



UM PROTÓTIPO VESTÍVEL PARA ACOMPANHAMENTO DE SAÚDE

Erick Sanches Klein

Trabalho de Conclusão Apresentado ao Curso de Graduação em Sistemas de Informação do Centro de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ, como parte dos requisitos necessários à obtenção do título de Bacharel.

Orientador: Nilson Mori Lazarin

Nova Friburgo

Fevereiro 2024

UM PROTÓTIPO VESTÍVEL PARA ACOMPANHAMENTO DE SAÚDE

Trabalho de Conclusão Apresentado ao Curso de Graduação em Sistemas de Informação do Centro de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ, como parte dos requisitos necessários à obtenção do título de Bacharel.

Erick Sanches Klein

Banca Examinadora:



Documento assinado digitalmente

Nilson Mori Lazarin

Data: 15/02/2024 16:42:21-0300

Verifique em <https://validar.iti.gov.br>

Presidente, Prof. Me. Nilson Mori Lazarin (CEFET/RJ) (Orientador)

**PAULO HENRIQUE WERLY
GUALBERTO:01912873770**

Assinado de forma digital por PAULO HENRIQUE
WERLY GUALBERTO:01912873770
Dados: 2024.02.15 16:33:05 -03'00'

Prof. Me. Paulo Henrique Verly Gualberto (CEFET/RJ)



Documento assinado digitalmente

BRUNO FERNANDES GUEDES

Data: 12/02/2024 10:07:10-0300

Verifique em <https://validar.iti.gov.br>

Prof. Me. Bruno Fernandes Guedes (CEFET/RJ)

Nova Friburgo

Fevereiro 2024

CEFET/RJ – Sistema de Bibliotecas / Biblioteca Uned Nova Friburgo

K64p Klein, Erick Sanches
Um protótipo vestível para acompanhamento de saúde. /
Erick Sanches Klein. – Nova Friburgo, RJ: 2024.
vii, 41f.: il. (color.) : em PDF.

Trabalho de Conclusão de Curso (Curso de Bacharelado em
Sistemas de Informação) - Centro Federal de Educação
Tecnológica Celso Suckow da Fonseca, 2024.

Bibliografia : f. 38-41

Orientador: Nilson Mori Lazzarin

1. Sistemas de informação. 2. Vestuário – Inovações
tecnológicas 3. Tecnologia médica I. Lazzarin, Nilson Mori
(Orientador). II. Título.

CDD 005.1

DEDICATÓRIA

Aos meus mais preciosos pais, Marco e Simoni, e ao meu irmão, Caio que sempre me incentivam em meus projetos, colaborando de todas as formas para que possa realizar meus objetivos.

AGRADECIMENTOS

Agradeço, primeiramente, ao Deus Eterno que me amparou e me fortaleceu nos momentos mais angustiantes, aos meus pais e ao meu irmão, colaboradores e incentivadores, e, por fim, ao meu orientador, Nilson Mozi Larazin, que me aconselhou e orientou de maneira tal que este veio a ser concluído.

RESUMO

Um protótipo vestível para acompanhamento de saúde

Este trabalho apresenta o desenvolvimento e implementação de um protótipo *wearable* de monitoramento de saúde concebido para idosos em instituições de cuidados. O sistema proposto integra sensores não intrusivos e minimamente invasivos, incluindo frequência cardíaca (Amped 1.5), temperatura corporal (LM35) e acelerômetro (ADXL345), em um protótipo vestível integrado a uma camisa. Os dados coletados por esses sensores são processados pelo Arduino Mega 2560 e transmitidas sem fio para um servidor remoto por meio do transmissor RF de 433 MHz (FS1000A). O sistema foi projetado para oferecer monitoramento em tempo real para vários indivíduos simultaneamente, visando aumentar a eficiência no cuidado. A configuração do hardware inclui ainda o uso de uma bateria recarregável (Power bank) para garantir autonomia ao dispositivo. Os dados enviados pelo transmissor RF de 433 MHz (RWS-371-6) são recebidos por outro Arduino Mega 2560 conectado a uma Ethernet Shield para armazená-los temporariamente. Com o intuito de automatizar a recuperação desses dados do servidor web, o armazenamento local e a visualização gráfica, desenvolvemos um script Python. Este script foi implementado no dispositivo local do respectivo cuidador, proporcionando uma visão abrangente dos parâmetros de saúde do indivíduo monitorado. A funcionalidade do sistema foi validada por meio de um estudo de caso que envolveu a prova de conceito do monitoramento de um indivíduo em diversos cenários, incluindo simulações de quedas, para verificar sua viabilidade e desempenho. O protótipo proposto apresenta uma solução viável para monitoramento remoto da saúde, oferecendo aos cuidadores acesso seguro e eficiente a dados em tempo real, facilitando respostas oportunas a emergências. Esta pesquisa contribui para o avanço dos sistemas vestíveis de monitoramento de saúde, atendendo à crescente necessidade de soluções confiáveis e discretas em instalações de atendimento a idosos.

Palavras-chave: *Wearable*; Monitoramento de Saúde; Idosos.

ABSTRACT

A WEARABLE PROTOTYPE FOR HEALTH MONITORING

This work presents the development and implementation of a wearable health monitoring prototype designed for elderly people in care institutions. The proposed system integrates non-intrusive and minimally invasive sensors, including heart rate (Amped 1.5), body temperature (LM35) and accelerometer (ADXL345), into a wearable prototype integrated into a shirt. The data collected by these sensors is processed by the Arduino Mega 2560 and transmitted wirelessly to a remote server via the 433 MHz RF transmitter (FS1000A). The system was designed to offer real-time monitoring for multiple individuals simultaneously, aiming to increase care efficiency. The hardware configuration also includes the use of a rechargeable battery (Power bank) to guarantee autonomy for the device. The data sent by the 433 MHz RF transmitter (RWS-371-6) is received by another Arduino Mega 2560 connected to an Ethernet Shield to temporarily store it. In order to automate the retrieval of this data from the web server, local storage and graphical visualization, we developed a Python script. This script was implemented on the respective caregiver's local device, providing a comprehensive view of the monitored individual's health parameters. The system's functionality was validated through a case study that involved the proof of concept of monitoring an individual in various scenarios, including simulated falls, to verify its feasibility and performance. The proposed prototype presents a viable solution for remote health monitoring, offering caregivers safe and efficient access to real-time data, facilitating timely responses to emergencies. This research contributes to the advancement of wearable health monitoring systems, meeting the growing need for reliable and discreet solutions in senior care facilities.

Keywords: Wearable; Health Monitoring; Elderly.

SUMÁRIO

1	Introdução	5
1.1	Definição do Problema	6
1.2	Objetivo	7
2	Fundamentação Teórica	8
2.1	Arduino Mega	8
2.2	Shield Ethernet	8
2.3	Sensor de Batimentos Cardíacos Amped 1.5	9
2.4	Sensor de Temperatura LM35	10
2.5	Acelerômetro ADXL345	11
2.6	Módulo Real Time Clock DS3231	11
2.7	Transmissor FS1000A	12
2.8	Receptor RWS-371-6	13
2.9	Power Bank Maxprint	14
2.10	Arduino IDE	14
2.11	SPI	15
2.12	Wire	15
2.13	SparkFun ADXL345	16
2.14	SecurityVanet	16
2.15	Ethernet	16
2.16	ArduinoJson	17
2.17	RTClib	17
2.18	Spyder	18
3	Trabalhos Relacionados	19
3.1	A Novel Approach for IoT Based Wearable Health Monitoring and Messaging System	19

3.2	Healthcare Monitoring System and transforming Monitored data into Real time Clinical Feedback based on IoT using Raspberry Pi	20
3.3	Wearable Sensors for Remote Health Monitoring	21
4	Proposta	22
4.1	Implementação	23
4.1.1	HARDWARE	23
4.1.2	SOFTWARE	24
5	Estudo de Caso	33
6	Conclusão	36
	Referências	37

1- Introdução

A expectativa de vida ao redor do mundo vem crescendo rapidamente graças aos importantes avanços da tecnologia, principalmente, na área da medicina corretiva e preventiva. Conforme a Organização Mundial da Saúde (OMS), o número de indivíduos com mais de 60 anos chegará a 2 bilhões de pessoas até 2050, representando um quinto da população mundial. Somado também com um aumento considerável no planejamento familiar, que contribuiu para o declínio da taxa de natalidade, estima-se que o número de idosos ultrapassará o total de crianças entre zero e 14 anos em 2030 (ASAP, 2021).

Com o acréscimo do número de idosos, há também um constante aumento no número da procura por asilos que servem, justamente, para abrigar e cuidar dessas pessoas. Conforme o Instituto Brasileiro de Geografia e Estatística (IBGE) a procura por asilos públicos cresceu 33% entre 2012 e 2017, saltando de 45,8 mil para 60,8 mil, principalmente devido ao abandono de idosos, ficando evidente a necessidade de se ampliar a forma com que essa parcela da população é tratada (CAVICCHIOLI; VILARDAGA, 2021).

Cuidar de idosos em asilos não é uma tarefa simples, uma vez que não há a presença de muitos cuidadores, incorrendo na falta de atenção adequada a cada indivíduo. Há também muitas barreiras que dificultam o tratamento destes, tais como a demora de meses para a realização de exames, além da dificuldade de deslocamento para realizá-los (SILVA et al., 2021).

Além disso, é muito comum alterações significativas da frequência cardíaca e da termorregulação corporal, aumentando principalmente o risco de doenças cardiovasculares (CTCOR, 2024; CARE, 2019). Combinado com outras doenças que ocorrem naturalmente, há uma diminuição contínua das habilidades físicas e cognitivas dos idosos, aumentando o risco de quedas (G. STEFANACCI, 2022). De acordo com (MONITORAMENTO, 2021; OLIVEIRA; PAULO; MONTE MOR FILHO, 2018; CUNHA; LOURENÇO, 2014), cerca de 40% das pessoas de terceira idade já sofreram ao menos uma queda ao longo do ano, além delas serem responsáveis por 70% dos casos de óbito desta faixa etária (LOPES et al., 2019; NETO et al., 2017), tornando necessário o acompanhamento contínuo destes.

Utilizando o conceito de *Internet of Healthcare Things* (IoHT), é possível monitorar remotamente esses indivíduos por sensores minimamente invasivos e vestíveis totalmente conectados e sempre disponíveis para uso, os quais fornecem continuamente e em tempo real, diversos dados físicos das pessoas que os utilizam, como temperatura, batimentos cardíacos, queda, dentre outros. Esses dados, no que lhe concerne, podem ser compartilhados na rede para compreender o ambiente, melhorar a experiência humana e gerar qualidade de vida (R. VIEIRA; PONTE JUNIOR, 2018).

Desta forma, é possível criar um ambiente inteligente que proporciona ao idoso uma vida independente por um longo tempo, evitando visitas frequentes ao hospital, reduzindo os custos da saúde, contribuindo para uma menor pressão do sistema de saúde geral e, principalmente, auxiliando o funcionamento dos asilos, tornando muito mais fácil o monitoramento e a eficácia no atendimento aos idosos, especialmente, em casos de emergência (R. VIEIRA; PONTE JUNIOR, 2018).

1.1- Definição do Problema

Cuidar de idosos, na maioria das vezes, não é uma tarefa simples, pois além de zelar pela saúde e integridade do paciente, também se faz necessário estar atento ao seu bem-estar emocional e de sua vida social. Por esse motivo, é comum os parentes não conseguirem dar a assistência ideal ao familiar, preferindo deixá-lo aos cuidados de quem possa exercer essa função de maneira mais eficiente. Vale ressaltar que há também muitos casos de abandono de idosos por parte dos familiares ou até mesmo muitos que não possuem mais família, gerando um acréscimo de apoio indispensável a essas pessoas (BENTO; AMARAL; SILVA, 2021).

Para isso, foram criados lares para acolher e abrigar essas pessoas, buscando proporcionar melhor qualidade de vida na sua velhice. Esses lares, também chamados de asilos, por diversas vezes, são filantrópicos e possuem recursos escassos, dificultando a continuidade dos cuidados aos idosos (FAGUNDES et al., 2017).

Embora os idosos possam não adquirir as mesmas patologias em geral, muitas delas apresentam semelhanças, especialmente as conhecidas como doenças relacionadas à velhice, como hipertensão arterial e cardiopatias, sem falar que, nessa idade, as

quedas são mais frequentes (DA SAÚDE, 2019).

Enfim, tudo isso tem se tornado um dificultador no cuidado dos idosos, tornando imprescindível a criação de uma solução que contribua para diminuir a quantidade de funcionários necessários, diminuindo consequentemente os custos para tal, bem como um aumento da qualidade do atendimento aos idosos, tornando o atendimento mais rápido e contínuo (OLVEIRA; VERAS; CORDEIRO, 2019).

1.2- Objetivo

O objetivo deste trabalho é desenvolver um protótipo vestível para acompanhar e monitorar idosos em asilos, que envie os dados coletados para os respectivos cuidadores. Esse protótipo utilizará sensores minimamente invasivos e vestíveis, alimentados por uma bateria externa recarregável, podendo ser retirada a qualquer momento, para o monitoramento e coleta, contínua e em tempo real, de dados a respeito de uma possível queda, bem como os batimentos cardíacos e a temperatura do indivíduo.

Esses dados serão revestidos por uma camada de segurança antes de encaminhá-los, via módulo RF, ao receptor local. Uma vez no receptor, essas informações serão enviadas para um servidor local para serem capturadas por um aplicativo de análise de dados. O aplicativo, por sua vez, armazenará esses dados no banco de dados local e cuidará de exibí-los para os respectivos cuidadores que poderão aferir como está a saúde dos idosos.

2- Fundamentação Teórica

2.1- Arduino Mega

O Arduino Mega (Figura 1) é uma placa de desenvolvimento eletrônico baseada no microcontrolador ATmega2560. É uma versão expandida do Arduino Uno, oferecendo mais pinos de I/O, memória Flash, RAM e recursos. Possui 54 pinos de I/O, 256 KB de memória Flash, 8 KB de memória RAM, 4 KB de memória EEPROM e suporte a várias formas de comunicação, como UART, I2C e SPI. Pode ser alimentado por USB ou fonte externa, sendo compatível com shields do Arduino Uno (TEAM, 2023c). O Arduino Mega foi escolhido para o projeto devido a ser o primeiro dispositivo com capacidade suficiente de memória para atender às demandas específicas do sistema. Antes dessa escolha, foram testados o Lilypad e o Arduino Uno, mas ambos foram descartados devido à falta de memória para suportar todas as funcionalidades necessárias.

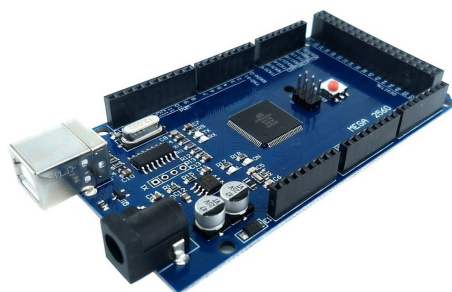


Figura 1 – Arduino Mega.

2.2- Shield Ethernet

Uma Shield Ethernet (Figura 2) é uma placa de expansão para placas Arduino que adiciona conectividade de rede Ethernet aos projetos. Equipada com um controlador

Ethernet, ela permite que o Arduino se comunique com redes locais (LAN) e acesse a Internet. Possui um conector RJ45 para a conexão física, LEDs indicadores e requer configuração de endereço IP e MAC. Utilizada em projetos que envolvem comunicação remota, controle online e monitoramento de sensores pela web, as Shields Ethernet são compatíveis com diferentes modelos de placas Arduino e simplificam o desenvolvimento de software por meio de bibliotecas e APIs específicas (TEAM, 2023a).

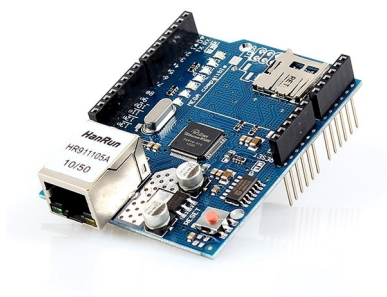


Figura 2 – Shield Ethernet.

2.3- Sensor de Batimentos Cardíacos Amped 1.5

O sensor de batimentos cardíacos Amped 1.5 (Figura 3), também conhecido como monitor de frequência cardíaca, é um dispositivo que mede os batimentos do coração por minuto. Este sensor utiliza a tecnologia de fotopletismografia, envolvendo LEDs e um fotodetector na pele e proporciona monitoramento em tempo real, sendo frequentemente incorporado a dispositivos vestíveis, como smartwatches e pulseiras fitness. Além do uso em atividades físicas, esse sensor tem aplicações médicas, sendo utilizado para monitoramento hospitalar e acompanhamento de condições cardíacas específicas. Em resumo, é um dispositivo crucial para o acompanhamento da saúde cardíaca e promoção de um estilo de vida saudável (MOTA MARTINS; ROCHA MATOS, 2012).



Figura 3 – Sensor de batimentos cardíacos Amped 1.5.

2.4- Sensor de Temperatura LM35

O LM35 (Figura 4) é um sensor de temperatura analógico amplamente utilizado em projetos eletrônicos. Ele gera uma saída de tensão proporcional à temperatura medida, com precisão e uma ampla faixa de medição de -55°C a 150°C . Alimentado por uma faixa de 4 a 30 volts, possui calibração interna e é comumente encontrado no formato TO-92 ("Transistor Outline - 92": tipo específico de encapsulamento de três pinos que são usados para conexão elétrica). Este sensor é popular em aplicações como automação residencial, controle de temperatura eletrônica e termostatos, devido à sua facilidade de uso e precisão. Para integrá-lo a sistemas digitais, pode ser necessário um conversor analógico-digital (ADC) (INSTRUMENTS INCORPORATED, 2017).

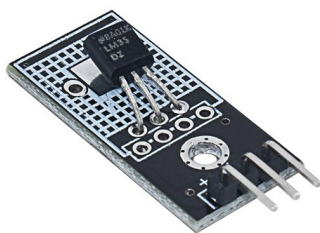


Figura 4 – Sensor de temperatura LM35.

2.5- Acelerômetro ADXL345

O ADXL345 (Figura 5) é um acelerômetro digital de três eixos fabricado pela Analog Devices, utilizando tecnologia MEMS ("Microeletromecânica": permite o desenvolvimento de produtos inteligentes, normalmente em escala microscópica). Ele mede aceleração nos eixos x, y e z, possui comunicação digital (I2C ou SPI), faixa de medição ajustável, resolução configurável e baixo consumo de energia. Amplamente utilizado em dispositivos móveis, drones, robótica e jogos, o ADXL345 é conhecido por sua versatilidade, sensibilidade e formato compacto, sendo uma escolha comum para projetos que exigem detecção e medição de movimento em três dimensões (DEVICES, 2022).

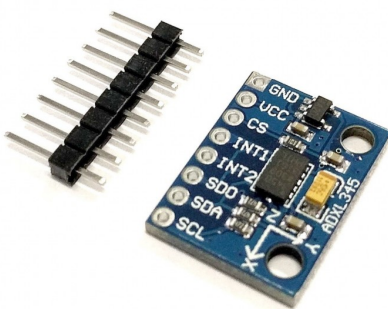


Figura 5 – Acelerômetro ADXL345.

2.6- Módulo Real Time Clock DS3231

O DS3231 (Figura 6) é um módulo *Real Time Clock* (RTC) conhecido por sua alta precisão e estabilidade no fornecimento de informações precisas sobre o tempo em tempo real. Utilizando um cristal de quartzo, oferece contagem de tempo estável, além de incluir calendário completo com suporte a anos bissextos. Com comunicação I2C, permite a configuração de alarmes programáveis e possui uma bateria de backup para

manter a hora mesmo em caso de falta de energia. Com baixo consumo de energia e formato compacto, é amplamente utilizado em diversas aplicações, como sistemas de registro de dados, controle de acesso e automação residencial (INTEGRATED, 2015).

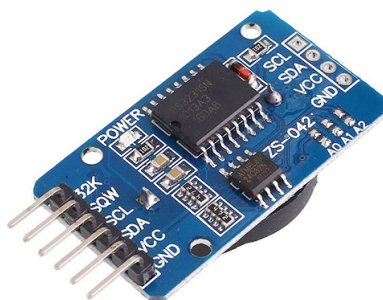


Figura 6 – Módulo Real Time Clock DS3231.

2.7- Transmissor FS1000A

O Transmissor FS1000A (Figura 7) é um módulo de comunicação sem fio que opera na frequência de 433MHz. Com um alcance variando de 20 a 200 metros, uma faixa de tensão de operação de 3,5 a 12V e uma potência de transmissão de 10mW, esse transmissor utiliza modulação ASK (Amplitude Shift Keying) e possui uma taxa de transferência de 4KB/s. Sua pinagem consiste em GND, DATA e VCC, e suas dimensões são compactas (19mm x 19mm x 8mm). Amplamente utilizado em projetos de controle remoto e automação, o FS1000A é uma opção eficiente para comunicação sem fio em curtas distâncias (ZERO, 2017).



Figura 7 – Transmissor FS1000A.

2.8- Receptor RWS-371-6

O Receptor (Figura 8) RWS-371-6 é um componente usado para receber sinais de comunicação sem fio na frequência de 433,92MHz. Ele opera com modulação ASK (*Amplitude Shift Keying*) e possui um circuito LC (circuito composto por indutor e capacitor). Com uma taxa de dados de 4800 bps, seletividade de -108 dBm e espaçamento de canal de $\pm 500\text{KHz}$, é alimentado por 5V. Destaca-se por seu design de alta sensibilidade passiva e é projetado para uma aplicação fácil com poucos componentes externos. Geralmente utilizado em sistemas de controle remoto e automação, o RWS-371-6 é eficaz na recepção confiável de sinais sem fio (WENSHING, 2023).

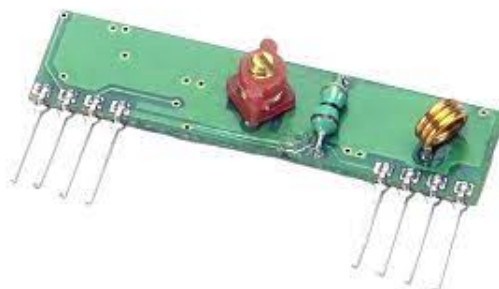


Figura 8 – Receptor RWS-371-6.

2.9- Power Bank Maxprint

O Power Bank Maxprint (Figura 9) é um carregador portátil projetado para recarregar smartphones e tablets em movimento. Com uma capacidade de 5200mAh e uma tensão de saída de 5V, oferece uma fonte de energia portátil para dispositivos eletrônicos. Possui uma bateria interna de Li-Íon recarregável, entrada de energia de 5V 1.0A e é compatível com dispositivos que possuem portas USB. Com dimensões compactas de 88mm x 45mm x 21mm, é construído com termoplásticos, metais e circuito eletrônico. Sua portabilidade e capacidade de carga rápida o tornam uma opção conveniente para alimentar sensores durante um dia inteiro (AUTOMAÇÃO, 2023).



Figura 9 – Power Bank Maxprint.

2.10- Arduino IDE

A Arduino IDE (Ambiente de Desenvolvimento Integrado) (Figura 10) é um software usado para programar placas Arduino. Com uma interface gráfica amigável, oferece um editor de código com destaque de sintaxe, compilador integrado e facilidade para upload de código para as placas. Além disso, possui um monitor serial para comunicação, bibliotecas padrão, compatibilidade com várias placas Arduino, suporte a diferentes sistemas operacionais e é um software de código aberto. Essa ferramenta é essencial para o desenvolvimento de projetos eletrônicos, sendo acessível a programadores e entusiastas, e integrada a um ecossistema abrangente de suporte e componentes (TEAM,

2023b).



Figura 10 – Arduino IDE.

2.11- SPI

A biblioteca SPI (*Serial Peripheral Interface*) é uma ferramenta de software que permite a comunicação serial síncrona entre microcontroladores e dispositivos periféricos, como sensores, memórias e outros microcontroladores. Ela oferece funções para configurar e controlar a transferência de dados por meio do protocolo SPI, facilitando a comunicação eficiente e confiável entre os dispositivos conectados (TEAM, 2024e).

2.12- Wire

A biblioteca Wire é uma biblioteca de software usada em plataformas Arduino para facilitar a comunicação com dispositivos periféricos que seguem o protocolo I2C (*Inter-Integrated Circuit*). Ela fornece funções para iniciar e controlar a comunicação I2C, permitindo que o Arduino atue como um mestre ou escravo em uma rede I2C. Essa biblioteca simplifica a implementação de comunicação entre dispositivos, permitindo que o Arduino troque dados com sensores, displays, módulos de expansão e outros dispositivos compatíveis com I2C de forma fácil e eficiente (TEAM, 2024f).

2.13- SparkFun_ADXL345

A biblioteca SparkFun_ADXL345 é uma biblioteca de software desenvolvida pela SparkFun Electronics para interagir com o acelerômetro ADXL345 em plataformas Arduino. Ela simplifica o processo de leitura e configuração do acelerômetro, fornecendo funções para inicialização, leitura de dados de aceleração nos eixos X, Y e Z, configuração de modos de operação e ajuste de parâmetros como taxa de amostragem e escala de medição. Essa biblioteca facilita a integração do acelerômetro ADXL345 em projetos Arduino, permitindo que os desenvolvedores obtenham medições precisas de aceleração com facilidade (TEAM, 2024d).

2.14- SecurityVanet

A biblioteca SecurityVanet é uma ferramenta de software voltada para garantir a segurança na comunicação de dispositivos Arduino em redes sem fio, especialmente em ambientes de comunicação veicular ad hoc (VANETs). Desenvolvida para o Arduino, essa biblioteca oferece funcionalidades para criptografia e autenticação de dados transmitidos, protegendo contra interceptação e manipulação por parte de terceiros não autorizados. Com a SecurityVanet, os desenvolvedores podem implementar camadas adicionais de segurança em suas aplicações, garantindo a integridade e confidencialidade das informações transmitidas em redes sem fio (FREITAS et al., 2021).

2.15- Ethernet

A biblioteca Ethernet é uma ferramenta de software projetada para facilitar a comunicação de dispositivos Arduino com redes Ethernet, permitindo que eles se conectem à Internet e a outros dispositivos em uma rede local. Essa biblioteca oferece uma série de funções e classes que simplificam a configuração e o uso de conexões

de rede Ethernet, incluindo a configuração de endereços IP, a definição de portas de comunicação e a troca de dados com servidores remotos por meio de protocolos TCP/IP. Com a biblioteca Ethernet, os desenvolvedores podem integrar facilmente recursos de rede em seus projetos Arduino, permitindo a comunicação com serviços online e outros dispositivos conectados à rede (TEAM, 2024b).

2.16- ArduinoJson

A biblioteca ArduinoJson é uma ferramenta de software desenvolvida para simplificar o processamento de dados no formato JSON em projetos Arduino. Ela oferece uma série de funções e classes que permitem aos desenvolvedores analisar, gerar e manipular dados JSON de forma eficiente e fácil. Com a ArduinoJson, os usuários podem realizar operações como análise de dados recebidos de servidores web, geração de respostas JSON para solicitações de clientes, e intercâmbio de dados estruturados entre dispositivos conectados à rede. Essa biblioteca é especialmente útil para projetos que envolvem comunicação de dados em formato JSON, tornando mais simples o desenvolvimento de aplicações baseadas em Arduino que requerem manipulação de informações nesse padrão de intercâmbio de dados (TEAM, 2024a).

2.17- RTCLib

A biblioteca RTCLib é uma ferramenta de software desenvolvida para facilitar o uso de módulos de relógio em tempo real (RTC) em projetos Arduino. Ela fornece uma interface simples para configuração e leitura de data e hora em dispositivos RTC, permitindo que os desenvolvedores acessem e gerenciem facilmente informações temporais em seus projetos. Com a RTCLib, os usuários podem integrar facilmente funcionalidades de tempo real em suas aplicações, como registro de eventos com carimbo de data e hora, programação de eventos periódicos e sincronização de relógios entre dispositivos interconectados. Isso torna a biblioteca RTCLib uma escolha popular para projetos que

exigem precisão temporal e controle de eventos baseados em tempo (TEAM, 2024c).

2.18- Spyder

O Spyder (Figura 11) é um ambiente de desenvolvimento integrado (IDE) projetado para programação em Python, com foco especial em análise de dados científicos. Oferece um console IPython interativo, editor de código avançado, explorador de variáveis e integração com bibliotecas científicas populares. Com ferramentas para criação de gráficos e visualizações, é altamente configurável, extensível e multiplataforma. Sua popularidade cresceu entre cientistas de dados e pesquisadores devido às suas características específicas para análise de dados. O Spyder é um software de código aberto, permitindo personalizações e contribuições da comunidade (SPYDER, 2023).



Figura 11 – Spyder.

3- Trabalhos Relacionados

3.1- A Novel Approach for IoT Based Wearable Health Monitoring and Messaging System

No trabalho de (MANAS et al., 2019) é apresentado um sistema de monitoramento de saúde vestível baseado em uma arquitetura de nuvem IoT que coleta diversos dados do usuário, como temperatura corporal e frequência cardíaca, além de conseguir detectar quedas do indivíduo. O dispositivo também consegue enviar mensagens de texto automáticas aos cuidadores e médicos caso haja uma emergência ou anomalias nos dados. Estes, no que lhe concerne, são coletados e enviados a um aplicativo remoto que possui uma interface própria para a exibição dos parâmetros de saúde. Esses dados são processados pelo aplicativo que gera um breve relatório de análise de saúde correspondente e o exibe na interface. Caso o relatório infira que a pessoa está angustiada, uma mensagem de texto é enviada a todas as pessoas que se registraram no aplicativo como amigos, familiares, médicos ou cuidadores, além de também serem armazenadas em um banco de dados online que pode ser acessado por qualquer pessoa que possua a URL compartilhada pelo próprio banco de dados. Além disso, o banco de dados plota um gráfico de tempo com as informações coletadas, que pode ser customizado para mostrar dados de qualquer um dos sensores durante um período.

Entretanto, notam-se algumas limitações neste projeto como a necessidade de um aplicativo para cada indivíduo, impedindo o monitoramento de várias pessoas em simultâneo, além da falta de segurança com os dados, uma vez que qualquer pessoa que tenha a URL disponibilizada pelo banco de dados pode ter acesso a essas informações.

Sendo assim, o diferencial do nosso projeto é que será possível monitorar várias pessoas em simultâneo, além dos dados, estarem protegidos e disponíveis apenas para os respectivos cuidadores.

3.2- Healthcare Monitoring System and transforming Monitored data into Real time Clinical Feedback based on IoT using Raspberry Pi

No estudo de (KHAN et al., 2019) foi desenvolvido um sistema acessível, rápido e portátil, utilizando Arduino, Raspberry Pi e sensores de pulso e temperatura, que aplica o conceito de IoT e topologias de rede. Os dados são coletados via Arduino UNO e enviados para o Raspberry Pi através de um cabo USB, que também possui uma câmera integrada para fins de vídeo. Todas as informações recebidas são lidas pelo Raspberry Pi a cada dois minutos e também é gravado, continuamente, o vídeo do paciente usando a câmera pi. Após essa etapa, os dados são enviados para um servidor de banco de dados, via Wi-Fi, que os armazena e os envia a uma página web chamada Dynamic Web, acessível ao médico de qualquer lugar, através do uso da internet. O médico entra no site usando uma senha personalizada de um local remoto e, então, ele pode ver o paciente continuamente e aconselhar a equipe usando o sistema de bate-papo e de gestos, presente na página. O médico não tem o direito de alterar nada nos dados, ele apenas atua como observador e conselheiro médico.

Diante disso, o foco deste trabalho é utilizar um dispositivo externo para o monitoramento do indivíduo, acarretando fácil perda, além de não garantir que a pessoa estará sempre utilizando o dispositivo. Também é possível notar que o aparelho coleta dados de maneira intermitente e que tem a função de enviar os dados para armazenamento externo, não estando disponíveis em tempo real, apenas para leitura futura.

Em contrapartida, a intenção deste trabalho é propor um protótipo que não seja invasivo, porém vestível, ou seja, um dispositivo pequeno e que permanece com a pessoa sempre que ela utiliza a vestimenta que o possui. Além disso, será possível monitorar os dados em tempo real visando, principalmente, o atendimento rápido em uma emergência.

3.3- Wearable Sensors for Remote Health Monitoring

Na pesquisa de (MAJUMDER; MONDAL; DEEN, 2017) foi realizado um levantamento de última geração sobre parâmetros fisiológicos e sistemas de monitoramento de atividades desenvolvidos em uma plataforma vestível, permitindo que as pessoas conduzam vidas independentes e ativas em seu ambiente familiar, em simultâneo, em que garantem uma vigilância contínua, não intrusiva, não invasiva e perfeita de sua saúde e bem-estar físico. Primeiramente, discorrem a respeito dos sistemas de medição bio-potencial e de um sistema BSN de múltiplos sensores. Em seguida, tratam de temas como a segurança dos dados, o baixo consumo de energia dos dispositivos e a acessibilidade destes. Por fim, é apresentada uma breve revisão sobre sensores de base têxtil, destacando suas aplicações na detecção de sinais fisiológicos.

Desta forma, fica evidente que o projeto não propõe nenhum dispositivo de monitoramento remoto de saúde, apenas apresenta diversas formas como estes dispositivos são criados, quais tecnologias possuem e os cuidados que se deve ter ao fazê-los, além da utilidade de cada um deles na vida das pessoas.

O presente projeto procura, não somente, apresentar o funcionamento e a utilidade de dispositivos de monitoramento remoto de saúde, mas também propor um protótipo de saúde vestível para auxiliar, especialmente, idosos em asilos.

4- Proposta

Para o cumprimento do objetivo do presente trabalho, desenvolveremos um protótipo vestível para monitoramento especialmente de idosos em asilos, por meio de uma pesquisa aplicada de estudo de caso. Esse protótipo é baseado em computação vestível, contendo uma camisa com sensores embutidos, minimamente invasivos e não intrusivos, de temperatura, batimentos cardíacos e um acelerômetro, que serão controlados por um Arduino e alimentados por uma bateria recarregável.

Utilizando o método indutivo na pesquisa experimental (COELHO, 2020), o Arduino (micro-computador que pode ser utilizado em projetos *wearables*) tratará esses dados e os transmitirá via módulo RF (transmissor e receptor de ondas eletromagnéticas de frequência de 433MHz) para um outro receptor Arduino, que estará conectado a energia e terá junto a ele uma Ethernet Shield capaz de enviar esses dados para rede.

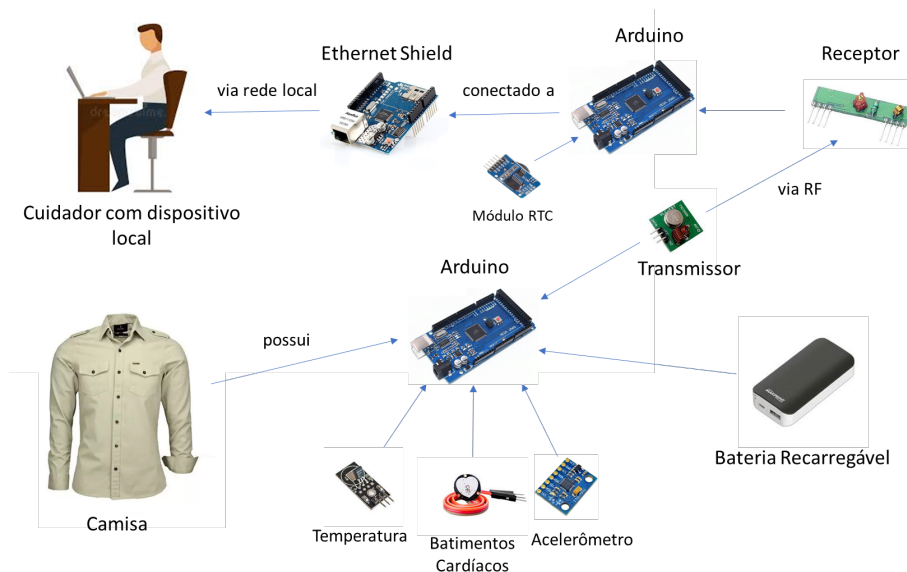


Figura 12 – Esquema conceitual do protótipo desenvolvido.

Através de um dispositivo conectado a LAN (Rede Local) e ao executar um aplicativo, os dados serão armazenados no dispositivo local dos respectivos cuidadores. Uma vez na máquina local, os encarregados pelo idoso poderão visualizar os dados obtidos pelos sensores em uma página web que abrirá automaticamente ao executar o aplicativo. Diante disso, é possível aplicar o protótipo desenvolvido a outros indivíduos, permitindo monitorar

vários simultaneamente, e auxiliando os cuidadores na assistência aos idosos, conforme Figura 12.

4.1- Implementação

A implementação do protótipo vestível para monitoramento de saúde de idosos em asilos envolve a integração de diversos componentes eletrônicos e o desenvolvimento de um sistema interconectado. Abaixo estão os principais aspectos da implementação¹.

4.1.1 HARDWARE

O Arduino Mega 2560 (Figura 1) é empregado como o controlador central do protótipo, inserido na camisa do paciente. Incorporam-se a ele os sensores de batimentos cardíacos Amped 1.5 (Figura 3) e de temperatura LM35 (Figura 4), além do acelerômetro ADXL345 (Figura 5). Esses sensores visam monitorar os batimentos cardíacos (em bpm), a temperatura corporal (em °C) e a posição do paciente em relação ao solo, respectivamente.

Os dados coletados passam por processamento e criptografia no Arduino Mega (Figura 1), sendo transmitidos pelo transmissor F1000A (Figura 7), conectado ao controlador central, via rádio frequência na faixa de 433 MHz. A energia para o funcionamento do sistema é provida pelo Power Bank Maxprint (Figura 9), garantindo autonomia e mobilidade sem necessidade de cabeamento (Figura 13).

No receptor, um segundo Arduino Mega 2560 (Figura 1), acompanhado do módulo RWS-371-6 (Figura 8), recebe os dados, realiza a descriptografia e os encaminha para um servidor web por meio da Ethernet Shield (Figura 2). Vale ressaltar que o arduino (Figura 1) está conectado à tomada por meio da fonte de alimentação própria para ele e a Ethernet Shield (Figura 2) está conectada à internet via cabo de rede (Figura 14).

¹<https://github.com/LabRedesCefetNF/WearableHealthMonitoring>

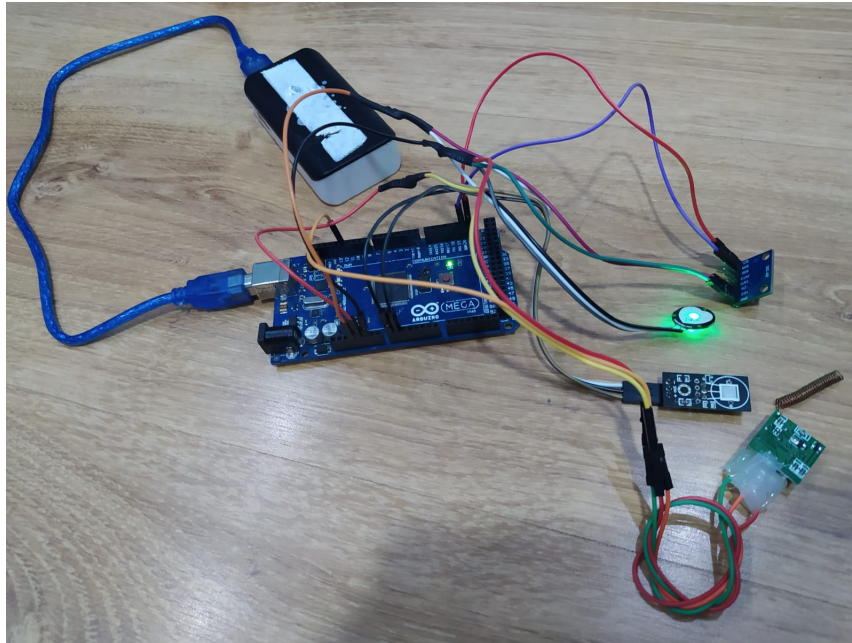


Figura 13 – Transmissor.

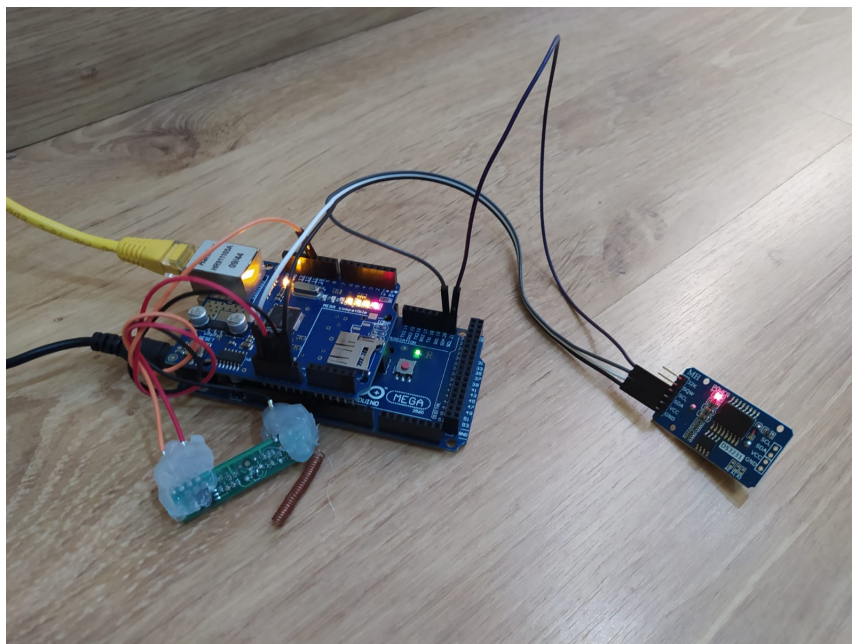


Figura 14 – Receptor.

4.1.2 SOFTWARE

4.1.2.1 Transmissor

Usando o Arduino IDE (Figura 10), o código começa incluindo bibliotecas necessárias, como SPI, Wire, SparkFun_ADXL345 e SecurityVanet. Em seguida, são

que inclui informações sobre batimentos cardíacos, temperatura, posição do corpo e a identificação do paciente. Essa string é transmitida usando a função "vanet.transmit", presente na biblioteca "SecurityVanet.h" (Figura 18).

```

69 // Concatena os valores dos sensores para serem enviados para o receptor
70 String transmissor = "";
71 transmissor.concat(String(bpm,0));
72 transmissor.concat("-");
73 transmissor.concat(String(temp,1));
74 transmissor.concat("-");
75
76 Serial.print("SITUAÇÃO - ");
77
78 if( z > 140 ){
79     Serial.println("Deitado de bruços");
80     transmissor.concat("B-");
81 } else if( ( z > 90 ) && ( y < -50 ) ){
82     Serial.println("Deitado de ponta cabeça");
83     transmissor.concat("P");

```

Figura 18 – Formato dos dados e transmissão.

Para também indicar visualmente a transmissão, o estado de um LED conectado ao pino digital 13 (presente no próprio arduino) é alternado. Um atraso de 4 segundos é incluído no final do loop para controlar a frequência de atualização e permitir a transmissão dos dados (Figura 19).

```

108 if(digitalRead(13) == HIGH){
109     digitalWrite(13, LOW);
110 } else {
111     digitalWrite(13, HIGH);
112 }
113
114 delay(4000); /* Atraso de 4s - Tempo aproximado até a nova atualização
115 | | | | | da página web com novos dados (total de 5 novos dados)*/
116 }

```

Figura 19 – Indicação visual e atraso.

4.1.2.2 Receptor

Usando também o Arduino IDE (Figura 10), o código inicia importando as bibliotecas necessárias, incluindo SPI, Wire, SecurityVanet, Ethernet, ArduinoJson e RTCLib. Em seguida, são definidos os pinos de transmissão e recepção para o módulo de comunicação RF (Radio Frequência), uma instância da classe SecurityVanet ("vanet"), e um objeto de

relógio em tempo real (RTC_DS3231) para interação com o módulo RTC (Figura 20).

```

1  #include <SPI.h> // Comunicação serial síncrona
2  #include <Wire.h> // Comunicação I2C
3  #include <SecurityVanet.h> // Comunicação de rede
4  #include <Ethernet.h> // Configurar e controlar a comunicação Ethernet no Arduino
5  #include <ArduinoJson.h> // Biblioteca para manipulação de JSON
6  #include <RTCLib.h> // Interação com módulos de relógio em tempo real (RTC)
7
8  #define TxPin 12 // Pino de transmissão
9  #define RxPin 11 // Pino de recepção
10
11 SecurityVanet vanet;
12 RTC_DS3231 rtc;

```

Figura 20 – Configuração inicial.

O endereço MAC e o endereço IP são configurados para o módulo Ethernet. O código utiliza DHCP para obter um endereço IP automaticamente (Figura 21). Em caso de falha na obtenção do endereço IP, o programa entra em um loop infinito, indicando um problema na configuração de rede (Figura 22).

```

14 // Complete com as informações da sua rede
15 byte mac[] = { 0xDE, 0xAD, 0xBE, 0xEF, 0xFE, 0xED };
16 IPAddress ip(192, 168, 1, 113); // Verificar a disponibilidade de IPs do roteador conectado!

```

Figura 21 – Configuração de rede.

```

37 if (Ethernet.begin(mac) == 0) {
38   Serial.println("Falha ao configurar Ethernet usando DHCP");
39   while (true) {
40     delay(1);
41   }
42 }

```

Figura 22 – Falha na configuração de rede.

O servidor web é inicializado na porta padrão HTTP (porta 80) (Figura 23). A inicialização também inclui a configuração do módulo RTC. Se o módulo RTC estiver sem energia, o código ajusta automaticamente a hora e a data usando a marca de tempo no momento da compilação (Figura 24).

```

26 // Inicializa o webserver com a porta 80 - porta padrão do HTTP
27 EthernetServer server(80);

```

Figura 23 – Inicialização do servidor web.


```

44   if (!rtc.begin()) {
45       Serial.println("Não foi possível encontrar o módulo RTC. Verifique a conexão!");
46       while (1);
47   }
48
49   // Descomente a linha abaixo para ajustar manualmente a hora e a data
50   // rtc.adjust(DateTime(2023, 11, 27, 18, 41, 00));
51
52   if (rtc.lostPower()) {
53       Serial.println("RTC perdeu a energia! Ajustando a hora e a data...");
54       // Ajuste manual da hora e da data
55       rtc.adjust(DateTime(F(__DATE__), F(__TIME__)));
56   }

```

Figura 24 – Inicialização do módulo RTC.

O loop principal do código é dividido em duas partes principais. A primeira parte lida com o recebimento de dados provenientes do módulo de comunicação RF ("vanet.receiver()"). Caso novos dados sejam recebidos, o LED no pino 13 é alternado, e as leituras dos sensores são armazenadas em um array circular ("sensorReadings") (Figura 25).

```

65   void loop() {
66       if (vanet.receiver()) {
67           if(digitalRead(13) == HIGH){
68               digitalWrite(13, LOW);
69           } else {
70               digitalWrite(13, HIGH);
71           }
72
73           String newReading = vanet.getPlainTxt();
74           sensorReadings[currentIndex] = newReading;
75           currentIndex = (currentIndex + 1) % 5; // Avança o índice circularmente
76           Serial.print(newReading);
77           // Extrai o valor do paciente da substring(10)
78           paciente = newReading.substring(10);
79       } else {
80           paciente = sensorReadings[0].substring(10);
81       }

```

Figura 25 – Loop principal: recebimento de dados.

A segunda parte do loop lida com a interação do servidor web. Quando um cliente web se conecta, o código verifica as requisições HTTP recebidas. Se uma solicitação válida for identificada, uma resposta HTTP é preparada, indicando o tipo de conteúdo como JSON e configurando a atualização automática da página a cada 20 segundos. Os dados do paciente, incluindo as últimas 5 leituras dos sensores, são organizados em um objeto JSON usando a biblioteca ArduinoJson. O JSON é então serializado e enviado como resposta ao cliente web (Figura 26).

```

83 // Fica escutando requisições de novos clientes
84 EthernetClient client = server.available();
85 if (client) {
86     Serial.println("Novo cliente");
87     boolean currentLineIsBlank = true; // Uma solicitação HTTP termina com uma linha em branco
88     while (client.connected()) {
89         if (client.available()) {
90             char c = client.read();
91             Serial.write(c);
92             if (c == '\n' && currentLineIsBlank) {
93                 // Envia um cabeçalho de resposta HTTP padrão
94                 client.println("HTTP/1.1 200 OK");
95                 client.println("Content-Type: application/json"); // Alterado para JSON
96                 client.println("Connection: close"); // A conexão será fechada após a conclusão da resposta
97                 client.println("Refresh: 20"); // Atualiza a página automaticamente a cada 20 segundos

```

Figura 26 – Loop principal: servidor web.

Por fim, duas funções auxiliares são definidas: a primeira "getSituacaoAcelerometro" converte a leitura do acelerômetro em uma descrição da situação do paciente (Figura 27) e a segunda "getFormattedTime" formata a data e hora obtidas do módulo RTC (Figura 28).

```

141 String getSituacaoAcelerometro(String acelerometro) {
142     if (acelerometro == "N") {
143         return "De pé";
144     } else if (acelerometro == "C") {
145         return "Deitado de costas";
146     } else if (acelerometro == "B") {
147         return "Deitado de bruços";
148     } else if (acelerometro == "D") {
149         return "Deitado para direita";
150     } else if (acelerometro == "E") {
151         return "Deitado para esquerda";
152     } else {
153         return "Deitado de ponta cabeça";
154     }
155 }

```

Figura 27 – Função auxiliar "getSituacaoAcelerometro".

```

157 String getFormattedTime(DateTime now) {
158     String formattedTime = "";
159
160     // Adiciona o dia
161     formattedTime += String(now.day());
162     formattedTime += "/";
163
164     // Adiciona o mês
165     formattedTime += String(now.month());
166     formattedTime += "/";
167
168     // Adiciona o ano
169     formattedTime += String(now.year());
170     formattedTime += " ";

```

Figura 28 – Função auxiliar "getFormattedTime".

4.1.2.3 Aplicativo

O arquivo "prototipo.py" utiliza a linguagem de programação Python, no ambiente de desenvolvimento Spyder (Figura 11), e possui a funcionalidade de coletar os dados do servidor web gerado anteriormente em intervalos regulares, salvá-los localmente, gerar gráficos com as informações coletadas e atualizar automaticamente uma página HTML para exibir esses dados de forma tabular.

O script começa importando diversas bibliotecas, como "os", "requests", "json", "time", "matplotlib", "datetime" e "selenium". Essas bibliotecas desempenham papéis cruciais na manipulação de arquivos, comunicação via HTTP, processamento de dados no formato JSON, gestão do tempo, geração de gráficos e automação do navegador web (Figura 29).

```
1  import os
2  import requests
3  import json
4  import time
5  import matplotlib.pyplot as plt
6  from datetime import datetime
7  from selenium import webdriver
```

Figura 29 – Configuração inicial e importação de bibliotecas.

O código inclui a declaração de variáveis globais e a inicialização de um driver do navegador Chrome por meio da biblioteca Selenium. A variável global "id_guia_html" é utilizada para rastrear e armazenar o identificador da guia do navegador (Figura 30).

```
9  # Variável global para armazenar o ID da guia
10 id_guia_html = None
11
12 # Inicializar o driver do navegador
13 driver = webdriver.Chrome() # Você pode precisar ajustar o caminho do driver para o seu ambiente
```

Figura 30 – Variáveis globais e inicialização.

A função "obter_e_salvar_dados" é responsável por realizar uma solicitação HTTP GET para obter dados do servidor web. Em caso de uma resposta bem-sucedida (código de status 200), os dados são filtrados e salvos localmente em um formato JSON (Figura 31).

```

16 # Função para obter e salvar dados
17 def obter_e_salvar_dados():
18     url = "http://192.168.1.113/" # Substitua pela URL correta
19
20     try:
21         response = requests.get(url)
22         if response.status_code == 200:
23             dados = response.json()
24
25             if dados:
26                 primeiro_nome_chave = next(iter(dados))
27                 nome_pasta = primeiro_nome_chave
28                 nome_arquivo = f"{nome_pasta}/{primeiro_nome_chave}.txt"
29
30                 if not os.path.exists(nome_pasta):
31                     os.makedirs(nome_pasta)

```

Figura 31 – Função "obter_e_salvar_dados".

A função "plotar_graficos" gera gráficos de temperatura, batimentos cardíacos e situação do paciente a partir dos dados contidos no arquivo JSON. Os gráficos são salvos como imagens PNG (Figura 32).

```

64 # Função para plotar gráficos de temperatura, batimentos e situação do paciente
65 def plotar_graficos(nome_arquivo):
66     with open(nome_arquivo, "r", encoding="utf-8") as file:
67         dados = file.read()
68
69         if dados:
70             dados = json.loads(dados)
71         else:
72             dados = []
73
74         dados = [dict(t) for t in {tuple(d.items()) for d in dados}]
75
76         dados_temperatura = sorted(dados, key=lambda x: datetime.strptime(x['dataHora'], '%d/%m/%Y %H:%M:%S'))
77         dados_batimentos = sorted(dados, key=lambda x: datetime.strptime(x['dataHora'], '%d/%m/%Y %H:%M:%S'))
78         dados_situacao = sorted(dados, key=lambda x: datetime.strptime(x['dataHora'], '%d/%m/%Y %H:%M:%S'))

```

Figura 32 – Função "plotar_graficos".

A função "abrir_pagina_html" verifica se uma página HTML existe; se não, cria uma com base em um modelo incorporado. Em seguida, abre a página no navegador, armazenando o identificador da guia para referências futuras (Figura 33).

```

129 # Função para abrir a página HTML
130 def abrir_pagina_html():
131     global id_guia_html # Utiliza a variável global
132
133     nome_pagina_html = "prototipo.html"
134
135     # Verificar se a página HTML existe, senão, criar
136     if not os.path.exists(nome_pagina_html):
137         with open(nome_pagina_html, "w", encoding="utf-8") as html_file:
138             html_file.write(
139                 """
140                 <html>
141                 <head>
142                     <title>Protótipo</title>
143                     <link rel="stylesheet" href="https://fonts.googleapis.com/icon?family=Material+Icons">
144                     <link rel="stylesheet" href="https://code.getmdl.io/1.3.0/material.indigo-pink.min.css">
145                     <script defer src="https://code.getmdl.io/1.3.0/material.min.js"></script>

```

Figura 33 – Função "abrir_pagina_html".

A função "atualizar_tabela_html" lê os dados do arquivo JSON e atualiza dinamicamente o conteúdo de uma tabela HTML, adicionando novas linhas conforme necessário

(Figura 34).

```

222 # Função para atualizar a tabela HTML com dados do arquivo
223 def atualizar_tabela_html(nome_arquivo):
224     nome_pagina_html = "prototipo.html"
225     with open(nome_arquivo, "r", encoding="utf-8") as file:
226         dados = file.read()
227
228         if dados:
229             dados = json.loads(dados)
230         else:
231             dados = []
232
233     # Criar uma linha na tabela para cada registro nos dados
234     linhas_tabela = ""
235     for registro in dados:
236         # Adicione a seguinte verificação para definir a cor da linha
237         cor_linha = "background-color: #FFEBEE;" if registro['situacao'] != "De pé" else ""

```

Figura 34 – Função "atualizar_tabela_html".

Por fim, o script entra em um loop infinito onde, a cada 20 segundos, invoca as funções mencionadas acima. Isso garante uma atualização regular dos dados, gráficos e da visualização tabular na página HTML (Figura 35).

```

314 # Executar a função de obter e salvar dados a cada 20 segundos
315 while True:
316     nome_arquivo = obter_e_salvar_dados()
317     if nome_arquivo:
318         plotar_graficos(nome_arquivo)
319         abrir_pagina_html()
320         atualizar_tabela_html(nome_arquivo)
321     time.sleep(20)

```

Figura 35 – Loop principal.

Dessa forma, o sistema oferece uma solução integrada e eficiente para monitoramento de saúde de idosos em asilos, proporcionando coleta precisa de dados, transmissão segura e acessibilidade remota aos cuidadores.

5- Estudo de Caso

Em um cenário de Instituição de Longa Permanência para Idosos (ILPI), a equipe de enfermagem se depara com a necessidade constante de monitorar a saúde dos residentes. Em situações que demandam respostas rápidas, como quedas ou emergências médicas, a busca por soluções inovadoras se torna imperativa.

Com o propósito de enfrentar esse desafio, foi desenvolvido um sistema vestível de monitoramento, integrando sensores de batimentos cardíacos, temperatura e acelerômetro em uma camisa especial. A implementação visa fornecer à equipe de enfermagem dados em tempo real sobre os parâmetros vitais e a situação física dos idosos, oferecendo uma resposta ágil a eventos críticos.

Para provar o conceito do sistema, uma simulação¹ foi realizada envolvendo uma queda simulada. Inicialmente, o acelerômetro integrado à camisa detectou a posição inicial do paciente, neste caso estava de pé (36), e quando ele caiu o acelerômetro detectou de maneira precisa a mudança de posição, enviando os dados correspondentes ao sistema central em tempo real (37).

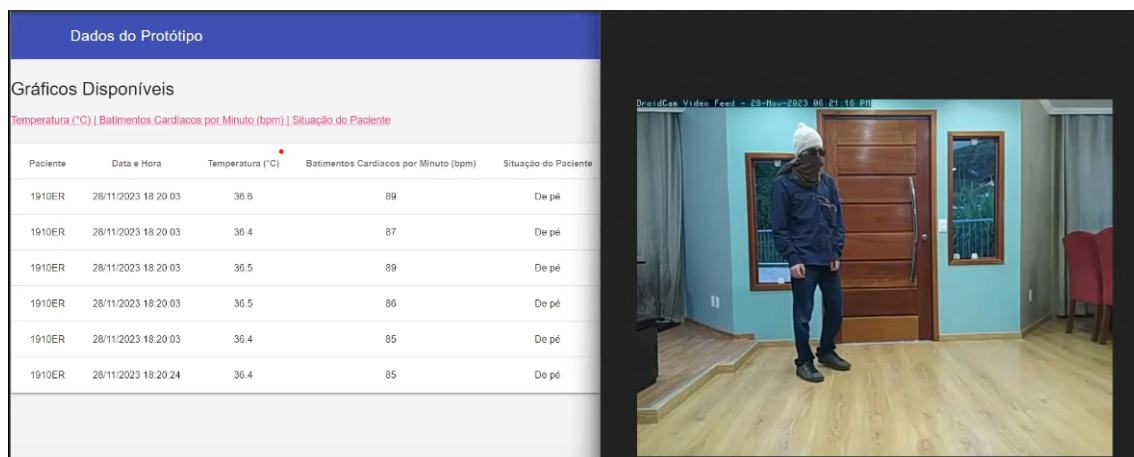


Figura 36 – Paciente de pé.

A resposta rápida do sistema foi evidenciada pelo processamento imediato dos dados pelo Arduino Mega 2560, que transmitiu as informações para um segundo Arduino conectado à internet. Em paralelo, um aplicativo Python coletou, analisou e exibiu os

¹<https://www.youtube.com/watch?v=Uvz3LJdC63M>

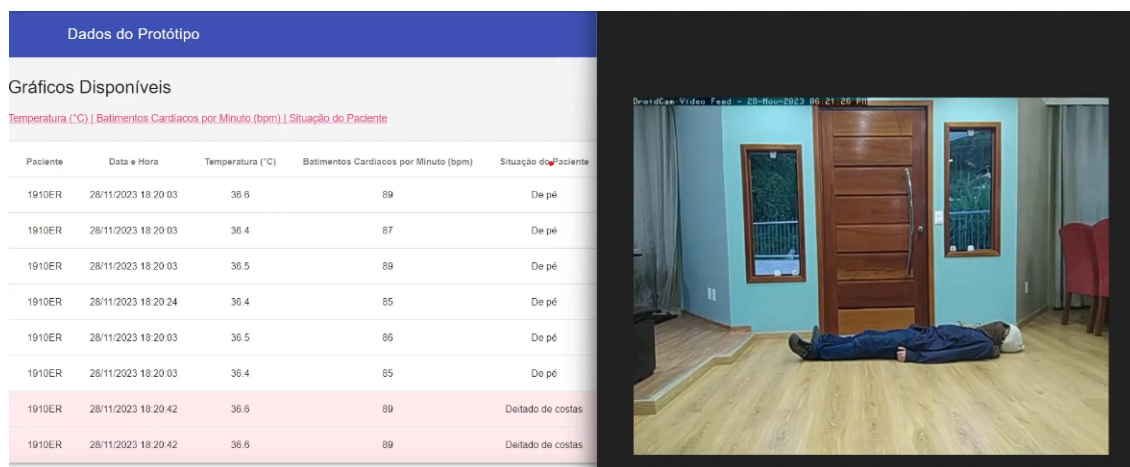


Figura 37 – Queda do paciente.

dados em intervalos regulares, gerando gráficos representativos dos batimentos cardíacos (38), temperatura corporal (39) e situação do corpo durante a simulação (40).

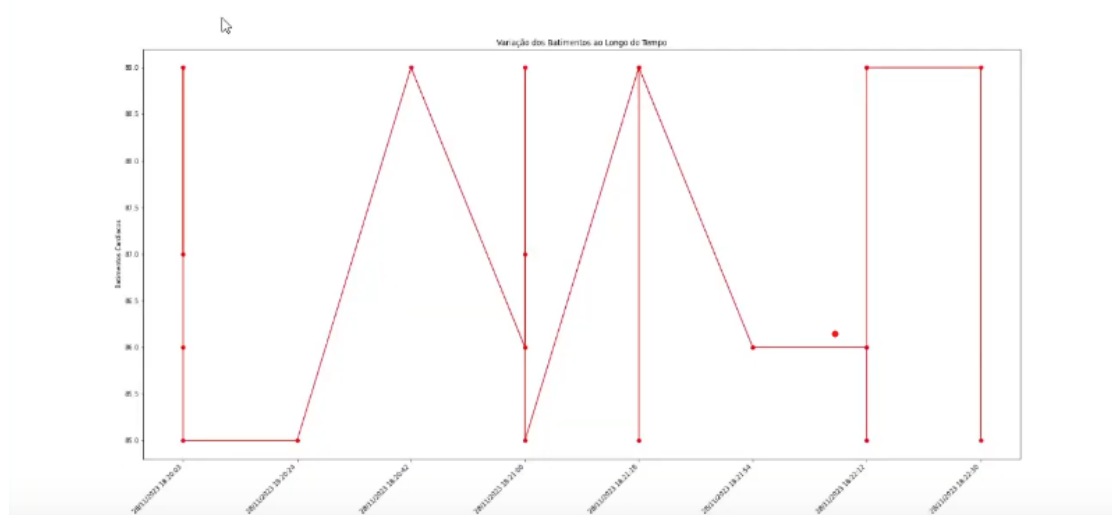


Figura 38 – Gráfico de batimentos cardíacos por minuto (bpm) do paciente.

Os prints do aplicativo Python destacam os gráficos em tempo real, proporcionando uma visão clara das variações nos dados vitais durante o período de simulação. O sistema demonstrou não apenas a capacidade de identificar simulações de quedas, mas também de fornecer dados cruciais para a equipe de enfermagem, permitindo uma tomada de decisão mais ágil.

O estudo de caso reforça o conceito do sistema de monitoramento vestível em um contexto simulado de uma instituição de cuidados a idosos. A detecção de quedas e a entrega de dados em tempo real destacam a utilidade prática desse sistema, contribuindo

6- Conclusão

O desenvolvimento do protótipo vestível para monitoramento de saúde em idosos em instituições de cuidados apresentou resultados promissores e soluções inovadoras para os desafios enfrentados na assistência a essa população. Ao integrar sensores minimamente invasivos e uma arquitetura de comunicação, a solução proposta visa oferecer uma abordagem abrangente e acessível para o monitoramento contínuo da saúde dos idosos.

Através de um estudo de caso, o protótipo, apesar de ter sido testado em uma única pessoa para este estudo, demonstrou sua capacidade de capturar dados de temperatura, batimentos cardíacos e posição corporal. Essa capacidade permite vislumbrar sua aplicação em um cenário mais amplo, podendo monitorar vários usuários simultaneamente, proporcionando uma visão em tempo real do estado de saúde de cada indivíduo.

Durante a simulação de um incidente de queda, o sistema reagiu rapidamente, registrando as mudanças na aceleração e alertando a equipe de cuidadores. A visualização dos dados por meio de gráficos e a atualização dinâmica da página HTML ofereceram uma experiência amigável aos cuidadores, facilitando o acompanhamento das condições de saúde do idoso.

A implementação do sistema, combinando hardware e software de forma integrada, permitiu uma transmissão eficiente de dados entre o dispositivo vestível e o servidor remoto. A segurança dos dados foi abordada por meio de criptografia, garantindo que as informações sensíveis dos pacientes permaneçam protegidas durante a transmissão e armazenamento.

As principais vantagens do protótipo incluem a capacidade de monitoramento em tempo real, a mobilidade proporcionada pela alimentação via bateria recarregável e a acessibilidade remota aos dados. Além disso, a proposta de utilizar uma camisa como base do dispositivo contribui para a comodidade e aceitação pelos usuários.

No entanto, algumas considerações devem ser levadas em conta para melhorias futuras. A necessidade de otimização do código, visando principalmente aperfeiçoar o hardware, a expansão das funcionalidades para abranger mais parâmetros de saúde e a realização de testes mais abrangentes em ambientes do mundo real são aspectos que

podem ser aprimorados na evolução do protótipo.

Em síntese, o protótipo vestível desenvolvido para monitoramento de saúde em idosos representa um avanço significativo na integração de tecnologias para assistência a uma população vulnerável. Ao fornecer informações em tempo real, permitindo uma resposta ágil a emergências e promovendo a autonomia dos idosos, o sistema destaca-se como uma ferramenta valiosa no contexto do cuidado à saúde.

A contribuição deste trabalho não se limita apenas à tecnologia, mas também ao potencial impacto positivo na qualidade de vida dos idosos e na eficiência dos profissionais de saúde que lidam com essa população.

Este estudo abre portas para futuras pesquisas e desenvolvimentos na área de tecnologias assistivas para idosos, estimulando a inovação em soluções que promovam o bem-estar e a segurança dessa parcela da população cada vez mais relevante em nossa sociedade.

Referências

ASAP, Comitê Técnico. **O Envelhecimento da população**. Online: ASAP Saúde, 2021. <https://asapsaude.org.br/jornada-de-gsp/o-envelhecimento-da-populacao/3672/>.

AUTOMAÇÃO, HM. **Carregador Portátil Celular Power-Bank Preto 5200mAh Maxprint 6011554**. pt. Online: HM Automação, 2023. <https://hmautomacao.com.br/carregador-portatil-celular-power-bank-preto-5200mah-maxprint-6011554.html>.

BENTO, Maria da Conceição Saraiva da Costa; AMARAL, António Salgueiro; SILVA, Abel Paiva e. IDOSOS A CUIDAR DE IDOSOS: UM DESAFIO À ORGANIZAÇÃO DOS CUIDADOS DOMICILIÁRIOS. pt. **Cogitare Enfermagem**, v. 26, e79093, nov. 2021. ISSN 1414-8536, 2176-9133. DOI: 10.5380/ce.v26i0.79093.

CARE, Bem Me. **Febre e Infecções em Idosos**. pt. Online: [s.n.], out. 2019. <https://www.bem-me-care.com/single-post/2019/10/12/febre-em-idosos>.

CAVICCHIOLI, Giorgia; VILARDAGA, Vicente. **O abandono dos idosos no Brasil**. pt-br. Online: ISTOÉ, mai. 2021. <https://istoe.com.br/o-abandono-dos-idosos-no-brasil/>.

COELHO, Beatriz. **Como delimitar a metodologia científica do seu trabalho?** Online: Mettzer, out. 2020. <https://blog.mettzer.com/metodologia-cientifica/>.

CTCOR. **Como o coração reage ao envelhecimento? Descubra os efeitos desse processo natural no sistema cardiovascular**. Online: CTCor - Centro de Tratamento do Coração. <https://www.ctcor.com.br/blog/como-o-coracao-reage-ao-envelhecimento-descubra-os-efeitos-desse-processo-natural-no-sistema-cardiovascular/>.

CUNHA, Alfredo; LOURENÇO, Roberto. Quedas em idosos: prevalência e fatores associados. **Revista Hospital Universitário Pedro Ernesto**, v. 13, n. 2, mar. 2014. ISSN 1983-2567, 1676-8280. DOI: 10.12957/rhupe.2014.10128.

DA SAÚDE, MINISTÉRIO. **SAÚDE BRASIL 2108: Uma análise da situação de saúde e das doenças e agravos crônicos: desafios e perspectivas**. Brasília: Ministério da Saúde, 2019. https://bvsms.saude.gov.br/bvs/publicacoes/saude_brasil_2018_analise_situacao_saude_doencas_agravos_cronicos_desafios_perspectivas.pdf.

DEVICES, Analog. **Data Sheet ADXL345**. [S.l.]: One Analog Way, 2022. <https://www.analog.com/media/en/technical-documentation/data-sheets/adxl345.pdf>.

FAGUNDES, Karolina Vitorelli Diniz Lima et al. Instituições de longa permanência como alternativa no acolhimento das pessoas idosas. pt. **Revista de Salud Pública**, v. 19, p. 210–214, abr. 2017. ISSN 0124-0064, 0124-0064. DOI: 10.15446/rsap.v19n2.41541.

FREITAS, João et al. Comunicação segura em VANET. In: ANAIS da XIX Escola Regional de Redes de Computadores. Charqueadas/RS: SBC, 2021. p. 109–114. DOI: 10.5753/errc.2021.18551.

G. STEFANACCI, Richard. Mudanças no corpo com o envelhecimento - Questões sobre a saúde de pessoas idosas. pt-BR. **Manual MSD Versão Saúde para a Família**, 2022. publisher: Thomas Jefferson University.

INSTRUMENTS INCORPORATED, Texas. **LM35 Precision Centigrade Temperature Sensors**. en. [S.l.]: Texas Instruments Incorporated, 2017.

INTEGRATED, Maxim. **DS3231**. en. [S.l.]: Analog Devices, 2015.

KHAN, Imran et al. Healthcare Monitoring System and transforming Monitored data into Real time Clinical Feedback based on IoT using Raspberry Pi. In: 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). [S.l.: s.n.], jan. 2019. p. 1–6. DOI: 10.1109/ICOMET.2019.8673393.

LOPES, Danielle de Freitas et al. Fatores relacionados a quedas em idosos. pt. **Revista de Iniciação Científica e Extensão**, v. 2, n. 3, p. 131–138, ago. 2019. ISSN 2595-4261.

MAJUMDER, Sumit; MONDAL, Tapas; DEEN, M. Jamal. Wearable Sensors for Remote Health Monitoring. en. **Sensors**, v. 17, n. 1, p. 130, jan. 2017. ISSN 1424-8220. DOI: 10.3390/s17010130.

MANAS, Munish et al. A novel approach for IoT based wearable health monitoring and messaging system. en. **Journal of Ambient Intelligence and Humanized Computing**, v. 10, n. 7, p. 2817–2828, jul. 2019. ISSN 1868-5145. DOI: 10.1007/s12652-018-1101-z.

MONITORAMENTO, LifeLink. Monitoramento para idosos: o que é e como funciona? pt-BR. **Viva — Blog VIVA**, abr. 2021.

MOTA MARTINS, Renata; ROCHA MATOS, Thalyson. **Monitor de Batimentos Cardíacos**. pt-br. [S.l.]: Universidade Federal de Rondônia - UNIR, 2012.

NETO, José Gomes et al. IDOSO X MORTE VIOLENTA: AVALIAÇÃO EPIDEMIOLÓGICA DAS MORTES DECORRENTES DE AGENTES EXTERNOS E UMA PROPOSTA DE PREDIÇÃO DE RISCO. pt. **Revista Uningá**, v. 51, n. 3, mar. 2017. ISSN 2318-0579. DOI: 10.46311/2318-0579.51.eUJ1369.

OLIVEIRA, Marcela Pereira; PAULO, Mara Cibele Santos Silva; MONTE MOR FILHO, Paulo Eduardo. Prevenindo quedas na velhice: Atuação do enfermeiro na Atenção Primária à Saúde. **Revista Kairós : Gerontologia**, v. 21, n. 3, p. 359–372, set. 2018. ISSN 2176-901X, 1516-2567. DOI: 10.23925/2176-901X.2018v21i3p359-372.

OLIVEIRA, Martha Regina De; VERAS, Renato Peixoto; CORDEIRO, Hésio De Albuquerque. A importância da porta de entrada no sistema: o modelo integral de cuidado para o idoso. pt. **Physis: Revista de Saúde Coletiva**, v. 28, e280411, fev. 2019. ISSN 0103-7331, 0103-7331, 1809-4481. DOI: 10.1590/s0103-73312018280411.

R. VIEIRA, Juan Lucas; PONTE JUNIOR, Luiz Antonio da. An Introduction to the Internet of Healthcare Things. pt-BR. **An Introduction to the Internet of Healthcare Things**, 2018. publisher: Uff.br.

SILVA, Raimunda Magalhães da et al. Desafios e possibilidades dos profissionais de saúde no cuidado ao idoso dependente. pt. **Ciência & Saúde Coletiva**, v. 26, p. 89–98, jan. 2021. ISSN 1413-8123, 1678-4561. DOI: 10.1590/1413-81232020261.31972020.

SPYDER. **Welcome to Spyder's Documentation — Spyder 5 documentation**. en. Online: Spyder, 2023. <https://docs.spyder-ide.org/current/index.html>.

TEAM, The Arduino. **Arduino Ethernet Shield — Arduino Documentation**. en. [S.l.]: Arduino.cc, 2023. <https://docs.arduino.cc/retired/shields/arduino-ethernet-shield-without-poe-module>.

_____. **Arduino Integrated Development Environment (IDE) v1 — Arduino Documentation**. en. Online: Arduino, 2023.

_____. **ArduinoJson - Arduino Reference**. en. [S.l.: s.n.], 2024. publisher: Arduino.

_____. **Ethernet - Arduino Reference**. en. [S.l.: s.n.], 2024. publisher: Arduino.

TEAM, The Arduino. **Mega 2560 Rev3 — Arduino Documentation**. en. [S.l.: s.n.], 2023.

_____. **RTCLib - Arduino Reference**. en. [S.l.: s.n.], 2024. publisher: Arduino.

_____. **SparkFun ADXL345 Arduino Library - Arduino Reference**. en. Online: Arduino, 2024.

_____. **SPI - Arduino Reference**. en. Online: Arduino, 2024.

_____. **Wire - Arduino Reference**. en. Online: Arduino, 2024.

WENSHING. **RWS-371 RF MODULE Series**. en. [S.l.]: Cloud Front.

ZERO, Energia. **Complete Guide for RF 433MHz Transmitter**. en. [S.l.]: Energia Zero, 2017.