

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA
CÂMPUS DE ILHA SOLTEIRA**

JOÃO VITOR FROIS DA COSTA BARREIROS

**SISTEMA EMBARCADO PARA COLETA E TRANSMISSÃO DE DADOS VIA
REDE GPRS**

Ilha Solteira
2024

JOÃO VITOR FROIS DA COSTA BARREIROS

**SISTEMA EMBARCADO PARA COLETA E TRANSMISSÃO DE
DADOS VIA REDE GPRS**

Trabalho de Conclusão de Curso
apresentado à Faculdade de Engenharia de
Ilha Solteira – Unesp como parte dos
requisitos para obtenção do título de
Bacharel em Engenharia Elétrica.

Orientador
Alexandre César Rodrigues da Silva

FICHA CATALOGRÁFICA

Desenvolvido pelo Serviço Técnico de Biblioteca e Documentação

B271s Barreiros, João Vitor Frois.
Sistema embarcado para coleta e transmissão de dados via rede GPRS / João Vitor Frois Barreiros. -- Ilha Solteira: [s.n.], 2024
65 f. : il.

Trabalho de conclusão de curso (Graduação em engenharia Elétrica) -
Universidade Estadual Paulista (UNESP), Faculdade de Engenharia, Ilha Solteira,
2024

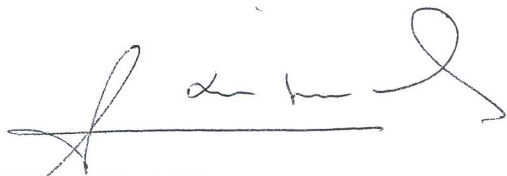
Orientador: Alexandre César Rodrigues da Silva

Inclui bibliografia

1. Internet das coisas. 2. Hardware livre. 3. Esp32. 4. Redes móveis. 5.
Dashboard. 6. Node-red.

ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO

Aos vinte e um dias do mês de junho do ano de dois mil e vinte e quatro, o discente João Vitor Frois da Costa Barreiros, matriculado sob o nº 172054681, tendo como banca examinadora o seu orientador, o *Prof. Dr. Alexandre César Rodrigues da Silva*, o *Eng. Eletricista Bruno Hiromoto Oikawa* e o *Dr. Lucas do Carmo Yamaguti*, apresentou o Trabalho de Graduação intitulado "SISTEMA EMBARCADO PARA COLETA E TRANSMISSÃO DE DADOS VIA REDE GPRS". Obtendo a nota 9,5 (NOVE E MEIO) e conceito APROVADO.



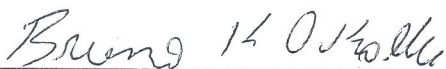
Prof. Dr. Alexandre César Rodrigues da Silva

- Orientador -



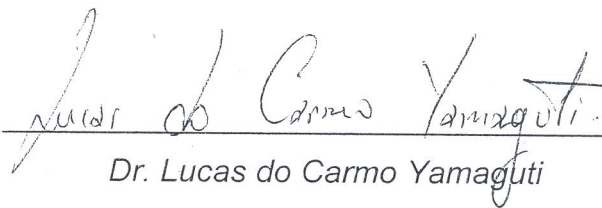
João Vitor Frois da Costa Barreiros

- Discente -



Eng. Eletricista Bruno Hiromoto Oikawa

- Membro da Banca -



Dr. Lucas do Carmo Yamaguti

- Membro da Banca -

RESUMO

O monitoramento de dados a distância aliado ao uso de veículos aéreos não tripulados tem ganhado cada vez mais espaço no mercado aeronáutico e também na agricultura, representando uma maneira simples e eficiente de cobrir grandes áreas para estudo de topologia e de condições atmosféricas. Tudo isso aliado ao uso de redes móveis, que tem se expandido ao longo dos anos desde a chegada da era da informação, onde a cobertura de sinal e sua qualidade atualmente permite uma rápida transferência de dados com amplo alcance. Por esta razão, o presente trabalho apresenta os conceitos de *Internet* das coisas, isto é, objetos físicos ligados a sensores e um *software* com objetivo de conectar os dados à *internet*. Então, é utilizado a plataforma de *hardware* livre ESP32, acoplado com o modem A7608 da SIMCom que permite a comunicação por dados móveis e sensores para coleta de temperatura, umidade e pressão absoluta. O trabalho foi desenvolvido em linguagem C e Javascript e usando compilador Arduino IDE. A exibição dos dados é feita em um *dashboard* produzido na ferramenta de fluxo de dados Node-Red, utilizando o protocolo de comunicação MQTT, que permite uma transferência de dados otimizada e sem perdas de pacotes. É possível visualizar os dados de qualquer dispositivo, acessando por meio do IP e porta fornecido para o acesso. O resultado é um sistema completo de monitoramento de dados em tempo real que permite a sua visualização de qualquer dispositivo.

Palavras-chave: Internet das coisas. *Hardware* livre. ESP32. Redes móveis. *Dashboard*. Node-Red.

ABSTRACT

The remote data monitoring allied with the using of unmanned aerial vehicles has been gaining space in the aeronautics market but also in the agriculture, representing a simple and efficient method to cover large areas for topology study and atmospheric conditions. All this combined with the use of mobile network that has expand through the years since the beginning of the information era, where the coverage signal and its quality signal currently allow a quickly data transfer with wide range. Therefore, the present work provides a concept of Internet of Things which is physical objects associated with sensors and a software with aim to connect data to the internet. Then, it's used an open hardware ESP32 attached with the modem A7608 of SIMCom, that permits a communication via mobile data and sensors for temperature, humidity and absolute pression measurement. Developed in C and Javascript programming language, using an Arduino IDE compiler. The data presentation is done in a dashboard made by a tool of data flow Node-Red, using a communication protocol MQTT that permits an optimized and non-losses/lossless data packet transfer. It is possible to visualize data from any device, accessing via IP and port provided for access. The result is a complete system of data monitoring in real time allowing visualization of any device.

Keywords: Internet of Things. Open Hardware. ESP32. Mobile Network. Dashboard. Node-Red.

LISTA DE FIGURAS

Figura 1 - Módulo Xbee Transmissor e Receptor.....	20
Figura 2 - Subdivisão do Espectro Eletromagnético.....	26
Figura 3 - Evolução da Tecnologia de Comunicação.....	27
Figura 4 - ESP32 TTGo T-SIMA7608SA.....	32
Figura 5 - Sensor de Temperatura e Umidade DHT22.....	34
Figura 6 - Circuito típico com uso de resistor de pull-up.....	34
Figura 7 - Sensor de Pressão Atmosférica BMP280.....	35
Figura 8 - Diagrama Funcional da Telemetria.....	37
Figura 9 - Local do Arquivo e Configuração do Mosquitto.....	39
Figura 10 - Prompt de Comando Node-Red.....	40
Figura 11 - Interface Web do Fluxo Node-Red.....	40
Figura 12 - Dashboard Node-Red.....	41
Figura 13 – Programação do Nó Densidade do Ar.....	42
Figura 14 - Diagrama Esquemático Teste DHT22.....	43
Figura 15 - Montagem do Circuito para Sensor DHT22.....	44
Figura 16 - Resultado da Leitura Sensor de Temperatura e Umidade.....	44
Figura 17 – Termohigrômetro Datalogger ICEL.....	45
Figura 18 - Diagrama Esquemático Teste BMP280.....	47
Figura 19 - Montagem do Circuito para Sensor BMP280.....	47
Figura 20 - Resultados do Monitor Serial para o Sensor BMP280.....	48
Figura 21 - Resposta do Comando "AT+CPIN?".....	49
Figura 22 - Intensidade do Sinal do Projeto.....	50
Figura 23 - Execução do Comando "AT+COPS=?" e "AT+COPS?".....	51
Figura 24 - Diagrama Esquemático Circuito Final.....	52
Figura 25 - Montagem do Sistema Completo.....	52
Figura 26 - Aba Dashboard “Tempo Real”.....	53
Figura 27 – Aba Dashboard “Variação”.....	53
Figura 28 - Tela de Debug.....	54
Figura 29 - Dashboard em Diferentes Dispositivos.....	54

LISTA DE SIGLAS E ABREVIATURAS

ABNT	Associação Brasileira de Normas Técnicas
AM	Amplitude Modulada
AMPS	<i>Advanced Mobile Phone System</i>
CDMA	<i>Code Division Multiple Access</i>
EDGE	<i>Enhanced Data Rates For GSM Evolution</i>
FDMA	<i>Frequency Division Multiple Access</i>
FM	Frequência Modulada
GPS	<i>Global positioning system</i>
GSM	<i>Global System for Mobile</i>
IoT	<i>Internet of Things</i>
I2C	<i>Inter-Integrated Circuit</i>
LTE	<i>Long Term Evolution</i>
MIMO	<i>Multiple Input Multiple Output</i>
OFDM	<i>Orthogonal Frequency Division Multiple Access</i>
SMS	<i>Short Message Service</i>
STBD	Serviço Técnico de Biblioteca e Documentação
TDMA	<i>Time Division Multiple Access</i>
UMTS	<i>Universal Mobile Telecommunication System</i>
Unesp	Universidade Estadual Paulista
VANT	Veículo Aéreo Não Tripulado
WCDMA	<i>Wide Band Code Division Multiple Access</i>

SUMÁRIO

1. INTRODUÇÃO	18
1.1 Motivação	20
1.2 Objetivos.....	21
2. REVISÃO DE LITERATURA	22
3. A ERA DA COMUNICAÇÃO SEM FIO	24
3.1 História da Radiofrequência.....	24
3.2 Propagação de Ondas Eletromagnéticas: Canais sem fio	25
3.3 Tecnologias de redes de comunicação sem fio	27
3.3.1 FDMA – <i>Frequency Division Multiple Access</i>	28
3.3.2 TDMA – <i>Time Division Multiple Access</i>	28
3.3.3 CDMA – <i>Code Division Multiple Access</i>	29
3.3.4 OFDMA – <i>Orthogonal Frequency Division Multiple Access</i>	30
3.4 IoT – <i>Internet of Things</i>	30
4. METODOLOGIA	31
4.1 Materiais Utilizados.....	31
4.1.1 Placa de desenvolvimento	31
4.1.2 Sensor de umidade	33
4.1.3 Sensor de pressão atmosférica.....	35
4.1.4 Servidor local	35
4.2 Diagrama Funcional.....	36
4.3 Montagem do Sistema de Recepção e Apresentação.....	37
4.3.1 Programação <i>Arduino IDE</i>	37
4.3.2 Dashboard <i>Node-Red</i>	38
5. RESULTADOS E DISCUSSÕES	43
5.1 Testes e calibrações dos sensores.....	43
5.1.1 Sensor de temperatura e umidade relativa– DHT22	43
5.1.2 Sensor de Pressão - BMP280.....	46
5.2 Testes de qualidade de sinal e envio de pacotes	48
5.2.1 AT+CPIN.....	48
5.2.2 AT+CSQ.....	49
5.2.3 AT+COPS	51
5.3 Funcionamento do sistema e Dashboard	51
6. CONCLUSÃO	55
REFERÊNCIAS.....	57
APÊNDICE A – Programação do Software para Coleta de Dados	62
ANEXO A – Programação testes comandos AT	67

ANEXO B – Cálculo da densidade do ar	71
--	----

1. INTRODUÇÃO

Revolução industrial, um termo não só utilizado para caracterizar a o movimento de substituição de métodos produtivos da manufatura por máquinas na segunda metade do século XVIII, como também é usado para demarcar períodos da história de grandes viradas tecnológicas (SOUSA, 2024). Cada um desses processos trouxe mudanças e criações de novos setores na economia mundial, além de impactar profundamente as relações sociais e políticas.

A primeira revolução industrial trouxe a mecanização dos meios de produção começando pela industrial têxtil e se estendeu para setores como metalurgia, transportes, agricultura e pecuária, caracterizando um aumento de produção e lucro para os industriários, como também serviu de estopim para o grande êxodo rural e discussões sobre a precarização do trabalho (BEZERRA, 2024).

Na segunda revolução industrial, iniciada a partir do século XIX, houve uma constante inovação dos produtos e criação de processos industriais como o Fordismo e o Taylorismo, pode ser notado cada vez mais o investimento em pesquisa e desenvolvimento para essa finalidade (BEZERRA, 2024).

A terceira revolução industrial, iniciada por volta dos anos de 1950, foi marcada pelo início do desenvolvimento da informática e automação das indústrias, também conhecida como "Revolução Técnico-Científica-Informacional", onde foi empregado tecnologias, principalmente da eletrônica, nos produtos e também os primeiros passos para a telecomunicação digital como é conhecido hoje (BEZERRA, 2024).

Atualmente o mundo vive a Quarta revolução industrial, que foca na informação para melhoria e aperfeiçoamento de processos. Iniciado em meados de 2010, essa transformação foi acompanhada de um uso cada vez maior de dispositivos conectados em rede, utilizando a internet criada no século anterior como meio de tráfego para diversas informações. Tais informações são desde páginas *web*, vídeos, fotos e até mesmo conversas entre pessoas que possuam computadores ou *smartphones* (SPAGNA, 2024).

Com isso, métodos produtivos antes analógicos e que ainda tinha alguma dependência do ser humano para operar, agora estão sendo aprimorados com sistemas capazes de coletar, analisar e tomar decisões por si. A telemetria, que significa a arte de medir coisas, vem para atender o anseio de setores produtivos

industrial e energético que buscam formas de coletar dados como forma de melhorar processos e aumentar produtividade. Um exemplo das vantagens ao empregar sensoriamento remoto é o uso na agricultura, onde é possível coletar dados atmosféricos como temperatura, umidade relativa, precipitação e umidade do solo para ajudar na tomada de decisões e aumentar a produtividade (TEBALDI, 2019).

A comunicação dos dados de telemetria depende bastante da aplicação e também dos meios disponíveis, uma forma bastante confiável é por meio de cabos de rede que interligam sensores à sistemas informatizados, podendo ser analisados ou ligados a alguma automação dependendo do processo. Em casos onde essa coleta é feita em lugares isolados, ou cujo o acesso físico é impossibilitado, são utilizados a comunicação por radiofrequência, onde é necessário componentes para transmitir e recepcionar tais dados.

A transmissão de dados a distância por meio de radiofrequência, apesar de bastante promissora para uso em telemetria, acaba apresentando diversas limitações como alcance, taxa de transferência e interferência, que acaba dependendo do aspecto que o projetista desejar focar, onde a melhora de um aspecto representa uma piora em outro. Exemplo, ao optar por um alcance maior a frequência escolhida, por sua vez, deve ser baixa, o que acaba representando uma maior probabilidade de problemas com interferências devido a quantidade de dispositivos operando na mesma faixa, além de uma baixa taxa de transferência. Frequências maiores possuem a vantagem de entregar dados em maior taxa de transferência e não ser tão suscetível a interferências, em contraponto há uma diminuição do alcance. Esse assunto será explorado de forma mais aprofundada na seção 3.

Em relação a aplicação, o monitoramento pode ser feito de diversas maneiras de acordo com os objetivos a serem alcançados. O trabalho se inspira na no uso em veículos aéreos não tripulados, também conhecidos como VANT, pois é um exemplo de onde o envio de dados de telemetria é totalmente feito a distância sem uso de cabos. Atualmente é discutido diversos possíveis usos para esses dispositivos, que variam desde a pequenas aeronaves até drones, um deles é na agricultura de precisão para condições atmosféricas e até mesmo estudo da topografia, que será discutido ao longo da seção 2. Outro uso bastante interessante é a entrega de encomendas por drones, uma aplicação que tem funcionado na China e Reino Unido desde 2015, o que demanda também um controle sobre

geolocalização do VANT a todo momento e a longas distâncias, onde tecnologias como a telefonia móvel dentro de perímetros urbanos podem facilmente fornecer (IFOOD, 2022).

1.1 Motivação

A principal inspiração para o desenvolvimento do trabalho é o uso aplicado em VANT, que combina a possibilidade de um monitoramento aéreo com tecnologias com uma transferência de dados otimizada via telefonia. A telemetria dentro de qualquer projeto também é um elemento chave em diversos setores, tornando as escolhas de projeto um ponto crucial para o sucesso de um sistema. Atualmente são utilizados transmissores e receptores que trabalham na faixa de 2,4GHz e que possuem um alcance máximo de 1,2 km em ambiente aberto, um exemplo é o módulo Xbee utilizados para Telemetria de VANT's, conforme Figura 1.

Figura 1 - Módulo Xbee Transmissor e Receptor.



Fonte: (VIKACONTROLS, 2024).

Na maior parte das aplicações esse valor é o suficiente para cobrir uma transmissão de dados de forma remota de forma confiável, porém existem aplicações em que esse sistema pode apresentar dificuldades, principalmente se essa demanda distâncias maiores ou que esteja localizado em perímetros urbanos, onde há obstáculos como construções e uma maior gama de dispositivos que operam na mesma frequência. Um exemplo é na agricultura, onde propriedades possuem grande extensão de terras produtivas, como por exemplo o grupo Amaggi com 258 mil hectares, sendo necessário receptores e transmissores mais potentes (CANAL RURAL, 2021).

Com isso a solução sugerida no trabalho é, ao invés de utilizar transmissor e receptor de tecnologia de radiofrequência, utilizar a transmissão via rede *GPRS* dado que existe um grande investimento de estrutura e torres de transmissão que possibilita, além de uma cobertura na maior parte do território do estado de São Paulo, uma qualidade e estabilidade do sinal. Dado que aviões não tripulados tendem a percorrer grandes distâncias, é possível que receptores e transmissores locais não sejam o suficiente para cobrir toda extensão, o que torna ainda mais atrativo alternativas utilizando telefonia móvel.

1.2 Objetivos

O objetivo deste trabalho de graduação é o desenvolvimento de um sistema de telemetria com transmissão via rádio frequência para redes móveis, fazendo a coleta de dados atmosféricos e de condições de voo por meio de sensores, utilizando o microcontrolador *ESP32* em testes de bancada.

2. REVISÃO DE LITERATURA

Na seção Revisão de Literatura será abordado dissertações publicadas a respeito da comunicação sem fio de VANT, sistemas de telemetrias e transmissão via dados móveis de telefonia.

Em Caue Ferreira. (Ferreira, 2019), descreve a utilização da telemetria para aquisição de dados para VANT, onde foi coletado as deflexões relativas das superfícies de comando como profundor, aileron e posicionamento geográfico. Foi utilizado sensor de posição SS10693, módulo GPS NEO-6M, sensor diferencial de pressão MPXV7002 em conjunto com tubo de pitot e o microcontrolador ATMEGA 2560, que assim como os demais projetos funciona com o Arduino IDE e programação em linguagem C. O diferencial desse projeto é a forma com que foi enviado os dados, no caso, através de um conjunto de antenas transmissor e receptor que transmitem a uma frequência de 915 MHz. Nesse sistema os dados são coletados, convertidos e enviados para o computador que fará o pós-processamento em tempo real em software desenvolvido utilizando o software *Scilab*.

Em Vinícius Luiz de Souza (Souza, 2015), apresenta um estudo mencionando um comparativo com sistemas proprietários de telemetria para VANT de empresas como Eagle Tree System e FrSky, que faz menção a principal dificuldade que é poucas possibilidades de modificação e adaptação ao uso, mesmo utilizando sensores de baixo custo e potência. Como microcontrolador utilizou ATmega328PU que também utiliza da programação via Arduino IDE e linguagem C, os dados coletados são de Tensão, Corrente, Velocidade, Aceleração e Posição Angular, que utilizam um sensor de corrente por efeito *hall* e um sensor de pressão diferencial, no trabalho não foi feito menção do sensor para coleta da aceleração e da posição angular, mas supõe-se que foi feito por meio de acelerômetro posicionado nas superfícies de controle. Em relação a forma de transmissão de dados, o mesmo foi via radiofrequência com a utilização do módulo XBee.

Em Bernardo Serpa Borges (Borges, 2014), a proposta foi a construção de um sistema de telemetria baseado na transmissão via redes móveis, no caso o sinal compreende a faixa de frequência da rede GSM, que compreende uma banda de 850 MHz, 900 MHz, 1800 MHz e 1900 MHz, o famoso *QuadBand*. O autor escolheu desenvolver o sistema do zero, incluindo a construção da própria antena para envio

e recepção de dados. A recepção desses dados se dá por meio de mensagens por SMS e tratamento utilizando o software Excel, já os dados coletados são de pressão e temperatura do ambiente. Para a programação do sistema não foi mencionado uma linguagem, porém o modem foi programado utilizando comandos AT.

Em Oni Reasilvia de Almeida Oliveira Sichonany (Sichonany, 2012) detalha um sistema de telemetria que faz o uso de transmissão via rede GSM e radiofrequência com módulo XBee, no estudo esse sistema está acoplado a colhedora automotriz utilizada na agricultura. Os dados colhidos são de combustível, velocidade de operação e largura de trabalho da máquina, os sensores utilizados consistem de um fluxômetro, potenciômetro analógico e um módulo GPS. O objetivo do estudo é produzir um sistema de coleta de dados consistente e com fluxo de dados constante, onde a transmissão GSM e o por radiofrequência assumem na ausência de um ou do outro, ou seja, consiste em um sistema com redundâncias para evitar perda de dados. O diferencial desse artigo é o fato de comparar as duas tecnologias onde o módulo XBee fez a maior parte da aquisição de dados enquanto a transmissão via GSM dependia da cobertura do sinal o que não foi satisfatório em 3 dos 4 dias de teste, desempenhando muito abaixo do esperado com 256 registros de 2749 coletados em comparação com o módulo XBee.

Em Getúlio Teruo Tateoki (2007), se trata de um estudo no qual é desenvolvido um sistema de monitoramento de dados via internet, o diferencial desse trabalho é o fato dos dados serem enviados para uma página web, o que vai de encontro com o objetivo dessa dissertação. O autor escolheu o microcontrolador PIC16f877A, modem TC45 e os dados coletados são de temperatura onde foi utilizado o LM35. O sistema então consiste em uma coleta de dados feita pelo sensor em conjunto com o microcontrolador, após o tratamento esses dados são enviados através do modem TC45 para a página na *WEB* com um *script* desenvolvido em linguagem PHP. O armazenamento de dados é feito utilizando um bando de dados, no caso o MySQL.

3. A ERA DA COMUNICAÇÃO SEM FIO

Neste capítulo apresenta-se o conhecimento teórico necessário para a compreensão da tecnologia de comunicação sem fio, abordando desde seu surgimento até a evolução dos canais de comunicação via sinais modulados como é conhecido atualmente.

3.1 História da Radiofrequência

Durante o século XIX, Michael Faraday e James Clerk Maxwell fizeram diversas descobertas científicas que inauguraram o estudo em radiofrequência, que foi no campo do eletromagnetismo e a criação de equações que o descrevem, além de experimentos que o comprovam realizados por Heinrich Rudolf Hertz. Essas equações mostram que um campo magnético variável funciona como fonte de campo elétrico e vice-versa, formando uma onda que se propaga através do espaço. Tais descobertas possibilitam diversas aplicações como forma de transmissão em emissoras de TV, micro-ondas, aparelhos de raio-X, radares e também para comunicação com em torres de telefonia celular (HWANG; GONZALEZ; MONTEIRO, 2013).

Em 1899 é criado o primeiro sistema de telégrafos sem fios por Guglielmo Marconi, sendo essa transmissão no Canal da Mancha. Entretanto nessa época a voz ainda não era transmitida por meio desse princípio, somente em 1906 que a voz apareceu com experimentos de Lee de Forest, sendo constantemente aprimorada ao passar dos anos (THIAGO PERUCH, 2024).

No mundo as transmissões de rádio começaram a ser cada vez mais frequentes a partir da década de 1920 com músicas, conversas e até aulas. No Brasil o rádio chegou oficialmente com a comemoração do centenário da independência do Brasil em 7 de Setembro de 1922 (THIAGO PERUCH, 2024).

3.2 Propagação de Ondas Eletromagnéticas: Canais sem fio

A propagação de ondas, observando o caso de canais sem fio, por natureza viajam em linha reta assim como a luz visível. Sendo assim, é necessário maneiras para se defletir tais ondas para aumentar o alcance, caso contrário a propagação será limitada a aproximadamente 48 Km, dependendo do terreno e altura das antenas, explicando o fato de torres de transmissão serem posicionadas em altos de colinas, montanhas ou através do tamanho das mesmas.

A eficiência de uma transmissão depende da frequência do sinal a ser transmitido. A conta para dimensionamento físico de antenas por exemplo exigem que tenham pelo menos 1/10 do comprimento de onda λ do sinal.

$$\lambda = \frac{c}{f} \text{ [m/s]} \quad (1)$$

$c = \text{constante de propagação da luz no vácuo} = 3 * 10^8 \text{ [m/s]}$

$f = \text{frequência [m/s]}$

Sendo assim, a transmissão de sinais de áudio, que contém componentes de frequências baixas como 100Hz exigiria um comprimento de antena impraticável de 3 Km. No entanto é possível driblar esse problema modulando esse sinal para uma portadora com frequência muito maior, um exemplo é uma modulação a 100 MHz, que resulta em um comprimento de onda de 3 m e antenas de apenas 30 cm (KITANO, 2023).

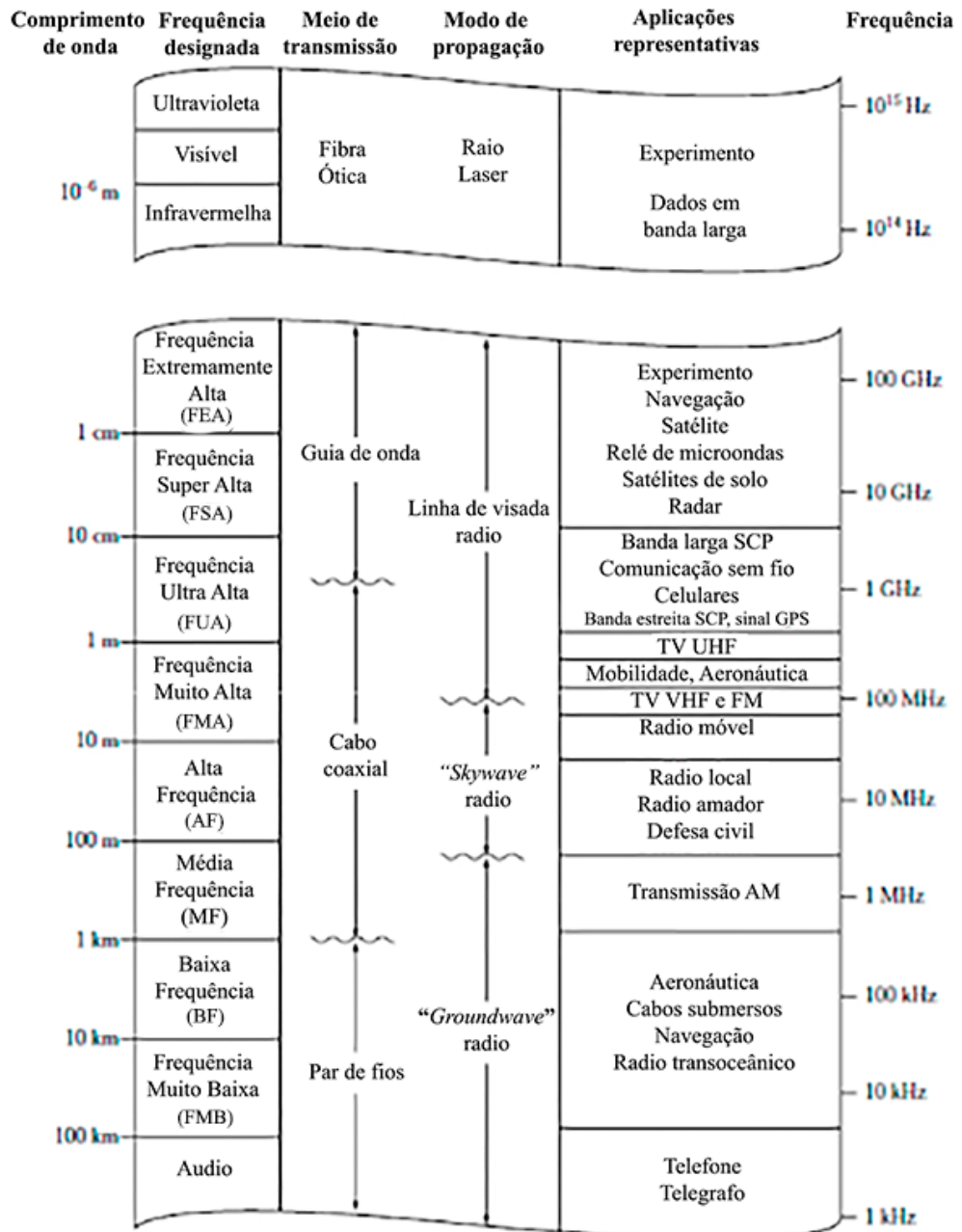
Com isso, se tem diversas frequências de portadoras, que cumprem objetivos distintos dependendo do uso, conforme visto na Figura 2.

A modulação AM é uma forma de transportar uma informação de frequência mais baixa, como no caso de sinais de áudio, em uma onda de frequência mais alta, também chamada de portadora. A principal característica desta modulação é que a frequência e a fase da onda não são alteradas, porém a amplitude altera ao longo do tempo, cujo sua envoltória (envelope) tem a mesma forma do sinal original (KITANO, 2023).

A frequência modulada para rádios AM é bastante inferior em relação à modulação FM, o que permite um alcance muito maior, por outro lado está mais suscetível a ruído devido à proximidade da frequência dos sinais. A modulação FM tem o mesmo objetivo, porém com a amplitude permanecendo constante, em contraponto a sua frequência é alterada ao longo do tempo. A grande vantagem é a

maior imunidade a ruído, uma capacidade maior de operar em altas frequências e diminuição expressiva do tamanho da antena necessária para envio e recepção de sinal, ao custo de uma diminuição expressiva no seu alcance.

Figura 2 - Subdivisão do Espectro Eletromagnético



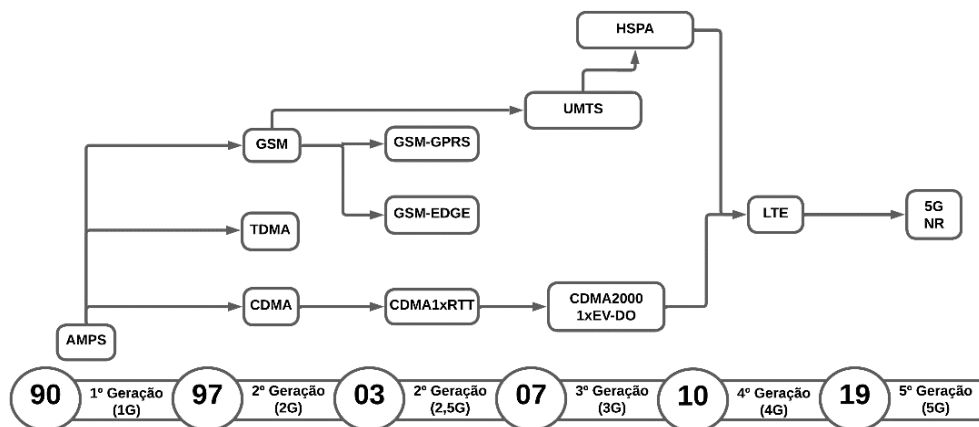
Relé Microondas: combinação de equipamento de recepção, amplificação e transmissão que é usado para coletar, amplificar e retransmitir um sinal de microondas.

SCP - Serviço de Comunicação Pessoal (1850 - 1990 MHz: celulares, mensagens de texto e internet)

3.3 Tecnologias de redes de comunicação sem fio

Aplicando a teoria de propagação de ondas em canais sem fio para comunicação de dispositivos móveis, obtêm-se um importante recurso para transferência de dados móveis em longas distâncias. Tecnologias essas que foram desenvolvidas durante a década de 80 nos Estados Unidos e que vêm se aprimorando desde então. Será apresentado nesta seção as principais ideias por trás das gerações dessa tecnologia: Primeira Geração (1G), Segunda Geração (2G), Terceira Geração (3G) e Quarta Geração (4G).

Figura 3 - Evolução da Tecnologia de Comunicação



Fonte: Próprio Autor.

A primeira geração é caracterizada pelo início da aplicação de sistemas de radiofrequência para mobilidade, essa tecnologia é analógica cujo funcionamento começou com simples transmissores e receptores de rádio associado a um número de telefone. O padrão utilizado é o AMPS, usando a tecnologia FDMA onde a banda é dividida em radiofrequências (modulação FM), e exige uma transmissão em par de canais para transmitir e receber, funcionando na faixa de 800 MHz (SILVA, 2010).

Com o crescimento no uso de telefone celular, também foram surgindo propostas para melhorar tal tecnologia e a migração para sistemas digitais. Com isso, a segunda geração teve início nos anos de 1990 e foi impulsionado pelo avanço de circuitos integrados. As vantagens em comparação com a geração anterior se deram com a maior eficiência espectral, qualidade nas ligações, tráfego

de dados e criptografia. Durante essa geração foram criados os sistemas CDMA e TDMA que fazem parte do padrão GSM (SILVA, 2010).

A terceira geração veio de forma a evoluir os padrões já criados de forma a aumentar principalmente as taxas de transferência e a imunidade à interferência. Foi marcado com a criação de um novo padrão UMTS, que utiliza a interface de rádio WCDMA ou EDGE. Com essa tecnologia foi possível prover serviços que demandam altos consumos de banda como streaming, transferência de grandes arquivos e videoconferência. Também se teve a evolução do sistema CDMA, que passou a possuir alta performance e a operar na faixa de 800 a 1900 MHz. Ambos os sistemas passaram a possuir uma transferência de dados com picos de até 2,4 Mbps nessa geração (SILVA, 2010).

A quarta geração é marcada pela tecnologia LTE, onde é utilizado o sistema MIMO, que consiste no conjunto de duas antenas para recepção e transmissão. Com esse novo sistema, é utilizado novas técnicas para transmissão e recepção como a codificação espaço-tempo e a Multiplexação espacial, além da modulação OFDM que previne interferências.

3.3.1 FDMA – *Frequency Division Multiple Access*

Como o próprio nome já diz, cada usuário acessava um par de portadoras (*Uplink* e *Downlink*), ou seja, frequências específicas dentro a banda que vai de 825 MHz até 845 MHz em alguns casos permanecendo nesse par até o fim da ligação, justamente para que outro usuário possa utilizar o canal. O grande problema é a limitação de usuários simultâneos, pois à medida que canais próximos estejam sendo utilizados por usuários ao mesmo tempo, aumenta as chances de ocorrer uma interferência devido à proximidade dos canais no espectro de frequência, o que seria um problema nos dias atuais dado que grande parte da população realiza chamadas telefônicas e acessam internet com seus dispositivos móveis.

3.3.2 TDMA – *Time Division Multiple Access*

Essa tecnologia veio para resolver o problema inicial de acesso de múltiplos usuários, logo além de dividir o espectro de frequência em canais, como é feito no FDMA. O TDMA também divide o tempo em cada canal em 8 intervalos. Cada

espaço de tempo é ocupado por um usuário, o que impede problemas de interferência e permite usuários simultâneos (TELECO, 2024).

A principal desvantagem que esse sistema apresenta é a distorção de múltiplos caminhos, que ocorre sempre quando a onda de rádio é refletida ou refratada por fatores ambientais, que resulta em uma interferência construtiva ou destrutiva do sinal (WORKNETS, 2024).

3.3.3 CDMA – Code Division Multiple Access

Esse sistema utiliza espalhamento digital como meio de acesso para permitir que vários usuários compartilhem a mesma banda de frequência. Com isso, a informação analógica é convertida em digital em conjunto com seus respectivos bits de controle e depois espalhadas no espectro de outros usuários de forma pseudo aleatória. Nessa tecnologia o transmissor e receptor daquele usuário específico vai trabalhar em cima dos bits de controle combinados, rejeitando todo resto como ruído. A principal vantagem está na escalabilidade onde o usuário pode aproveitar todo o espectro de frequência com uma grande redução na interferência, outra característica interessante é o *downlink* e o *uplink* ter velocidades de transmissão iguais, permitindo que serviços como de streaming, videoconferência e outros que demandam latência muito baixa sejam usados.

Utilizando de uma analogia para explicar a diferença entre as tecnologias, supondo uma grande sala com pessoas conversando, o FDMA representaria uma divisão dessas pessoas em grupos onde cada grupo realizaria a própria conversa simultaneamente, o TDMA representa todas essas pessoas reunidas, porém falando uma de cada vez, enquanto o CDMA seria todas falando ao mesmo tempo, porém em idiomas diferentes (DUSSAUX; KASHIWAGI; CAPITANGO; HEINZELMANN, 2010).

A desvantagem desse tipo de sistema é o problema “perto-longe”, que ocorre devido a estações localizadas perto e longe da estação base, pois nele supõe que os sinais são enviados na mesma potência. Durante a terceira geração, esse sistema evoluiu para WCDMA, que traz consigo além da estrutura anterior, uma maior banda de frequência “*Wideband*”, isso permitiu uma maior velocidade de *download* e *upload*.

3.3.4 OFDMA – *Orthogonal Frequency Division Multiple Access*

É uma tecnologia que provê uma alta eficiência espectral, ou seja, uma maior taxa de transferência de dados. É feito uma multiplexação por divisão de frequências ortogonais, o que leva o nome dessa tecnologia, a banda é dividida em múltiplas portadoras ortogonais chamadas subportadoras, a ortogonalidade é devido a não possuírem sobreposição de frequência, dessa forma não interferindo umas nas outras (COUTINHO, 2008).

Com isso, o princípio básico de funcionamento é a conversão de um fluxo de dados serial de taxa de transmissão elevada em múltiplos subfluxos de transmissão baixa, após a conversão cada subfluxo de dados é modulado em uma subportadora. A principal vantagem disso é que em relação a técnicas que usam uma única portadora, ela pode obter a mesma taxa de transferência com maior resistência a ruídos como a atenuação de altas frequências e a interferência de múltiplos caminhos, devido ao paralelismo de subportadoras de taxas baixas (DUSSAUX; KASHIWAGI; CAPITANGO; HEINZELMANN, 2010).

3.4 IoT – *Internet of Things*

A IoT é uma tendência tecnológica que visa conectar elementos físicos incorporados com sensores ao ambiente Web, esses dispositivos variam de objetos domésticos até máquinas industriais sofisticadas. Com isso, é possível tornar a vida mais confortável, pois com os elementos conectados em rede muitas ações que demandam esforço podem ser feitas de maneira automática, com o mínimo de esforço humano. No ponto de vista econômico, o monitoramento de dados pode levar a uma melhora na gestão dos recursos, isso quando tal mudança não possa ser feita de maneira automática também (ORACLE, 2024).

Uma amostra dessa inovação é o advento dos sensores de baixo custo e potência, a conectividade evoluiu muito com novos protocolos que facilitam a conexão de sensores, a computação na nuvem que permite armazenamento e processamento de informação sem que os consumidores gastem com infraestrutura. Além da estrutura física também se tem a análise de dados que é possibilitada com o *Machine Learning*, onde os dados não somente são coletados como interpretados.

4. METODOLOGIA

A metodologia utilizada para estruturar o projeto se baseia na tecnologia GPRS junto de plataformas para IoT como Arduino IDE e Node-Red para criação de um sistema para leitura de dados atmosféricos, para ser empregado em VANT's. Arduino é uma plataforma programável para prototipagem eletrônica de código aberto, objetivando a criação de ferramentas acessíveis de baixo custo. (ARDUINO, 2021). Já o Node-Red, é uma plataforma de desenvolvimento programável, assim como o Arduino, porém com o foco em unir elementos de hardware em programação visual, se utiliza da linguagem Javascript para criação de projetos (RED, 2024).

Nesta seção será descrito os elementos cruciais para a realização tal como hardware e o processo detalhado de funcionamento do projeto, além de testes de bancada e seus resultados. A intenção final do projeto era testar a coleta de dados em algum VANT como os usados na competição SAE Aerodesign ou drone, porém por fatores externos, como a não disponibilidade de um avião remotamente controlado pronto e os custos altos para aquisição de um drone, os testes se limitaram a bancada apenas.

4.1 Materiais Utilizados

4.1.1 Placa de desenvolvimento

A placa de desenvolvimento escolhida é o ESP32 TTGo T-SIMA7608SA, produzida pela LILYGO, empresa focada em pesquisas e desenvolvimento de tecnologias para IoT, segundo Figura 4.

Essa placa conta com o ESP32 versão WROVER-E, da série de microcontroladores de baixo custo e baixo consumo de energia (EXPRESSIF, 2023), com as seguintes especificações:

- Xtensa® Dual-Core 32-bit LX6
- ROM: 448 KBytes
- RAM: 520 Kbytes
- Flash: 4 MB
- Clock: 80 a 240MHz (Ajustável)
- Tensão de Alimentação: 4,5 à 12,0 VDC (Pino Vin)

- Upload link: 50Mbps (max) e 85,6Kbps (min);
- Download link: 150Mbps (max) e 85,6Kbps (min);
- Suporte a protocolos de comunicação: TCP, IP, IPV4, IPV6, DNS, Multi-PDP, FTP, FTPS, HTTP, HTTP, MQTT;

A característica mais atrativa desse módulo é a capacidade de operar na faixa de frequência do sinal de 5G, que reflete na alta taxa de transferência de dados. Como cada pacote transmitido não excede 50 Bytes, o módulo em geral mais que atende as especificações de projeto no sentido da velocidade de transmissão.

Outro diferencial do módulo é o suporte a bateria 18650, que pode ser comprado em versões com capacidade de até 2200mAh. Já o consumo do módulo como um todo segundo o fabricante é em torno de 170mA no modo ativo. Utilizando a equação (1) para a autonomia da bateria temos que:

$$T = \frac{C}{I_c} [h] \quad [1]$$

Onde:

T: Autonomia-horas

C: Capacidade da bateria (Ah)

I_c: Consumo decorrente (A)

Com o cálculo é constatado que a autonomia teórica do módulo alimentado com bateria externa é de:

$$T = \frac{2,2}{0,170} \cong 13h$$

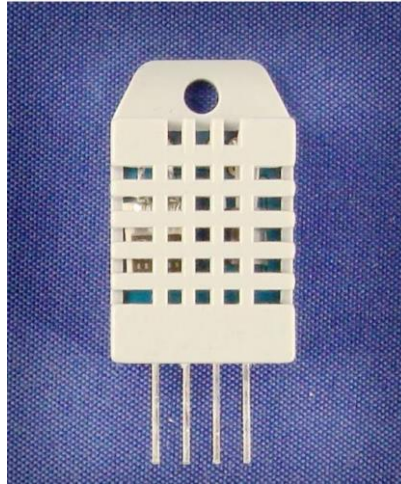
Com isso pode constatar que mesmo considerando possíveis perdas o módulo pode mais que fornecer a autonomia necessária para o projeto, que visa fazer voos de curta duração, de aproximadamente 3min.

4.1.2 Sensor de umidade

DHT22 é um sensor de temperatura, como mostrado na Figura 5, com range de -40 à 80 °C, e umidade de 0 a 100%. Outras características intrínsecas é a alimentação ser de 3,3 V à 6 V e um período de amostragem de aproximadamente 2 segundos (AOSONG, 2024).

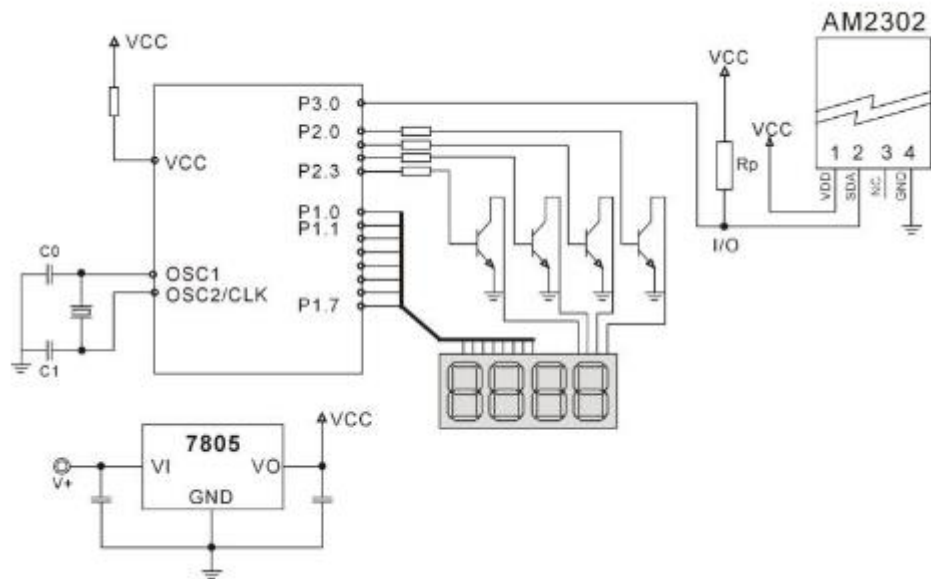
Segundo o datasheet do fabricante, recomenda-se o uso de um resistor de *pull-up* de 5,1 k Ω para casos onde a ligação seja feita com cabos menores que 20m, conforme detalhado na Figura 6 (AOSONG, 2024).

Figura 5 - Sensor de Temperatura e Umidade DHT22



Fonte: (AOSONG, 2024).

Figura 6 - Circuito típico com uso de resistor de pull-up



Fonte: (AOSONG, 2024).

Conforme as condições de utilização apresentadas, verifica-se a necessidade de utilizar essa configuração. Como o módulo utiliza um microcontrolador, a comunicação desse sensor com o ESP32 é feita de forma digital e serial. Os dados referentes a características relativas à precisão do módulo se encontra abaixo:

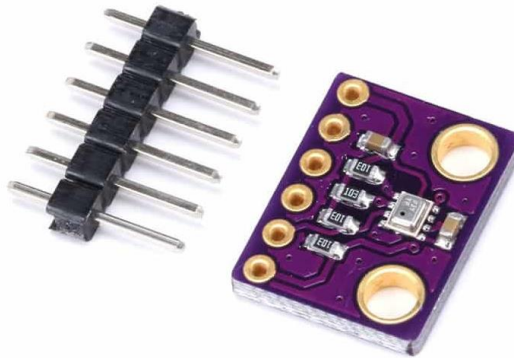
- Faixa de medição de umidade: 0 à 100% UR

- Faixa de medição de temperatura: -40° à $+80^{\circ}\text{C}$
- Corrente: 2,5 mA max durante uso, em *stand by* de 100 μA 150 μA
- Precisão de umidade de medição: $\pm 2,0\%$ UR
Precisão de medição de temperatura: $\pm 0,5^{\circ}\text{C}$
- Resolução: 0,1
- Tempo de resposta: 2s

4.1.3 Sensor de pressão atmosférica

O Sensor BMP280 é um circuito integrado para medição de pressão atmosférica absoluta, normalmente utilizado para aplicações que exigem pequenas dimensões e baixo consumo como Smartphones e módulos GPS (BOSCH, 2024).

Figura 7 - Sensor de Pressão Atmosférica BMP280



Fonte: (BOSCH, 2024).

O hardware conta com uma resolução de 0,16 Pa, e um range de 0 até 20000 hPa, considerando que a pressão atmosférica tende a diminuir à medida que a altura aumenta em relação ao nível do mar, sendo aproximadamente 1013,25 hPa, o sensor acaba atendendo com alta precisão, de aproximadamente $\pm 0,12$ hPa, os requisitos de funcionamento. Em relação às características elétricas principais se destacam a tensão de operação de 1,20 à 1,71 V, o que não é um problema pois o módulo conta com um regulador de tensão para aceitar a tensão da placa de desenvolvimento, que é 3,3 V, o mesmo funciona também com interface I2C sendo possível outros sensores compartilharem o mesmo barramento, possui filtros digitais e por fim uma taxa de amostragem de 6 μs .

4.1.4 Servidor local

Como o projeto não demanda poder computacional, tão pouco memória, o servidor para atuar como intermediário poderia ser desde um microprocessador, como é o caso das plataformas Raspberry PI, ou até mesmo um computador pessoal. Como a intenção do sistema é ser de fácil implementação, se fez uso do computador pessoal do autor, desde que o mesmo esteja constantemente conectado à internet para receber os dados de telemetria.

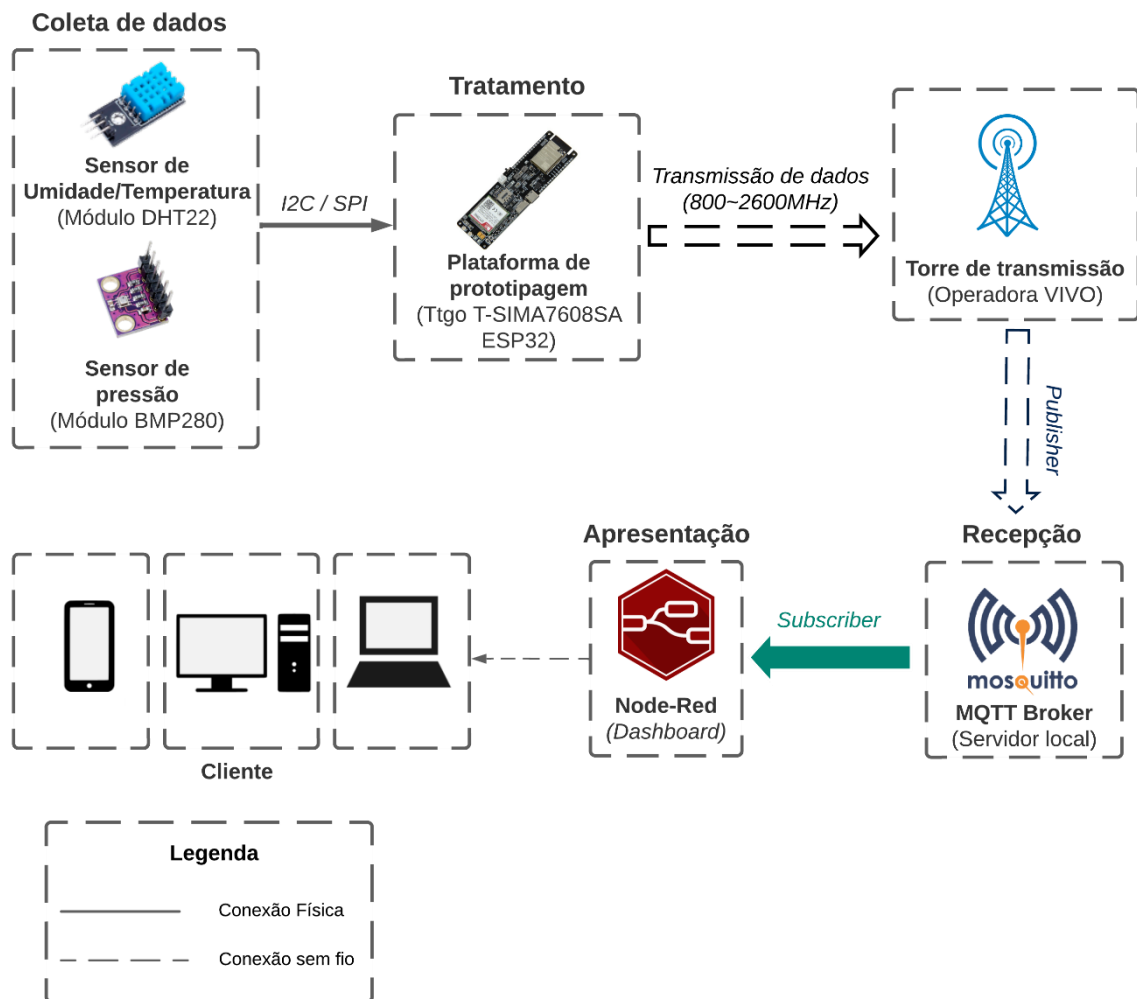
Para o servidor servir de intermediário da comunicação MQTT, vinda da placa de desenvolvimento TTGo T-SIMA7608, foi necessário a instalação do *Broker* Mosquitto, um software *open-source* desenvolvido pela Eclipse Foundation que traz dentre as vantagens a portabilidade e o baixo consumo de recursos computacionais.

Outro recurso utilizado é Node-Red, uma ferramenta de programação *Open-source* baseado no conceito de *Low Code*, o que permite através de uma interface gráfica programar fluxo de interação dos dados, que nesse caso, vai ser para elementos gráficos tais como o dashboard que será exibido ao acessar a página por meio do IP e porta do Servidor Local. Também é possível programar escrevendo códigos em Javascript dentro dos nós da aplicação no fluxo.

4.2 Diagrama Funcional

O projeto segue o diagrama funcional como mostrado na Figura 8. O processo de começa na fase de coleta de dados no qual são utilizados os sensores apresentados ao longo da seção 3.1, passando para a fase de tratamento utilizando a plataforma de prototipagem TTGo T-SIMA7608 ESP32 responsável por preparar o pacote de dados a ser enviado via protocolo MQTT por meio de um sinal de telefonia móvel, normalmente operando na banda do sinal 3G/4G. O sinal é transmitido para a torre da operadora e redirecionado para o servidor local contendo o *broker* Mosquitto fazendo a recepção, no qual o *Node-Red* vai interceptar fazendo a parte do *subscriber*. Ao fim, todos os computadores e smartphones que possuem o endereço do *Node-Red* podem visualizar os dados.

Figura 8 - Diagrama Funcional da Telemetria



Fonte: Próprio autor.

4.3 Montagem do Sistema de Recepção e Apresentação

Nesse capítulo será detalhado o processo de montagem do sistema de recepção e apresentação do trabalho.

4.3.1 Programação *Arduino IDE*

Utilizando a linguagem C e o compilador *Arduino IDE*, foi criado um software que fará a leitura dos dados coletados em tempo real. Para isso será utilizado bibliotecas disponibilizadas pela *Adafruit*, empresa de Hardware de código aberto, permitindo a criação de funções e objetos que facilitam a programação, e por sua vez a depuração.

Foi também necessária a configuração de uma comunicação serial, que faz a ponte para os dados coletados pelo microcontrolador para o modem. A instrução utilizada é a “*SerialAT.begin()*” onde “SerialAT” é o nome definido para transmissão serial de comandos AT. A instrução completa requer a definição de uma taxa de transferência de bits por segundo (*baud rate*), o número de bits por transmissão e a definição dos pinos de transmissão e recepção. O *baud rate* definido para o projeto foi de 115200 bits por segundo, com 10 bits por transferência (sendo 1 bit de paridade, 8 bits de dados e 1 bit de parada), com pinos de TX e RX sendo as portas 26 e 27 respectivamente.

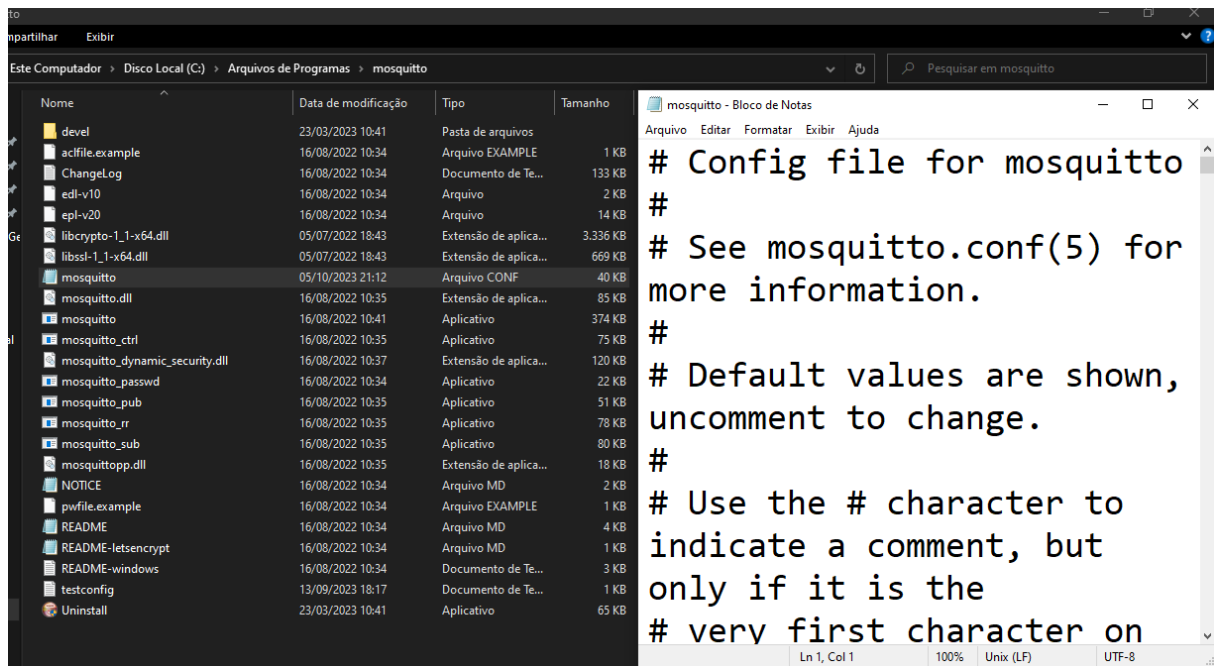
Com a comunicação serial criada, foi necessário a criação de uma função para facilitar o envio de instruções AT para o modem, definida na programação como “*gsm_send_serial*” sendo composta do comando AT mais um tempo de espera (*delay*), esse “*delay*” é de extrema importância para que o modem tenha um tempo adequado para fornecer um retorno do envio de cada comando. Por fim, foi criado também funções responsáveis para o envio de comando de inicialização do modem, definida como “*Init*” e para começar o serviço MQTT, definido como “*getMQTTS*”.

A lógica da programação é, na íntegra, coletar dados da telemetria armazenando-as em variáveis, abrir um tópico MQTT com destino ao broker, informar o tópico no qual será publicado (no caso “Telemetria”), carregar a informação por meio de um comando de “*payload*” e por fim publicar, a frequência de envio de dados será de aproximadamente 5 segundos, de forma a respeitar o tempo necessário para que os sensores *DHT22* e *BMP280* faça a leitura correta de dados. O código completo está disponível no Apêndice A.

4.3.2 Dashboard Node-Red

Primeiro foi feito o download do *Broker Mosquitto* e sua instalação, depois foi executado por meio do Prompt de comando do Windows 10 através da execução na raiz, geralmente quando instalado o mesmo fica em “C:\Program Files\mosquitto”, com isso basta executar o comando “*net start mosquitto*”. O próximo passo é configurar o mosquitto de forma que ele permita que um cliente possa acessar uma porta do sistema para Publicar (*Publisher*) ou Assinar (*Subscriber*). Isso é feito acessando o arquivo *mosquitto.conf* disponível na pasta do mosquitto, a alteração do arquivo é feita por meio de um arquivo de texto conforme Figura 9. Por padrão a porta utilizada é a 1883, porém pode ser também facilmente configurada.

Figura 9 - Local do Arquivo e Configuração do Mosquitto



Fonte: Próprio autor.

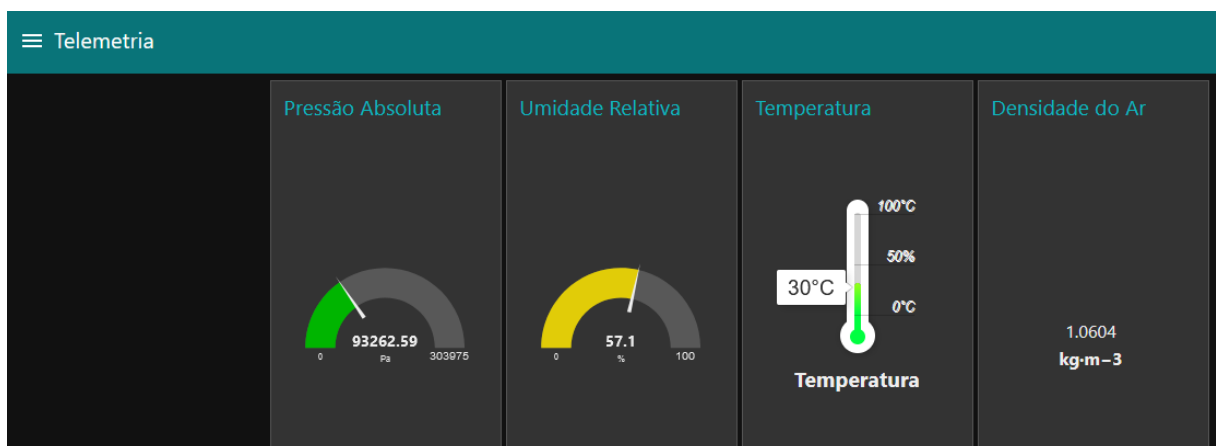
Por motivos de privacidade, no documento não será mencionado as configurações adotadas no projeto, porém para funcionar o protocolo pelo deve-se configurar “*allow_anonymous_true*”, com isso a porta passa ouvir qualquer cliente que publicar na porta configurada para MQTT sem solicitar credenciais (Usuário e Senha). Após feitas as alterações, basta salvar o arquivo “.conf” e novamente executar o comando “net start mosquitto” como administrador no prompt de comando.

Para a instalação do Node-Red, seguiu-se as instruções disponíveis na própria página da Node-Red.org. Para executar apenas basta digitar no *prompt* como administrador o “node-red”. É possível verificar no próprio prompt a execução do programa e a utilização dos arquivos de sistema, conforme Figura 10.

As funções a esquerda são chamadas de nó, com ela foi criado a conexão com o MQTT do próprio servidor local através do nó “*mqtt in*”, com duplo clique foi configurado o IP e Porta do MQTT, o tópico foi definido como “Telemetria”, o mesmo do qual foi definido no sistema de aquisição e envio. Com isso o sistema pode “escutar” a porta do MQTT e resgatar o pacote com o tópico Telemetria.

Para o Dashboard, foi escolhido 3 nós de texto que receberá o objeto respectivo a sua unidade, e um nó medidor que receberá os dados de umidade relativa, devido ao fato do valor ser em porcentagem e facilitar a visualização. Cada uma das variáveis acompanha um gráfico de linha que fará o papel de mostrar a evolução dos parâmetros no tempo como forma de visualizar a variação, ao acessar com o IP e porta do computador através de qualquer dispositivo é possível visualizar a tela conforme Figura 12.

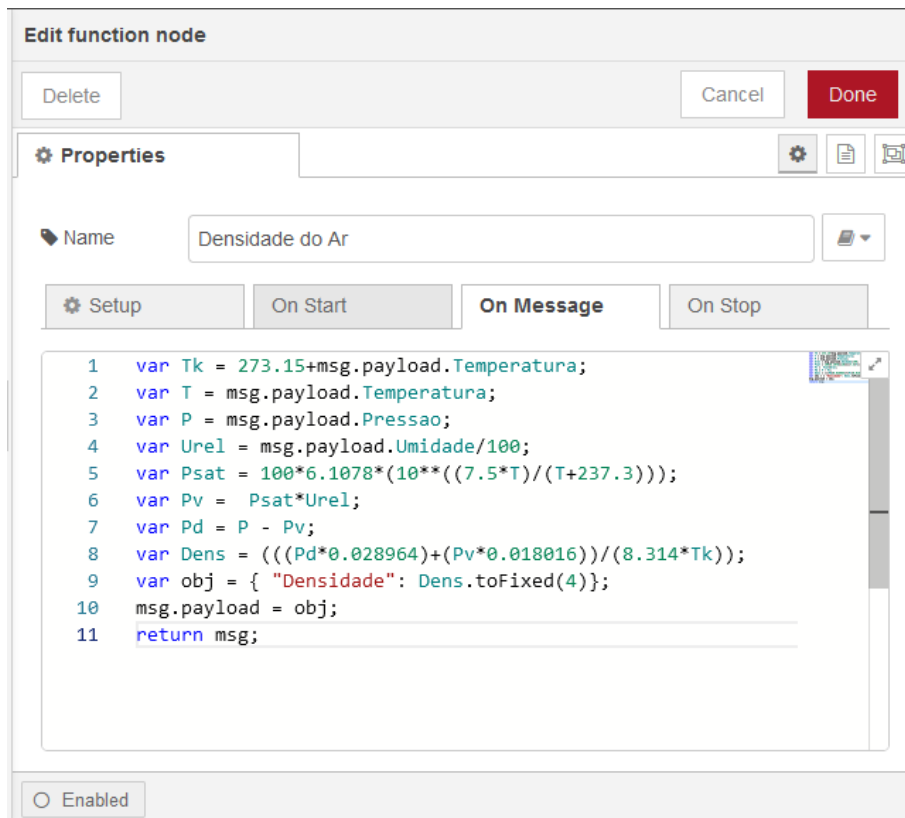
Figura 12 - Dashboard *Node-Red*



Fonte: Próprio Autor.

Um parâmetro que não foi mencionado nas seções anteriores é a densidade do ar. Essa variável é calculada utilizando os dados de pressão absoluta, umidade relativa e temperatura, onde o cálculo está disponível ao longo do Anexo B. A programação deste parâmetro é feita criando um nó “Densidade do Ar”, coletando os parâmetros “Temperatura”, “Pressão” e “Umidade” e realizando o cálculo, como é demonstrada na Figura 13.

Figura 13 – Programação do Nó Densidade do Ar



Fonte: Próprio autor.

5. RESULTADOS E DISCUSSÕES

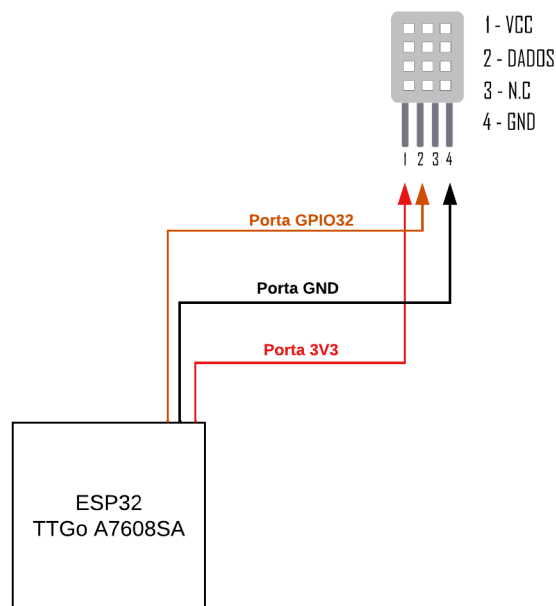
Nesta seção apresenta-se os resultados do desenvolvimento do sistema de telemetria, utilizando os sensores e dispositivos apresentados na seção anterior, iniciando por testes e calibrações necessárias dos componentes individualmente e posteriormente o sistema como um todo funcionando simultaneamente.

5.1 Testes e calibrações dos sensores

5.1.1 Sensor de temperatura e umidade relativa– DHT22

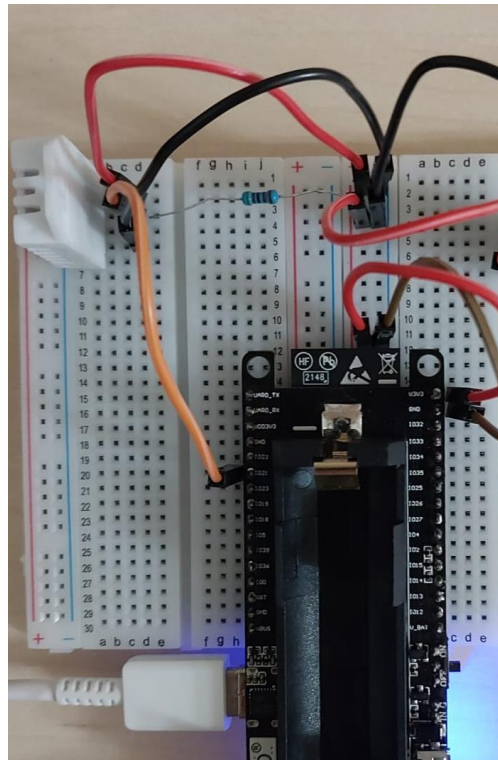
Para o teste do sensor de temperatura e umidade relativa DHT22, foi utilizado uma programação de exemplo disponível na biblioteca “DHT.H”, programação disponível no GitHub em domínio público e no Anexo A. A montagem do protótipo para teste segue o diagrama de conexões da Figura 14, com a montagem feito conforme Figura 15. Os resultados foram coletados via “*Monitor Serial*” do Arduino, conforme Figura 16.

Figura 14 - Diagrama Esquemático Teste DHT22



Fonte: Próprio Autor.

Figura 15 - Montagem do Circuito para Sensor DHT22.



Fonte: Próprio Autor.

Figura 16 - Resultado da Leitura Sensor de Temperatura e Umidade

COM3

14:57:16.435	->	Humidity: 45.10%	Temperature: 31.10°C	87.98°F	Heat index: 31.86°C	89.36°F
14:57:18.465	->	Humidity: 45.10%	Temperature: 31.20°C	88.16°F	Heat index: 32.01°C	89.62°F
14:57:20.435	->	Humidity: 44.80%	Temperature: 31.20°C	88.16°F	Heat index: 31.96°C	89.53°F
14:57:22.477	->	Humidity: 44.70%	Temperature: 31.20°C	88.16°F	Heat index: 31.94°C	89.50°F
14:57:24.458	->	Humidity: 44.60%	Temperature: 31.20°C	88.16°F	Heat index: 31.92°C	89.46°F
14:57:26.456	->	Humidity: 44.60%	Temperature: 31.30°C	88.34°F	Heat index: 32.07°C	89.73°F
14:57:28.471	->	Humidity: 44.70%	Temperature: 31.20°C	88.16°F	Heat index: 31.94°C	89.50°F
14:57:30.486	->	Humidity: 44.30%	Temperature: 31.30°C	88.34°F	Heat index: 32.02°C	89.64°F
14:57:32.480	->	Humidity: 44.40%	Temperature: 31.30°C	88.34°F	Heat index: 32.04°C	89.67°F
14:57:34.504	->	Humidity: 44.50%	Temperature: 31.30°C	88.34°F	Heat index: 32.06°C	89.70°F

Fonte: Próprio Autor.

Como comparação dos resultados foi utilizado o Termo-higrômetro *Datalogger* da marca ICEL com certificado de calibração, possui função de medição e registro de Temperatura, Umidade Relativa e Pressão.

Figura 17 – Termohigrômetro *Datalogger* ICEL.

Fonte: (ICEL, 2024).

Na Tabela 1 é apresentado valores mensurados de Umidade Relativa (%) no período compreendido entre 17h01 até 18h01 do dia 08/04/2024. Foram calculados o desvio, média e erro percentual de cada medida em relação ao HT-4010, utilizado como valor teórico, relação ao DHT22 com os valores experimentais.

Tabela 1 – Amostragem da Umidade Relativa com Sensor DHT22

Horário	Amostragem HT-4010 (%)	Amostragem DHT22 (%)	Diferença entre medidas (%)
17h01	70,9	69,4	1,5
17h06	72,4	70,5	1,9
17h11	76,2	74,4	1,8
17h16	74,6	72,9	1,7
17h21	73,1	71,3	1,8
17h26	72,4	72,7	0,3
17h31	71,7	71,3	0,4
17h36	69,1	68,3	0,8
17h41	68,8	66,9	1,9
17h46	68,8	68,3	0,5
17h51	68,8	68,1	0,7
17h56	69,3	68,5	0,8
18h01	69,6	69,4	0,2
Média	71,2	70,2	1,1
Erro (%)		1,5	

Fonte: Próprio Autor.

Na Tabela 2 é feito o mesmo processo, mas para a medida de temperatura, apresentando o desvio, a média e o erro percentual da comparação entre o teórico e experimental.

Tabela 2 – Amostragem da Temperatura com DHT22

Horário	Amostragem HT-4010 (°C)	Amostragem DHT22 (°C)	Diferença entre medidas (°C)
17h01	25,6	26,0	0,4
17h06	25,4	25,8	0,4
17h11	25	25,3	0,3
17h16	25,1	25,5	0,4
17h21	25,3	25,7	0,4
17h26	25,3	25,2	0,1
17h31	25,5	25,7	0,2
17h36	25,4	25,2	0,2
17h41	25,5	25,5	0,0
17h46	25,3	25,2	0,1
17h51	26,2	25,3	0,9
17h56	25,1	25,3	0,2
18h01	25,1	25,2	0,1
Média	25,4	25,5	0,1
Erro (%)		0,3	

Fonte: Próprio autor

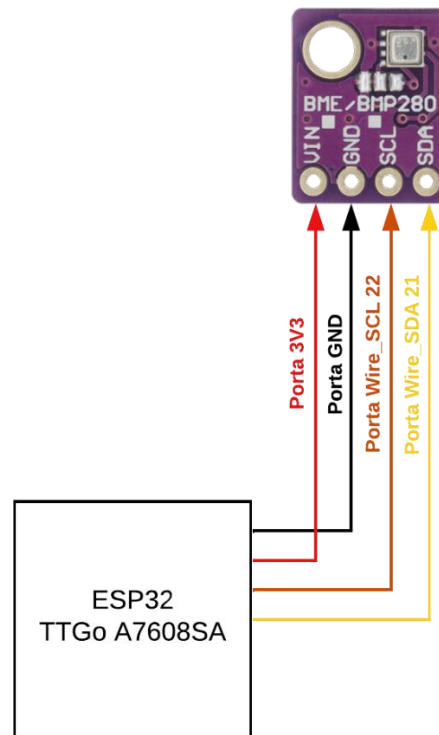
Pode se observar que para Umidade Relativa o desvio do valor teórico em relação ao experimental foi em média 1,1 %, que se encontra dentro da precisão informada pelo fabricante. O mesmo pode-se afirmar da medida da temperatura, cujo o desvio da medida entre a média da amostragem teórica e prática foi de apenas 0,1 °C, que também se encontra dentro da faixa informada. Com isso pode-se inferir que o sensor DHT22 é preciso.

5.1.2 Sensor de Pressão - BMP280

Seguindo a mesma ideia do Sensor DHT22, o código usado para o teste foi um exemplo disponível na biblioteca “Adafruit_BMP280.h”, também disponível em domínio público no GitHub. O objetivo é comparar os resultados das medições do Monitor Serial com o Termo-higromêtro *Datalogger*.

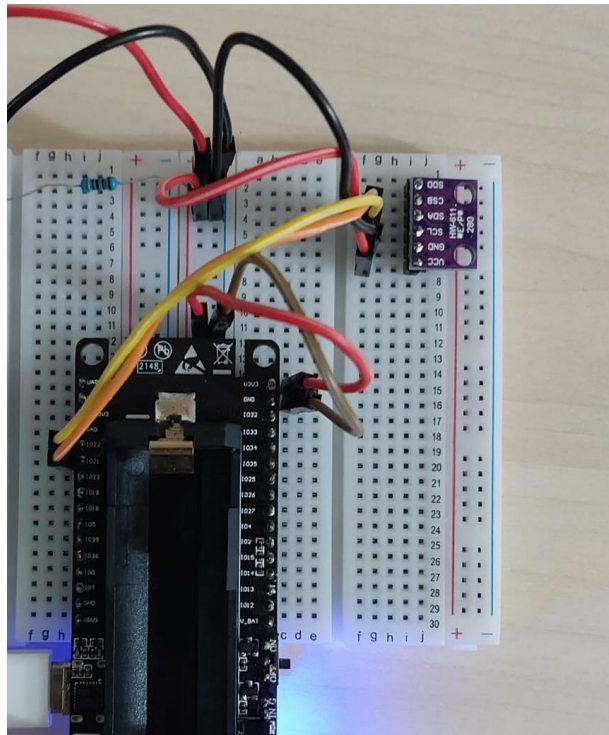
Na Figura 18, é mostrado o diagrama esquemático do circuito a ser testado, sua montagem completa na Figura 19 e a amostragem dos valores via *Monitor Serial* do Arduino IDE na Figura 20. Nesse caso, a medida está sendo feita em atm, cujo o fator de conversão é de 0,000986923 atm para 1 hPa (hectopascal). Foi feito a média de 5 medidas em um intervalo de 5 minutos, apresentado na Tabela 3.

Figura 18 - Diagrama Esquemático Teste BMP280



Fonte: Próprio Autor.

Figura 19 - Montagem do Circuito para Sensor BMP280



Fonte: Próprio Autor.

Figura 20 - Resultados do Monitor Serial para o Sensor BMP280

```

COM3
17:04:46.721 -> Temperature: 30.79 °C - Pressure: 0.92 atm - Altitude: 681.02 m
17:04:49.715 -> Temperature: 30.79 °C - Pressure: 0.92 atm - Altitude: 681.17 m
17:04:50.711 -> Temperature: 30.79 °C - Pressure: 0.92 atm - Altitude: 681.07 m
17:04:51.707 -> Temperature: 30.79 °C - Pressure: 0.92 atm - Altitude: 680.74 m
17:04:52.702 -> Temperature: 30.79 °C - Pressure: 0.92 atm - Altitude: 681.07 m
17:04:53.745 -> Temperature: 30.78 °C - Pressure: 0.92 atm - Altitude: 681.24 m
17:04:54.735 -> Temperature: 30.78 °C - Pressure: 0.92 atm - Altitude: 681.39 m
17:04:55.728 -> Temperature: 30.78 °C - Pressure: 0.92 atm - Altitude: 681.39 m
17:04:56.725 -> Temperature: 30.78 °C - Pressure: 0.92 atm - Altitude: 681.00 m
17:04:57.723 -> Temperature: 30.78 °C - Pressure: 0.92 atm - Altitude: 681.10 m
17:04:58.763 -> Temperature: 30.77 °C - Pressure: 0.92 atm - Altitude: 680.84 m

```

Fonte: Próprio Autor.

Tabela 3 - Resultado de Testes com Sensor BMP280

Horário	Amostragem HT-4010 (hPa)	Amostragem DHT22 (hPa)	Diferença entre medidas (hPa)
17h01	934,2	934,1	0,1
17h06	934,3	934,2	0,1
17h11	934,3	934,2	0,1
17h16	934,3	934,2	0,1
17h21	934,2	934,1	0,1
Média	934,3	934,2	0,1
Erro (%)		0,1	

Fonte: Próprio Autor.

O desvio da média entre os valores experimentais e o teórico nesse caso foi de 0,1 hPa, que assim como no caso do sensor DHT22 está dentro da faixa de precisão informada pelo fabricante, confirmando a precisão do sensor BMP280.

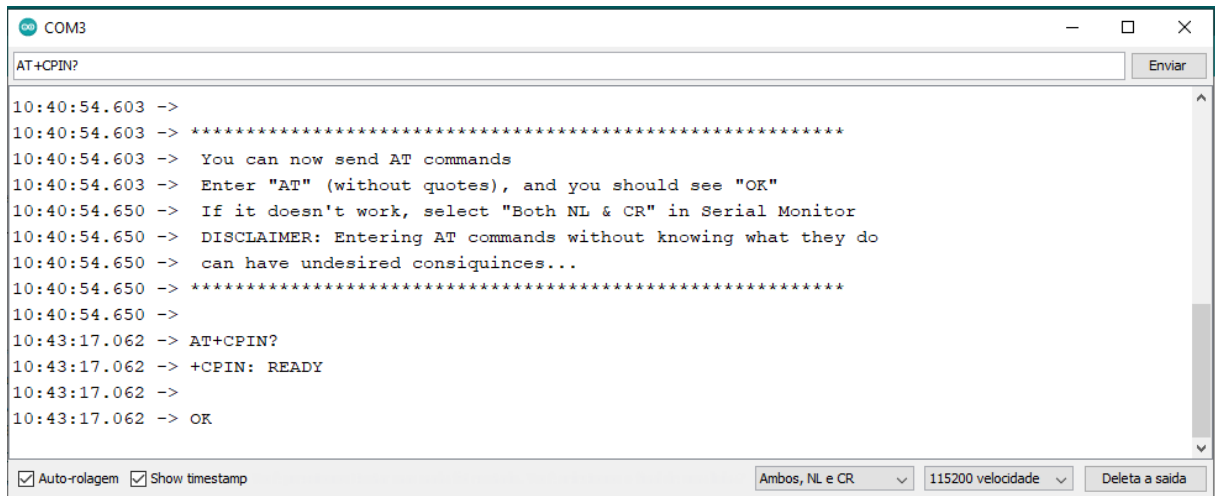
5.2 Testes de qualidade de sinal e envio de pacotes

Para esse teste será utilizado uma rotina que permite os testes de qualidade de sinal, para isso será estabelecido uma comunicação serial com o modem SIM-A7608SA e executar comandos AT, a rotina está disponível no Anexo A.

5.2.1 AT+CPIN

É um comando que permite verificar se é necessário enviar um “PIN code” para o modem utilizar o chip, no caso, da operadora VIVO. Conforme a Figura 21, a resposta do modem é “READY”, que significa segundo o manual da Simcom (2023) que o chip se encontra pronto para uso e sem necessidade de digitar o “PIN code”.

Figura 21 - Resposta do Comando "AT+CPIN?"



Fonte: Próprio autor.

5.2.2 AT+CSQ

É um comando que permite verificar a qualidade do sinal no sentido da potência de transmissão e recepção. Em Simcom (2023), manual descreve por meio da fórmula (2):

$$+CSQ: <rsqi>,<ber> \quad (2)$$

Onde:

<rsqi> Received signal strength indication

<ber> Bit Error Indication

O <rsqi> apresenta parâmetros que representa uma faixa de intensidade em dBm a qual se enquadra o momento da execução do comando, como observado no Tabela 4.

Tabela 4 - Possíveis Resultados e seus Respectivos Valores <rsqi>

Parâmetro <rsqi>	Valores
0	-113 dBm ou menor
1	-111dBm
2 à 30	-109 à -53 dBm
31	-51 dBm ou maior
99	Não detectável

Fonte: (SIMCOM, 2023)

O parâmetro <ber> mede a porcentagem de erro que o sinal pode apresentar, o significado de cada resultado é dado no Tabela 5.

Tabela 5 - Parâmetros e seus Respectivos Valores <ber>

Parâmetro <rsqi>	Valores em %
0	Menor que 0,01
1	0,01 à 0,1
2	0,1 à 0,5
3	0,5 à 1,0%
4	1,0% à 2,0%
5	2,0 à 4,0
6	4,0 a 8,0
7	Maior que 8,0
99	Não conhecido ou indetectável

Fonte: (SIMCOM, 2023)

Ao executar o comando é apresentado o valor mostrado na Figura 22.

Figura 22 - Intensidade do Sinal do Projeto



Fonte: Próprio autor.

A intensidade do sinal do projeto se encontra na faixa de -109 a -53dBm, como o valor do parâmetro é 27 significa que ele se encontra mais próximo de -53dBm do que de -109dBm. Quanto maior o valor, mais próximo de uma intensidade de sinal forte, logo podemos concluir que é um sinal com intensidade muito boa. Em relação a taxa de erro é indicado, segundo a legenda da Tabela 5, que o valor não é conhecido ou indetectável, reforçando a confiabilidade do sinal para transmissão de dados.

5.2.3 AT+COPS

É um comando utilizado para verificar quais são as redes suportadas pelo modem SIMA7608 e qual está sendo utilizada pelo mesmo.

Na Figura 23, é possível ver que a antena consegue identificar diversas operadoras e modos de transmissão com o comando "AT+COPS=?", o modem está configurado para escolher automaticamente a melhor rede (GSM, LTE e entre outros) da operadora do chip, visto ao executar o comando "AT+COPS?".

Figura 23 - Execução do Comando "AT+COPS=?" e "AT+COPS?"

```

COM3
14:42:19.043 -> AT+COPS=?
14:42:19.079 -> ERROR
14:42:24.569 -> at+cops=?
14:42:47.710 -> +COPS: (2, 'VIVO', 'VIVO', '72410', 7), (1, 'VIVO', 'VIVO', '72410', 0), (1, 'VIVO', 'VIVO', '72410', 2), (1, 'Claro BRA', 'Claro', '72405', 1)
14:42:48.041 ->
14:42:48.041 -> OK
14:43:53.271 -> AT+COPS?
14:43:53.307 -> +COPS: 0,2,'72410',7
14:43:53.307 ->
14:43:53.307 -> OK

```

Fonte: Próprio autor.

5.3 Funcionamento do sistema e Dashboard

O esquemático completo do circuito com os sensores mencionados e a placa está contemplado na Figura 24, com a montagem demonstrada na Figura 25. A antena responsável por envio de dados é do padrão LTE que compreende a frequência de 1710 até 2690 MHz.

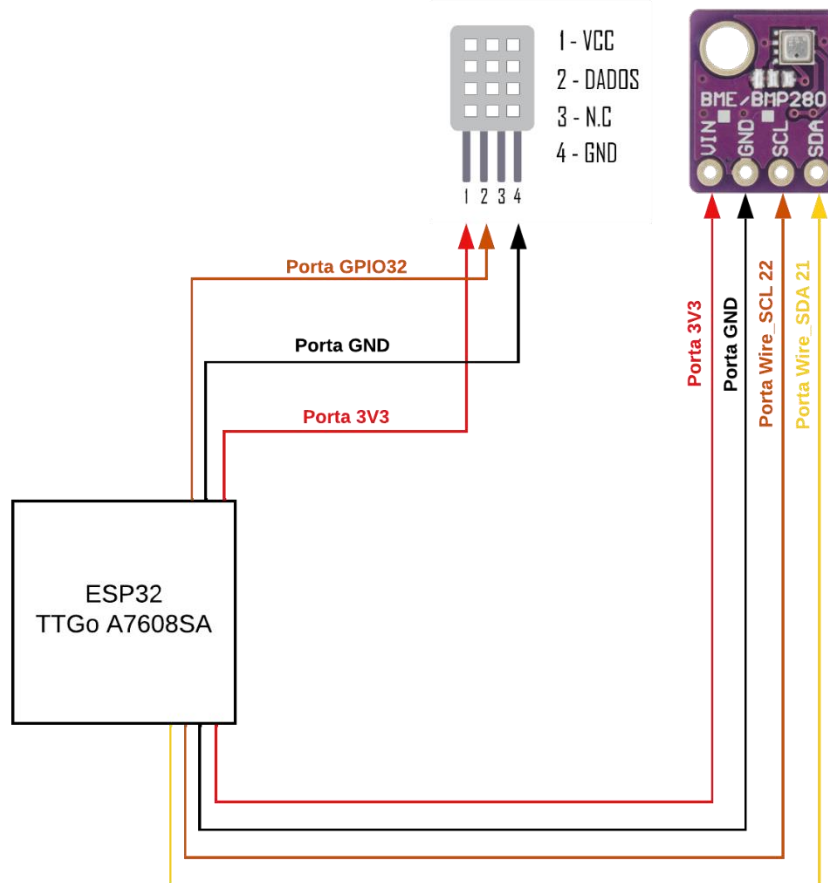
Para o Dashboard, foi criado duas abas que permitem mostrar os dados em tempo real e outra com uma variação temporal, conforme mostrado nas Figuras 26 e 27.

Como pode ser notado na Figura 26, é possível fazer o acompanhamento em tempo real dos valores dos parâmetros durante um voo com um VANT, para uma análise temporal dos resultados tem-se a aba do dashboard da Figura 25 que mostra exatamente a variação ao longo do tempo.

Na Figura 27 é mostrado a tela de *Debug*, no qual mostra como os dados chegam do modem até o *broker*, o pacote é escrito no formato JSON. É importante notar que o *delay* segue sendo de 5 segundos conforme projetado.

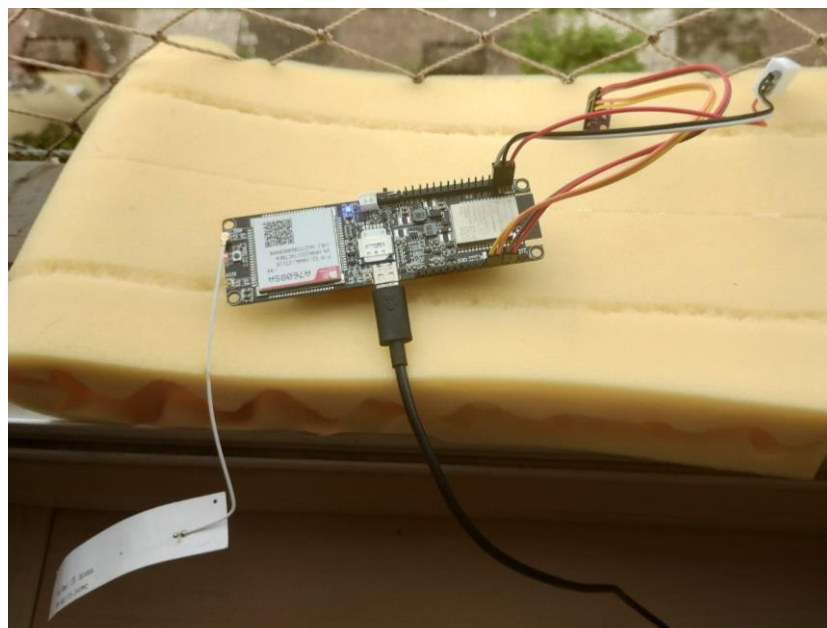
O sistema também foi pensado para que fosse possível o acesso a qualquer outro dispositivo, não apenas ao servidor, como mostrado na Figura 29.

Figura 24 - Diagrama Esquemático Circuito Final.



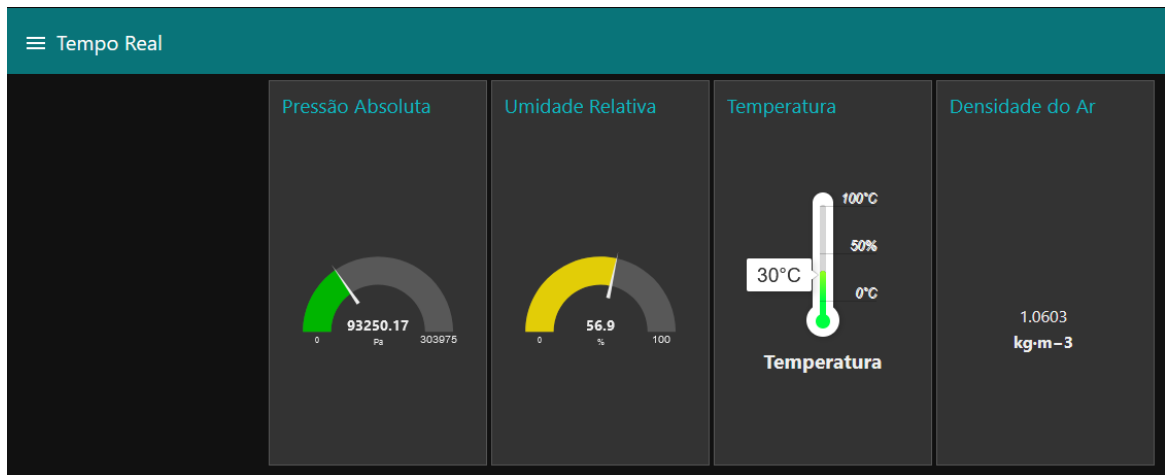
Fonte: Próprio Autor.

Figura 25 - Montagem do Sistema Completo



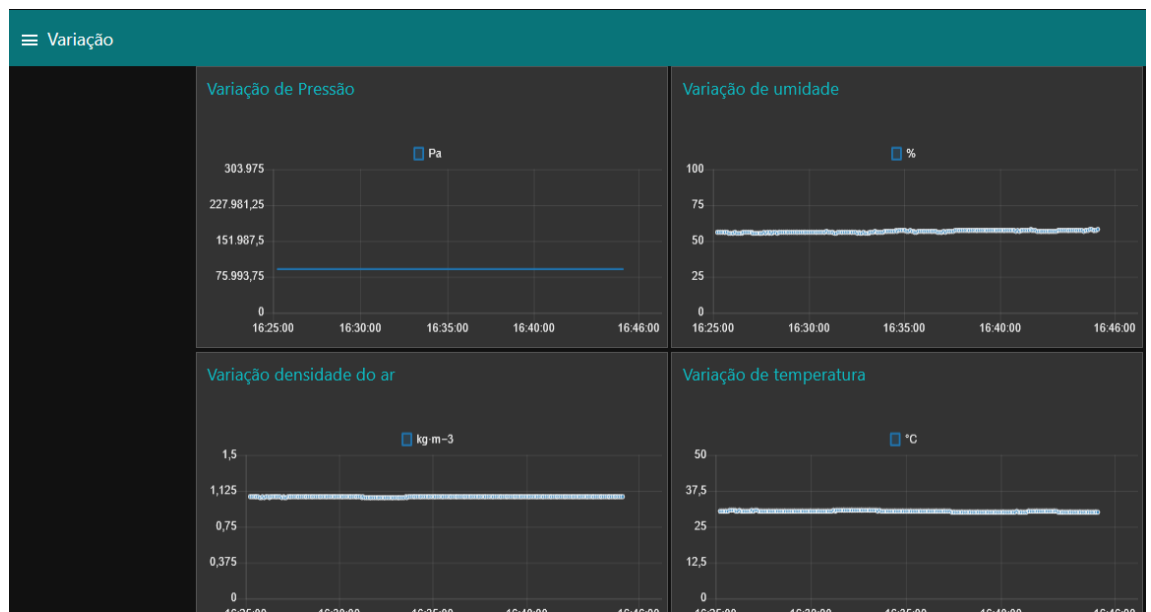
Fonte: Próprio autor.

Figura 26 - Aba Dashboard “Tempo Real”

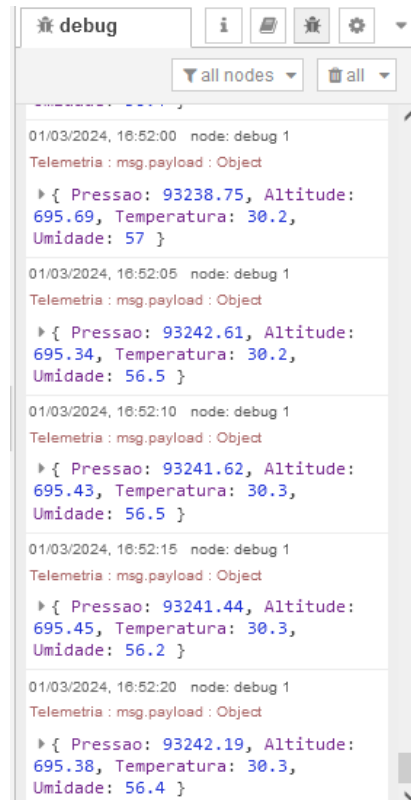


Fonte: Próprio autor.

Figura 27 – Aba Dashboard “Variação”

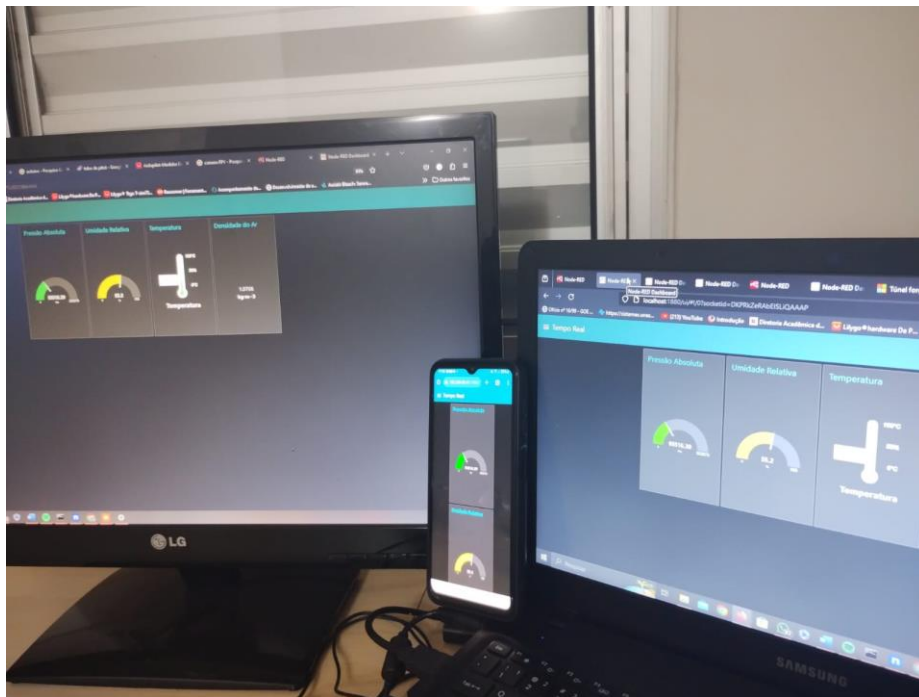


Fonte: Próprio autor.

Figura 28 - Tela de *Debug*

Fonte: Próprio autor.

Figura 29 - Dashboard em Diferentes Dispositivos.



Fonte: Próprio autor.

6. CONCLUSÃO

O trabalho consistiu na construção de um sistema de telemetria para envio de dados atmosféricos baseado em redes móveis, utilizando como estudo de caso o uso em VANT's que têm como um de seus vários empregos o monitoramento a distância. Foi desenvolvido utilizando plataformas de prototipagem eletrônica de hardware livre ESP32, linguagem C e Javascript, compilador de código aberto Arduino IDE e de ferramentas de fluxo de dados Node-RED.

A expansão das torres de telefonia móvel no território nacional, assim como a melhoria na qualidade do serviço, permite cada vez mais a utilização para outros fins além da cobertura de internet para dispositivos móveis, tais como o uso adotado no projeto para coleta de dados a distância. A vantagem desse sistema é a grande cobertura que esse sistema oferece aliado com a velocidade e confiabilidade do envio de dados.

O protótipo desenvolvido alia sensores, também de hardware livre, tais como sensor de temperatura e umidade DHT11 e sensor de pressão barométrica BMP280. Utilizando conceitos de IoT para fazer envio de dados para um servidor que atua como Broker e por fim repassando para a ferramenta Node-RED que fará a interpretação e apresentação dos dados em dashboard.

Em relação a qualidade da amostragem de dados pelos sensores, se mostrou extremamente confiável, com erros abaixo de 1%. O maior erro experimental foi de 8% com a medida de umidade, que foi devidamente justificado com as considerações de que o local físico onde foi feito as medidas pode apresentar características levemente diferentes do local onde o protótipo está localizado, por mais que se trate da mesma cidade.

A qualidade da transmissão de dados também se mostrou bastante satisfatória com a medição da intensidade, considerado no meio da telecomunicação como um sinal de altíssima qualidade. Por fim, os dados foram enviados utilizando protocolo MQTT que proporciona uma entrega de pacotes otimizados e sem perdas, como é observado nos testes onde foi observado uma entrega de dados com delay de 5 segundos, como esperado do projeto.

O Dashboard proporcionou uma visão gráfica dos dados do projeto, mostrando em tempo real assim como a variação ao longo do tempo. Outra vantagem demonstrada é o acesso ao dashboard, que não precisa ser visualizado

necessariamente do *broker*, e sim de qualquer dispositivo com o endereço IP público e porta do servidor.

Para futuros trabalhos, propõe-se o incremento de outros sensores que possam ser interessantes para aplicações em VANT's, tais como GPS para fornecer informações de geoposicionamento, sensor de pressão diferencial e tubo de pitot, para fornecer parâmetros de velocidade relativa no ar e até mesmo recursos para fazer o monitoramento por imagens tais como câmeras FPV.

Outra sugestão de melhoria é o incremento de medidas de segurança para proteger os servidores contra invasões. Como o objetivo do trabalho foi mais focado na criação do sistema, medidas importantes como autenticação não foram incluídas no projeto, deixando para uma próxima oportunidade.

REFERÊNCIAS

AOSONG. Temperature and humidity module: dht22 product manual. Guangzhou: Aosong, 2024. Disponível em: <https://www.makerhero.com/img/files/download/DHT22-AM2302-Datasheet.pdf>. Acesso em: 27 jan. 2024.

ARDUINO. About Arduino. 2021. Disponível em: <https://www.arduino.cc/en/about>. Acesso em: 27 jan. 2024.

BEZERRA, Juliana. Revolução Industrial: o que foi (resumo). Toda Matéria, [s.d.]. Disponível em: <https://www.todamateria.com.br/revolucao-industrial/>. Acesso em: 27 Janeiro de 2024

BOSCH. BMP280: Digital Pressure Sensor: datasheet. Germany: Bosch, 2024. Disponível em: <https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf>. Acesso em: 27 jan. 2024.

CANAL RURAL. **Quem é o 'Rei dos hectares' no Brasil? Conheça nossos 3 maiores produtores agrícolas.** 2021. Disponível em: <https://www.canalrural.com.br/agricultura/quem-e-o-rei-dos-hectares-no-brasil-conheca-nossos-3-maiores-produtores-agricolas/>. Acesso em: 30 abr. 2024.

CEPAGRI (Campinas). Unicamp. **Tempo e Clima Unicamp.** 2024. Disponível em: <https://www.cpa.unicamp.br/>. Acesso em: 28 jan. 2024.

COMO funcionam as ondas de rádio? 2024. Disponível em: <https://www.electricity-magnetism.org/pt-br/como-funcionam-as-ondas-de-radio/>. Acesso em: 27 jan. 2024.

DUSSAUX, Gautier; KASHIWAGI, Marcela; CAPITANGO, Miguel; HEINZELMANN, Ricardo. **Tecnologias de Rede em Telefonia Móvel.** 2010. Disponível em: https://www.gta.ufrj.br/grad/10_1/movel/index.html. Acesso em: 30 abr. 2024.

UNIVERSIDADE FEDERAL DO RIO DE JANEIRO (Rio de Janeiro). OFDM/OFDMA (Orthogonal Frequency Division Multiplexing / Multiple Access). Disponível em:

[https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/coutinho/OFDM.OFDMA\(OrthogonalFrequencyDivisionMu.html](https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/coutinho/OFDM.OFDMA(OrthogonalFrequencyDivisionMu.html). Acesso em: 27 jan. 2024.

COUTINHO, Otto Pedro Smith. **Rádios Cognitivos**. 2008. 7 f. TCC (Graduação) - Curso de Engenharia de Computação e Informação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2008. Disponível em: https://www.gta.ufrj.br/ensino/eel879/trabalhos_vf_2008_2/coutinho/index.html. Acesso em: 30 abr. 2024.

EMBRAPA. A Cidade de Campinas. 2024. Disponível em: <https://www.embrapa.br/fauna-campinas/a-cidade-de-campinas>. Acesso em: 28 jan. 2024.

EXPRESSIF. ESP32-WROVER-E & ESP32-WROVER-IE: datasheet. Shanghai: Expressif, 2023. 31 p. Disponível em: https://www.espressif.com/sites/default/files/documentation/esp32-wrover-e_esp32-wrover-ie_datasheet_en.pdf. Acesso em: 27 jan. 2024.

Ferreira, C.O., Nepomuceno, L.M., Negreiros, S.L.V., Neves, A.M., Silva, R.G.A., Góes, L.C. & Fischer, C. Telemetria e sistema de aquisição de dados em voo para uma Aeronave Remotamente Pilotada (RPA) telemaster. *The Academic Society Journal*, 3(1) 42-49, 2019. DOI: doi.org/10.32640/tasj.2019.1.42.

HWANG, André; GONZALEZ, Antonio Galiza Cerdeira; MONTEIRO, José Guilherme Tavares. **Identificação por Rádio Frequência**. 2013. 8 v. Tese (Doutorado) - Curso de Engenharia de Controle e Automação, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2013. Cap. 2. Disponível em: https://www.gta.ufrj.br/grad/13_1/rfid/index.html. Acesso em: 30 abr. 2024.

ICEL (Manaus). Manual Datalogger HT-4010. 2024. Disponível em: http://www.icel-manau.com.br/produto_descricao.php?id=720. Acesso em: 10 abr. 2024.

IFOOD (São Paulo). Entrega por drones: saiba tudo sobre esse serviço em ascensão. 2022. Disponível em: <https://institucional.ifood.com.br/inovacao/entrega-por-drones-saiba-tudo-sobre-esse-servico-em-ascensao/>. Acesso em: 27 jan. 2024.

KITANO, Claudio. Elementos e Limitações dos Sistemas de Comunicações. Ilha Solteira: Faculdade de Engenharia de Ilha Solteira, 2023. 17 slides, color. Disponível em:

<https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/optoeletronica/capitulo-1---secao-unica.pdf>. Acesso em: 27 jan. 2024.

KITANO, Claudio. Modulação de Amplitude em Banda Lateral Dupla. Ilha Solteira: Faculdade de Engenharia de Ilha Solteira, 2023. 13 slides, color. Disponível em: <https://www.feis.unesp.br/Home/departamentos/engenhariaeletrica/optoeletronica/capitulo-4---secao-4.2-loe.pdf>. Acesso em: 27 jan. 2024.

ORACLE. O que é IoT? Disponível em: <https://www.oracle.com/br/internet-of-things/what-is-iot/>. Acesso em: 27 jan. 2024.

PEDRO CÉSAR TEBALDI. Opservices. **Telemetria: o que é e como funciona?** 2019. Disponível em: <https://www.opservices.com.br/telemetria/>. Acesso em: 30 abr. 2024.

RED, Node. About: Node Red. Disponível em: <https://nodered.org/about/>. Acesso em: 27 jan. 2024.

SERPA BORGES, B. CONSTRUÇÃO DE UM SISTEMA DE AQUISIÇÃO DE SINAIS DE TELEMETRIA DE MÚLTIPLOS TRANSMISSORES. Revista da Graduação, [S. l.], v. 7, n. 2, 2014. Disponível em: <https://revistaseletronicas.pucrs.br/ojs/index.php/graduacao/article/view/19342>. Acesso em: 27 jan. 2024.

SICHONANY, Oni Reasilvia de Almeida Oliveira; Schlosser, José Fernando; Medina, Roseclea Duarte; Roggia, Iria Brucker; Lôbo, Jaziel Souza; Santos, Fernando Beux. Telemetria na transmissão de dados de desempenho de máquinas agrícolas

utilizando tecnologias GSM/GPRS e ZigBee. *Ciência Rural*, Santa Maria, v. 42, n. 8, p. 1430-1433, ago. 2012.

SILVA, Ildelano Ferreira. *Padrões de Telefonia Móvel Digital*. Pontifícia Universidade Católica de Minas Gerais. Minas Gerais. 2010.

SIMCOM (China). *A76XX Series_AT Command Manual*. Shanghai: Simcom Wireless Solutions Limited, 2023. 512 p.

SOUSA, Rafaela. "Primeira Revolução Industrial"; *Brasil Escola*. Disponível em: <https://brasilecola.uol.com.br/geografia/primeira-revolucao-industrial.htm>. Acesso em 27 de janeiro de 2024.

SOUZA, V. L.; RODRIGUES, M. S. ; BATISTA, E. V. L. ; SILVA, F. H. de et al. *TELEMETRIA DE BAIXO CUSTO PARA VEÍCULOS AÉREOS NÃO TRIPULADOS*. In: V COEN UFSJ, 5., 2015, São João del Rei. V COEN UFSJ. São João del Rei: Ufsj, 2015. p. 1-13.

SPAGNA, Julia. *Quarta Revolução Industrial: o que é e quais os impactos na atualidade; Guia do Estudante*; Disponível em: https://guiadoestudante.abril.com.br/atualidades/quarta-revolucao-industrial-o-que-e-quais-os-impactos-na-atualidade/#google_vignette. Acesso em: 27 Janeiro de 2024.

TATEOKI, Getúlio Teruo. *Monitoramento de Dados via Internet baseado em Telefonia Celular*. 2007. 124 f. Dissertação (Mestrado) - Curso de Engenharia Elétrica, Departamento de Engenharia Elétrica, Faculdade de Engenharia de Ilha Solteira, Ilha Solteira, 2007.

TELECO (São Paulo). *Dados na Rede Celular: Evolução das Tecnologias*. Disponível em: https://www.teleco.com.br/tutoriais/tutorialtrafdados/pagina_2.asp. Acesso em: 27 jan. 2024.

THIAGO PERUCH (Minas Gerais). Universidade Federal de Minas Gerais. História do Rádio. Disponível em: <https://www.ufmg.br/espacodoconhecimento/historia-do-radio/>. Acesso em: 27 jan. 2024.

VIKACONTROLS. Módulos XBee – Serie ZigBee. 2024. Disponível em: <https://www.vikacontrols.com.br/produto/modulos-xbee-serie-zigbee/>. Acesso em: 27 jan. 2024.

WORKNETS. Time Division Multiple Access (TDMA). Disponível em: <https://www.worknets.com.br/2016/05/29/time-division-multiple-access-tdma/>. Acesso em: 27 jan. 2024.

XINYUAN-LILYGO. **T-A7608X**. Disponível em: <https://github.com/Xinyuan-LilyGO/T-A7608X>. Acesso em: 30 abr. 2024.

APÊNDICE A – Programação do Software para Coleta de Dados

```

// * Bibliotecas para Comunicação Serial de sensores e Modem * //
#include <Arduino.h>
#include <Wire.h>
#include "DHT.h"
#include <Adafruit_BMP280.h>

// * Definição de parâmetros do modem para comunicação Serial * //
#define TINY_GSM_MODEM_SIM7600
#define SerialMon Serial
#define SerialAT Serial1
#define GSM_PIN ""
#define uS_TO_S_FACTOR 1000000ULL // Fator de conversão de micro segundos para segundos
#define TIME_TO_SLEEP 60
#define UART_BAUD 115200
#define MODEM_DTR 25
#define MODEM_TX 26
#define MODEM_RX 27
#define MODEM_PWRKEY 4
#define BAT_ADC 35
#define BAT_EN 12
#define MODEM_RI 33
#define RESET 5
#define SD_MISO 2
#define SD_MOSI 15
#define SD_SCLK 14
#define SD_CS 13
const char apn[] = "zap.vivo.com.br"; // APN do CHIP VIVO
const char gprsUser[] = ""; // Usuário do CHIP VIVO
const char gprsPass[] = ""; // Senha do CHIP VIVO

// * Definição de Pinos de sensores e criação de objetos * //
#define DHTPIN 0
#define DHTTYPE DHT11

```

```

DHT dht(DHTPIN, DHTTYPE); // Objeto Sensor Umidade
Adafruit_BMP280 bmp; // Objetivo Sensor de Pressão

// * Definição das variáveis globais * //
String broker = "0.tcp.sa.ngrok.io"; // Endereço do broker
String MQTTport="11240"; // Porta do Broker
String payload_telemetria; // Variável para envio de dados

//String gsm_send_serial(String command, int delay);

// * Função de comandos AT interface ESP32-MODEM * //
String gsm_send_serial(String command, int delay)
{
  String buff_resp = "";
  Serial.println("Send ->: " + command);
  SerialAT.println(command);
  long wtimer = millis();
  while (wtimer + delay > millis())
  {
    while (SerialAT.available())
    {
      buff_resp = SerialAT.readString();
      Serial.println(buff_resp);
    }
  }
  Serial.println();
  return buff_resp;
}

// * Função para executar comandos de configuração inicial do modem * //
void Init(void)
{
  delay(1000);
  gsm_send_serial("AT+CFUN=1", 1000);
  gsm_send_serial("AT+CPIN?", 1000);
  gsm_send_serial("AT+CSQ", 1000);
}

```

```

gsm_send_serial("AT+CREG?", 1000);
gsm_send_serial("AT+COPS?", 1000);
gsm_send_serial("AT+CGATT?", 1000);
gsm_send_serial("AT+CPSI?", 500);
gsm_send_serial("AT+CGDCONT=1,\"IP\", \"zap.vivo.com.br\"", 1000);
gsm_send_serial("AT+CGACT=1,1", 1000);
gsm_send_serial("AT+CGATT?", 1000);
gsm_send_serial("AT+CGQREQ=1,1,1,5,9,31", 1000);
gsm_send_serial("AT+CGQMIN=1,1,1,5,9,31", 1000);
gsm_send_serial("AT+NETOPEN", 500);
}

// * Função para estabelecer comunicação com o Broker */
void getMQTTs()
{
    gsm_send_serial("AT+CMQTTSTART", 1000); // Começa o serviço MQTT, é necessário executar essa função SEMPRE antes de qualquer comando
    gsm_send_serial("AT+CMQTTACCQ=0,\"client test0\",0", 1000);
    gsm_send_serial("AT+CMQTTACCQ?", 1000); // Usa para fazer a aquisição do MQTT Client, deve ser executada após CMQTTSTART (sempre)

    gsm_send_serial("AT+CMQTTCONNECT=0,\"tcp://"+broker+": "+MQTTport+"\",20,1",
    1000);

}

void setup()
{
    Serial.begin(115200);
    dht.begin();
    if (!bmp.begin()) {
        Serial.println(F("Could not find a valid BMP280 sensor, check wiring or "
            "try a different address!"));
        while (1) delay(10);
    }
    SerialAT.begin(UART_BAUD, SERIAL_8N1, MODEM_RX, MODEM_TX);
    pinMode(MODEM_PWRKEY, OUTPUT);
}

```

```

digitalWrite(MODEM_PWRKEY, HIGH);
delay(1000);
digitalWrite(MODEM_PWRKEY, LOW);
delay(1000);
digitalWrite(MODEM_PWRKEY, HIGH);
delay(2000);
Init();
getMQTTTS();
}

void loop()
{
  // * Variáveis para armazenamento de dados * //
  float pressao = bmp.readPressure();
  float alt = bmp.readAltitude(1013.2);
  float umidade = dht.readHumidity();
  float temperatura = dht.readTemperature();

  // * Montagem do pacote de dados a ser enviado * //
  String payload_telemetria = "{";
  payload_telemetria+="\"Pressao\":";
  payload_telemetria+= String(pressao, 2);
  payload_telemetria+=",";
  payload_telemetria+="\"Altitude\":";
  payload_telemetria+= String(alt, 2);
  payload_telemetria+=",";
  payload_telemetria+="\"Temperatura\":";
  payload_telemetria+= String(temperatura, 2);
  payload_telemetria+=",";
  payload_telemetria+="\"Umidade\":";
  payload_telemetria+= String(umidade, 2);
  payload_telemetria+="}";

  // * Execução da função de comunicação serial com modem * //
  gsm_send_serial("AT+CMQTTTOPIC=0,10", 200);
  gsm_send_serial("Telemetria\x1A", 200);
}

```

```
    gsm_send_serial("AT+CMQTTPAYLOAD=0,"+(String)(payload_telemetria.length()),  
200);  
    gsm_send_serial(payload_telemetria+"\x1A", 200);  
    gsm_send_serial("AT+CMQTT PUB=0,0,60", 200);  
  
}
```

ANEXO A – Programação testes comandos AT

```

/*
  FILE: ATdebug.ino
  PURPOSE: Test functionality
*/

#define TINY_GSM_MODEM_SIM7600 //A7608's AT instruction is
compatible with SIM7600
#define TINY_GSM_RX_BUFFER 1024 // Set RX buffer to 1Kb
#define SerialAT Serial1
// See all AT commands, if wanted
#define DUMP_AT_COMMANDS
#include <TinyGsmClient.h>
#include <SPI.h>
#include <SD.h>
#include <Ticker.h>
#ifdef DUMP_AT_COMMANDS // if enabled it requires the streamDebugger
lib
#include <StreamDebugger.h>
StreamDebugger debugger(SerialAT, Serial);
TinyGsm modem(debugger);
#else
TinyGsm modem(SerialAT);
#endif
#define uS_TO_S_FACTOR 1000000ULL // Conversion factor for micro
seconds to seconds
#define TIME_TO_SLEEP 60 // Time ESP32 will go to sleep (in
seconds)
#define UART_BAUD 115200
#define PIN_DTR 25
#define PIN_TX 26
#define PIN_RX 27

```

```
#define PWR_PIN 4
#define BAT_ADC 35
#define BAT_EN 12
#define PIN_RI 33
#define RESET 5
#define SD_MISO 2
#define SD_MOSI 15
#define SD_SCLK 14
#define SD_CS 13
int counter, lastIndex, numberOfPieces = 24;
String pieces[24], input;
bool reply = false;
void modem_on()
{
    // Set-up modem power pin
    Serial.println("\nStarting Up Modem...");
// BAT EN
    pinMode(BAT_EN, OUTPUT);
    digitalWrite(BAT_EN, HIGH);
    pinMode(PIN_DTR, OUTPUT);
    digitalWrite(PIN_DTR, LOW);
//A7608 Reset
    pinMode(RESET, OUTPUT);
    digitalWrite(PWR_PIN, LOW);
    delay(100);
    digitalWrite(RESET, HIGH);
    delay(1000);
    digitalWrite(RESET, LOW);
//A7608 Power on
    pinMode(PWR_PIN, OUTPUT);
    digitalWrite(PWR_PIN, LOW);
    delay(100);
    digitalWrite(PWR_PIN, HIGH);
    delay(1000);
```

```

digitalWrite(PWR_PIN, LOW);
int i = 10;
Serial.println("\nTesting Modem Response...\n");
Serial.println("*****");
while (i) {
    SerialAT.println("AT");
    delay(500);
    if (SerialAT.available()) {
        String r = SerialAT.readString();
        Serial.println(r);
        if ( r.indexOf("OK") >= 0 ) {
            reply = true;
            break;;
        }
    }
    delay(500);
    i--;
}
Serial.println("*****\n");
}

void setup()
{
    Serial.begin(UART_BAUD); // Set console baud rate
    SerialAT.begin(115200, SERIAL_8N1, PIN_RX, PIN_TX);
    delay(100);

    modem_on();
    if (reply) {
        Serial.println(F("*****"));
        Serial.println(F(" You can now send AT commands"));
        Serial.println(F(" Enter \"AT\" (without quotes), and you should see
\"OK\""));
    }
}

```

```

        Serial.println(F(" If it doesn't work, select \"Both NL & CR\" in Serial
Monitor"));

```

```

        Serial.println(F(" DISCLAIMER: Entering AT commands without knowing
what they do"));

```

```

        Serial.println(F(" can have undesired consequences..."));

```

```

Serial.println(F("*****\n"));

```

```

    } else {

```

```

        Serial.println(F("*****"));

```

```

        Serial.println(F(" Failed to connect to the modem! Check the baud and
try again."));

```

```

Serial.println(F("*****\n"));

```

```

    }

```

```

}

```

```

void loop()

```

```

{

```

```

    while (true) {

```

```

        if (SerialAT.available()) {

```

```

            Serial.write(SerialAT.read());

```

```

        }

```

```

        if (Serial.available()) {

```

```

            SerialAT.write(Serial.read());

```

```

        }

```

```

        delay(1);

```

```

    }

```

```

}

```

ANEXO B – Cálculo da densidade do ar

A densidade do ar é uma variável da dinâmica da atmosfera, que sofre alterações quando o ar é aquecido e resfriado, quando há alteração na umidade ou quando há um aumento de pressão que depende diretamente da altitude. Vale mencionar que essa densidade também é alterada quando há misturas de outros gases em maior concentração além dos habituais. Em geral é considerado que a nível do mar e a 15°C o ar tem uma densidade de 1,225 kg/m³.

A fórmula mais precisa para se calcular a densidade, levando em consideração a umidade é:

$$\rho_{ar\acute{u}mido} = \frac{p_d}{R_d T} + \frac{p_v}{R_v T} = \frac{p_d M_d + p_v M_v}{R T} \quad (3)$$

Onde:

$$\rho_{ar\acute{u}mido} = \text{Densidade do ar úmido} \left(\frac{kg}{m^3} \right)$$

$$p_d = \text{Pressão parcial do ar seco} (Pa)$$

$$R_d = \text{Constante específica do gás para o ar seco} 287,058 \left(\frac{J}{kg \cdot K} \right)$$

$$T = \text{Temperatura} (K)$$

$$p_v = \text{Pressão do vapor d'água} (Pa)$$

$$R_v = \text{Constante específica do gás para o vapor d'água} 461,495 \left(\frac{J}{kg \cdot K} \right)$$

$$M_d = \text{Massa molar do ar seco} 0,028964 \left(\frac{kg}{mol} \right)$$

$$M_v = \text{Massa molar do vapor d'água} 0,018016 \left(\frac{kg}{mol} \right)$$

$$R = \text{Constante do gás ideal} 8,314 \left(\frac{J}{K \cdot mol} \right)$$

A pressão de vapor da água pode ser calculada pela pressão de saturação do vapor e a umidade relativa, sendo obtida por:

$$p_v = UR\% \cdot p_{sat} \quad (4)$$

Onde:

$$UR\% = \text{Umidade relativa}$$

Para o cálculo da p_{sat} é utilizado a equação (5):

$$p_{sat} = 6,1078 \cdot 10^{\frac{7,5T}{T+237,3}} \quad (5)$$

Onde a variável T se encontra em graus Célcius. Para pressão parcial do ar seco é considerado a pressão absoluta, conforme equação (6):

$$p_d = p - p_v \quad (6)$$

Onde:

$$p = \text{Pressão absoluta (Pa)}$$