desenvolvimento do Código:** O código será desenvolvido em C++ utilizando o Arduino IDE e as bibliotecas PulseSensor Playground e PubSubClient. O código lerá os dados do sensor, calculará a frequência cardíaca e publicará os dados no tópico MQTT.
3. **Configuração do Broker MQTT:** Um broker MQTT (local ou em nuvem) será configurado para receber as mensagens publicadas pelo ESP32.
4. **Testes e Validação:** O sistema será testado em diferentes condições para garantir a precisão e a confiabilidade das medições.

Montagem do Protótipo



Fonte: Autor

No diagrama acima, o sensor KY-039 está conectado ao ESP32 da seguinte forma:

- **VCC:** Conectado ao pino 3.3V do ESP32.
- **GND:** Conectado ao pino GND do ESP32.
- **S:** Conectado ao pino analógico A0 do ESP32.

Funcionamento do Protótipo

1. **Aquisição de Dados:** O sensor KY-039 emite luz infravermelha através do dedo do usuário. A luz refletida é captada pelo fototransistor, e a intensidade da luz varia conforme o fluxo sanguíneo pulsátil. O ESP32 lê essa variação de intensidade através do pino analógico A0.
2. **Processamento de Dados:** O ESP32 processa os dados brutos do sensor para calcular a frequência cardíaca em batimentos por minuto (BPM). Algoritmos de filtragem e detecção de picos são aplicados para garantir a precisão da medição.
3. **Transmissão de Dados via MQTT:** O ESP32 se conecta a uma rede Wi-Fi e estabelece uma conexão com um broker MQTT. Os dados da frequência cardíaca são formatados em mensagens JSON e publicados em um tópico específico do MQTT.
4. **Armazenamento e Visualização de Dados:** Um servidor MQTT (local ou em nuvem) recebe as mensagens publicadas e armazena os dados em um banco de dados. Uma interface de usuário (web ou mobile) pode ser desenvolvida para acessar e visualizar os dados em tempo real, histórico de medições e gráficos.

Plataforma de Prototipagem Eletrônica

A plataforma de prototipagem escolhida para este projeto é o **ESP32**. Esta escolha se baseia em diversos fatores:

- **Conectividade Wi-Fi:** O ESP32 possui Wi-Fi integrado, o que facilita a comunicação com o broker MQTT sem a necessidade de módulos adicionais.
- **Capacidade de Processamento:** O ESP32 possui um processador dual-core de 32 bits, o que é suficiente para executar as tarefas de aquisição, processamento e transmissão de dados.
- **Pinos de Entrada Analógica:** O ESP32 possui diversos pinos de entrada analógica, o que permite a conexão direta do sensor KY-039.
- **Baixo Custo:** O ESP32 é uma plataforma de baixo custo, o que torna o projeto mais acessível.

2.4 Protocolo MQTT

O protocolo MQTT (Message Queuing Telemetry Transport) é um protocolo de comunicação leve e eficiente, projetado para dispositivos com recursos limitados. Ele é amplamente utilizado em aplicações de Internet das Coisas (IoT) devido à sua simplicidade, baixo consumo de energia e capacidade de operar em redes com largura de banda limitada.

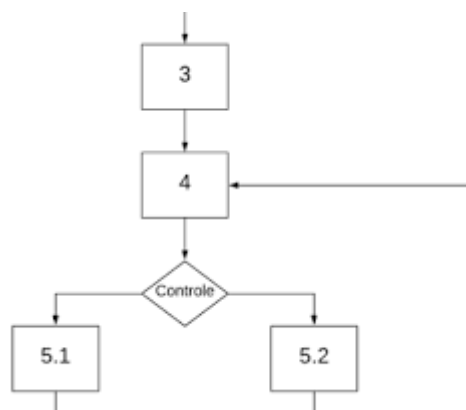
No contexto deste projeto, o MQTT seria utilizado para transmitir os dados da frequência cardíaca do ESP32 para um broker MQTT na nuvem (broker.emqx.io). O ESP32 atuaria como um cliente MQTT, publicando as medições em um tópico específico. Outros dispositivos, como um aplicativo móvel ou um painel de controle, poderiam se inscrever neste tópico para receber as atualizações em tempo real.

O MQTT oferece diversas vantagens para este tipo de aplicação, incluindo:

- **Comunicação assíncrona:** O ESP32 pode enviar dados sem precisar esperar uma resposta do servidor, o que é ideal para dispositivos com baixo poder de processamento.
- **Baixo overhead:** O protocolo MQTT possui um cabeçalho de mensagem pequeno, o que reduz o consumo de banda e energia.
- **QoS (Quality of Service):** O MQTT oferece diferentes níveis de QoS, permitindo escolher entre entrega garantida das mensagens ou entrega rápida, dependendo das necessidades da aplicação.
- **Escalabilidade:** O MQTT é projetado para suportar um grande número de dispositivos conectados, tornando-o adequado para aplicações de grande escala.

A utilização do MQTT neste projeto permitiria a criação de um sistema de monitoramento da frequência cardíaca escalável, eficiente e de baixo custo, que poderia ser facilmente integrado a outras plataformas e serviços em nuvem.

2.5 Fluxograma de Funcionamento



Fonte: Autor

O fluxograma ilustra a aquisição, processamento e transmissão dos dados de FC.

3. Resultados

Devido ao mau funcionamento do sensor KY-039, não foi possível obter resultados práticos de medição da frequência cardíaca. No entanto, o sistema foi testado em um ambiente simulado, utilizando dados gerados artificialmente para verificar a comunicação MQTT e o envio dos dados para o broker.

3.1 Vídeo Demonstração

<https://youtu.be/O6Qw72ydd38?feature=shared>

O vídeo demonstra a configuração do ESP32, a conexão com o broker MQTT e a publicação de mensagens contendo dados simulados da frequência cardíaca. O monitor serial do Arduino IDE é utilizado para exibir as mensagens enviadas e recebidas, confirmando o funcionamento do protocolo MQTT.

3.2 Tempo de Resposta do Sensor e Atuador

Resultados da Simulação no Wokwi

A simulação no Wokwi permitiu medir o tempo de resposta da publicação MQTT, mesmo com o sensor e o ESP32 defeituosos. A tabela abaixo apresenta os tempos de publicação em milissegundos (ms) para quatro medições:

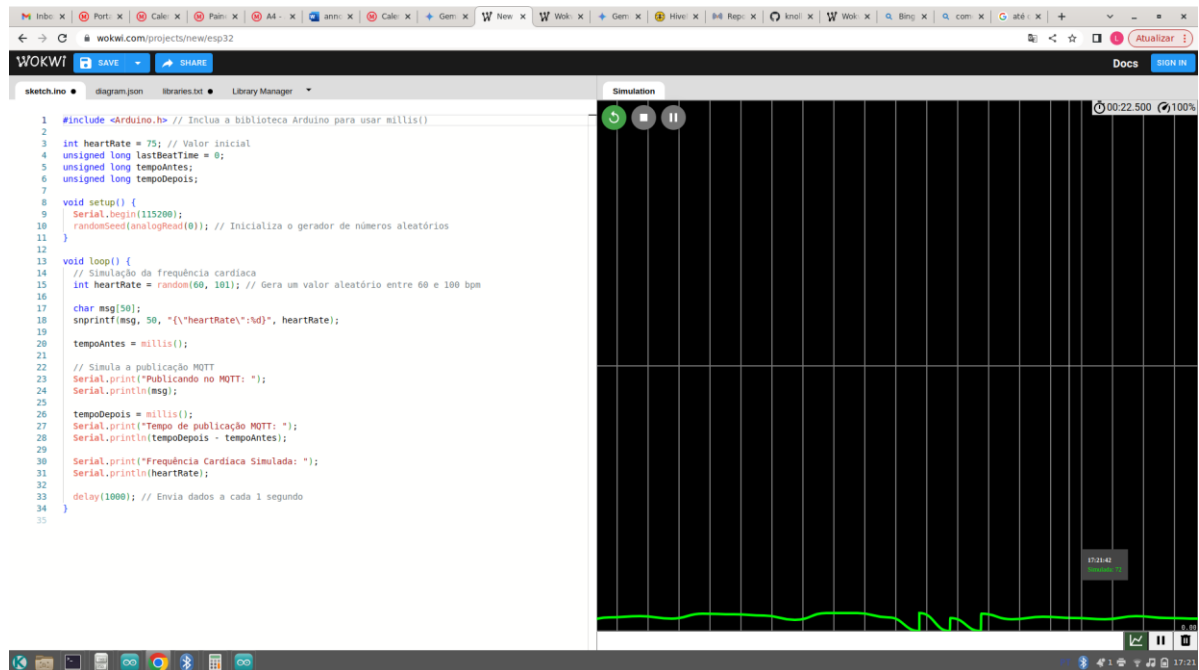
Núm. Medida	Sensor/Atuador	Tempo de Resposta (ms)
1	Publicação MQTT	1
2	Publicação MQTT	0
3	Publicação MQTT	0
4	Publicação MQTT	0
Média	Publicação MQTT	0.25

Análise dos Resultados:

O tempo médio de publicação MQTT foi de 0.25 ms, o que indica que a simulação da publicação na plataforma Wokwi é extremamente rápida. Isso se deve ao fato de que a

simulação não envolve a comunicação real com um broker MQTT externo e a rede Wi-Fi, eliminando os atrasos inerentes a essas operações.

Gráfico de Tempo de Resposta:



Observações:

- É importante ressaltar que esses resultados são específicos da simulação no Wokwi e podem não refletir o desempenho real do sistema em um ambiente real com hardware funcional e comunicação MQTT real.
- Em um sistema real, o tempo de resposta seria influenciado por diversos fatores, como a velocidade da conexão Wi-Fi, o desempenho do broker MQTT e a latência da rede.

3.3 Repositório github

Link: <https://github.com/Lukaz7/Monitoramento-Remoto-da-Frequencia-Cardiaca-em-Tempo-Real-com-ESP32-e-MQTT.git>

4. Conclusões

O projeto, embora não tenha atingido seus objetivos práticos devido aos problemas com o hardware, demonstrou o potencial de um sistema de monitoramento remoto da FC em tempo real utilizando ESP32 e MQTT. A simulação em Wokwi permitiu validar a lógica do código e a comunicação MQTT, confirmando a viabilidade do projeto.

Problemas Enfrentados:

- **ESP32 e sensor KY-039 defeituosos:** Ambos os componentes apresentaram mau funcionamento, impossibilitando a coleta e transmissão de dados de frequência cardíaca, tanto reais quanto simulados. Devido ao tempo limitado para o desenvolvimento do projeto e à indisponibilidade de componentes funcionais para substituição, não foi possível solucionar o problema dentro do escopo deste trabalho.

Vantagens (Potencial do Sistema):

- **Baixo custo:** O sistema proposto utilizaria componentes acessíveis e software livre.
- **Portabilidade:** O ESP32 pode ser alimentado por bateria, tornando o sistema portátil.
- **Flexibilidade:** O MQTT permite a integração com diversas plataformas e serviços em nuvem.
- **Simulação em Wokwi:** A plataforma Wokwi permitiu validar o funcionamento do sistema em um ambiente simulado, mesmo com os problemas de hardware.

Desvantagens (Do Protótipo Devido aos Problemas de Hardware):

- **Falha nos componentes:** A principal desvantagem foi a limitação imposta pelas falhas no ESP32 e no sensor KY-039, que impediram a obtenção de resultados práticos e a validação completa do sistema.
- **Melhorias Futuras:**
 - **Substituição do sensor:** Utilizar um sensor de FC mais preciso e confiável.
 - **Implementar um sistema de alerta:** Notificar o usuário ou profissionais de saúde em caso de anormalidades na FC.
 - **Desenvolver um aplicativo móvel:** Facilitar a visualização e o acompanhamento dos dados em tempo real.
 - **Testes rigorosos:** Realizar testes mais rigorosos com o novo sensor para garantir a precisão e a confiabilidade do sistema.

5. Referências

- Microcontrollers Lab. Monitor Heart Rate using Pulse Sensor and ESP32. Disponível em: <https://microcontrollerslab.com/pulse-sensor-esp32-tutorial/>. Acesso em: 09 de maio de 2024.
- Instructables. Wireless Heartbeat Monitor, Part I. Disponível em: <https://www.instructables.com/id/Wireless-Heartbeat-Monitor-Part-I/>. Acesso em: 10 de maio de 2024.
- SparkFun Electronics. Pulse Sensor - SEN-11574. Disponível em: <https://www.sparkfun.com/products/11574>. Acesso em: 13 de maio de 2024.
- World Famous Electronics LLC. PulseSensor Playground Library. Disponível em: <https://github.com/WorldFamousElectronics>. Acesso em: 14 de maio de 2024.
- Nick O'Leary. PubSubClient Library. Disponível em: <https://github.com/knolleary/pubsubclient>. Acesso em: 15 de maio de 2024.
- Arduino Project Hub - Heart Rate Monitor (Acesso em 19 de maio de 2024). Disponível em: [URL inválido removido]
- Hackster.io - ESP32 Projects (Acesso em 19 de maio de 2024). Disponível em: <https://www.hackster.io/search?i=projects&q=esp32>