



**UNIVERSIDADE DO ESTADO DO AMAZONAS – UEA**  
**ESCOLA SUPERIOR DE TECNOLOGIA – EST**

**JESSAMINE MARIA DE LIMA AZEVEDO**

**UMA ESTRATÉGIA DE *WAKE-UP RADIO* PARA COMUNICAÇÃO  
ORIENTADA A EVENTOS EM UMA REDE LORAWAN CLASSE A**

**Manaus**

**2022**

**JESSAMINE MARIA DE LIMA AZEVEDO**

**UMA ESTRATÉGIA DE *WAKE-UP RADIO* PARA COMUNICAÇÃO  
ORIENTADA A EVENTOS EM UMA REDE LORAWAN CLASSE A**

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II, apresentado à banca avaliadora do curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador Prof. Dsc. Jozias Parente de Oliveira

Manaus

2022

**Universidade do Estado do Amazonas – UEA**  
**Escola Superior de Tecnologia - EST**

*Reitor:*

**André Luiz Nunes Zogahib**

*Vice-Reitor:*

**Kátia do Nascimento Couceiro**

*Diretor da Escola Superior de Tecnologia:*

**Ingrid Sammyne Gadelha Figueiredo**

*Coordenador do Curso de Engenharia Elétrica:*

**Israel Gondres Torné**

*Banca Avaliadora composta por:*

**Prof. Jozias Parente de Oliveira (Orientador)**

**Prof. Israel Gondres Torné**

**Prof. Rubens de Andrade Fernandes**

*Data da defesa: 14/10/2022.*

## **CIP – Catalogação na Publicação**

Azevedo, Jessamine Maria de Lima

Uma estratégia de *Wake-Up Radio* para comunicação orientada a eventos em uma rede LoRaWAN Classe A. / Jessamine Maria de Lima Azevedo; [orientado por] Jozias Parente de Oliveira. – Manaus: 2022.

65 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica).  
Universidade do Estado do Amazonas, 2022.

1. *IoT* 2. LoRaWAN 3. Sistemas Embarcados 4. *Wake-Up Radio*  
I. Parente de Oliveira, Jozias.

JESSAMINE MARIA DE LIMA AZEVEDO

UMA ESTRATÉGIA DE *WAKE-UP RADIO* PARA COMUNICAÇÃO  
ORIENTADA A EVENTOS EM UMA REDE LORAWAN CLASSE A

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II, apresentado à banca avaliadora do curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Bacharel em Engenharia Elétrica.

Nota obtida: 9,6 ( Nove vírgula seis )

Aprovado em 14/10/2022

Área de Concentração: Engenharia Elétrica

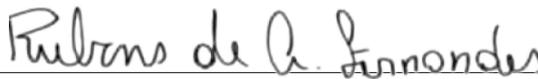
BANCA EXAMINADORA



Orientador: Jozias Parente de Oliveira, Dr.



Avaliador: Israel Gondres Torné, Dr.



Avaliador: Rubens de Andrade Fernandes, Ms.

Manaus

2022

*Dedico este trabalho a um querido amigo que hoje não se encontra entre nós. Queria ter notados seus sinais, pois agora o que me resta é eternamente sentir sua falta.*

## **AGRADECIMENTOS**

Agradeço a minha mãe por todo amor que me deu, pelo apoio e incentivo para que eu seguisse meus sonhos.

Agradeço ao meu irmão Vítor Norton de Lima Azevedo por sempre estar ao meu lado, por me ouvir, incentivar e me apoiar.

Agradeço ao Professor Jozias Parente de Oliveira, meu orientador, não só por toda atenção e tempo despendido, mas também pelo apoio e profissionalismo que tanto admiro.

Agradeço ao Rubens de Andrade Fernandes e ao Ruan Teixeira Mota, pelas recomendações e contribuições para o aperfeiçoamento deste estudo.

Agradeço aos meus queridos amigos, Beatriz Correia, Gabriel Caporazzo, Gabriel Rocha, Ingrid Mayumi, Luis Eduardo Mena, Nathália Cristina, Paula Fernanda e Wesley Fábio, pelo companheirismo, pelas risadas e angústias compartilhadas.

Agradeço a todos os amigos, professores e colegas do Laboratório de Sistemas Embarcados, pela oportunidade que me deram, por toda a experiência e aprendizado adquiridos.

Agradeço a todo o corpo docente do curso de Engenharia Elétrica da Universidade Estado do Amazonas, pelos conhecimentos repassados.

*"A persistência é o caminho do êxito."  
- Charles Chaplin*

## RESUMO

O presente trabalho tem como objetivo o desenvolvimento uma estratégia baseado na tecnologia de *Wake-Up Radio* para comunicação orientada a eventos em uma rede LoRaWAN Classe A. Para isso, foram desenvolvidos circuitos de radio despertar, bem como algoritmos de *software* embarcado para plataformas microcontroladas, com o objetivo da redução do consumo energético de um *end-node* de uma rede LoRaWAN classe A. Inicialmente, foi realizada uma pesquisa sobre plataformas microcontroladas que poderiam ser utilizadas para o desenvolvimento de uma rede LoRaWAN classe A. Em seguida, foram realizados levantamentos de materiais tecnocientíficos relacionados a *Wake-Up Radio* em redes LoRaWAN. Com isso, foi necessária uma pesquisa sobre rádios de baixo consumo, para serem utilizados com rádio secundário dos dispositivos pertencentes a rede. Por fim, foram estabelecidas as comunicações entre os dispositivos, para que, assim, fossem realizadas as validações do projeto e medições de consumo do dispositivo.

**Palavras-chave:** IoT. LoRaWAN. Sistemas Embarcados. *Wake-Up Radio*.

## **ABSTRACT**

This paper aims to develop a strategy based on Wake-Up Radio technology for event-driven communication in a Class A LoRaWAN network. For this, radio wake-up circuits were developed, as well as embedded software algorithms for microcontrolled platforms, with the objective of reducing the energy consumption of an end-node of a Class A LoRaWAN network. Initially, a survey was conducted on microcontrolled platforms that could be used for the development of a LoRaWAN class A network. Next, surveys of techno-scientific materials related to Wake-Up Radio in LoRaWAN networks were performed. This required research on low-power radios, to be used as secondary radios for the devices belonging to the network. Finally, communications between the devices were established so that the project could be validated and the device consumption measured.

**Keywords:** Embedded Systems. IoT. LoRaWAN. Wake-Up Radio.

## LISTA DE ILUSTRAÇÕES

|    |  |    |
|----|--|----|
| 1  | Elementos Típicos de um Microcontrolador . . . . .                               | 21 |
| 2  | Apresentação Paradigma de IoT . . . . .  | 21 |
| 3  | Topologia de Rede de Enlace Ponto a Ponto . . . . .                              | 22 |
| 4  | Topologias de Redes de Enlace por Difusão . . . . .                              | 23 |
| 5  | Escalas de Redes de Transmissão de Dados . . . . .                               | 23 |
| 6  | Fluxo de Dados (Simplex, Half-Duplex e Full-Duplex) . . . . .                    | 24 |
| 7  | Camadas da Arquitetura OSI e TCP/IP . . . . .                                    | 25 |
| 8  | Relação Consumo x Alcance de tecnologias de comunicação sem fio . . . . .        | 26 |
| 9  | Frequências dos Canais do Plano AU915 . . . . .                                  | 27 |
| 10 | Up e Down Chirp no Domínio do Tempo . . . . .                                    | 27 |
| 11 | Comparação do Fator de Espalhamento: SF7 à SF12 . . . . .                        | 29 |
| 12 | Camadas do Protocolo LoRaWAN . . . . .   | 30 |
| 13 | Arquitetura de uma rede LoRa . . . . .   | 31 |
| 14 | Diagrama de Distribuição das Chaves de Ativação para OTAA . . . . .              | 33 |
| 15 | Diagrama de funcionamento das Ativações ABP e OTAA . . . . .                     | 34 |
| 16 | Composição de uma Mensagem de Uplink . . . . .                                   | 34 |
| 17 | Composição de uma Mensagem de Downlink . . . . .                                 | 34 |
| 18 | Estrutura de uma Mensagem LoRaWAN . . . . .                                      | 35 |
| 19 | Janela de transmissão de uma rede LoRaWAN Classe A . . . . .                     | 36 |
| 20 | Janela de transmissão de uma rede LoRaWAN Classe B . . . . .                     | 36 |
| 21 | Janela de transmissão de uma rede LoRaWAN Classe C . . . . .                     | 36 |
| 22 | Exemplo de arquitetura de um <i>end-node</i> para <i>wake-up radio</i> . . . . . | 38 |
| 23 | Exemplo de arquitetura de um <i>gateway</i> para <i>wake-up radio</i> . . . . .  | 39 |
| 24 | Sinal em Modulação ASK . . . . .   | 39 |
| 25 | Sinal em Modulação ASK . . . . .   | 40 |
| 26 | Diagrama de Desenvolvimento do Projeto . . . . .                                 | 41 |
| 27 | Arquitetura Geral do Sistema . . . . .   | 42 |
| 28 | Módulo Heltec Wi-Fi Lora V2 . . . . .  | 44 |
| 29 | Módulo de Rádio SYN115 e SYN 480R . . . . .                                      | 46 |
| 30 | Definição na Biblioteca LMIC da Região e Rádio Utilizados . . . . .              | 48 |
| 31 | Função para Habilitar Comunicação via OOK . . . . .                              | 48 |
| 32 | Definição dos Parâmetros do Gateway . . . . .                                    | 49 |
| 33 | Definição da Rede Wi-Fi Utilizada . . . . .                                      | 50 |
| 34 | Função para Processamento de Mensagens MQTT . . . . .                            | 50 |
| 35 | Dashboard do Broker MQTT Utilizado . . . . .                                     | 51 |
| 36 | Tela de Inicial do Console . . . . .   | 52 |

|    |  |    |
|----|--|----|
| 37 | Tela de Inicial do Console . . . . .                                   | 52 |
| 38 | Tela para Adição de uma Nova Aplicação . . . . .                       | 52 |
| 39 | Tela de Visão Geral da Aplicação . . . . .                             | 53 |
| 40 | Tela de Inicial do Cadastro de um End-Node . . . . .                   | 53 |
| 41 | Configurações Avançadas do Cadastro de um Novo End-Node . . . . .      | 54 |
| 42 | Tela de Visão Geral do End-Node . . . . .                              | 54 |
| 43 | Registros das Chaves de Ativação no End-Node . . . . .                 | 55 |
| 44 | Tela para Adição de um Novo Gateway . . . . .                          | 55 |
| 45 | Tela de Inicial do Cadastro de um Gateway . . . . .                    | 55 |
| 46 | Registros Exibidos da Porta Serial do Gateway ao Inicializar . . . . . | 56 |
| 47 | Tela de Visão Geral do Gateway . . . . .                               | 56 |
| 48 | Osciloscópio e Ponta de Prova Utilizada para Aferições . . . . .       | 57 |
| 49 | Troca de mensagens via MQTT . . . . .                                  | 58 |
| 50 | Gateway inicializado e recebendo pacotes via MQTT e LoRaWAN . . . . .  | 58 |
| 51 | End-node realizando o envio mensagem via LoRaWAN . . . . .             | 59 |
| 52 | Log da Aplicação no console TTN . . . . .                              | 59 |
| 53 | Log do Gateway no console TTN . . . . .                                | 59 |
| 54 | Medição em Standby . . . . .   | 60 |
| 55 | Medição do primeiro pico da transmissão LoRa . . . . .                 | 60 |

## LISTA DE TABELAS

|   |   |    |
|---|---|----|
| 1 | Características de cada classes de Dispositivos . . . . .   | 37 |
| 2 | Especificações técnicas do Heltec Wi-Fi LoRa V2 . . . . .   | 45 |
| 3 | Especificações técnicas do SX1276 . . . . .   | 45 |
| 4 | Conexões entre o módulo SYN115 e o módulo Heltec Wi-Fi LoRa V2 .  | 46 |
| 5 | Conexões entre o módulo SYN480R e o módulo Heltec Wi-Fi LoRa V2   | 46 |
| 6 | Bibliotecas utilizadas para a implementação do <i>firmware</i> do <i>gateway</i> e<br><i>end-node</i> . . . . . | 47 |
| 7 | Comunicação . . . . .   | 59 |

## LISTA DE ABREVIATURAS E SIGLAS

|         |  |
|---------|--|
| ABP     | <i>Activation By Personalization</i>                       |
| AES     | <i>Advanced Encryption Standard</i>                        |
| ANATEL  | Agência Nacional de Telecomunicações                       |
| AppEUI  | <i>Application Unique Identifier</i>                       |
| AppKey  | <i>Application Key</i>                                     |
| AppSKey | <i>Application Session Key</i>                             |
| ASK     | <i>Amplitude Shift Keying</i>                              |
| BLE     | <i>Bluetooth Low Energy</i>                                |
| BW      | <i>Bandwidth</i>   |
| CHIRP   | <i>Compressed High Intensity Radar Puls</i>                |
| CR      | <i>Coding Rate</i>   |
| CSS     | <i>Chirp Spread Spectrum</i>                               |
| DevAddr | <i>Device Address</i>                                      |
| DevEUI  | <i>End-Device Unique Identifier</i>                        |
| EEPROM  | <i>Electrically Erasable Programmable Read-Only Memory</i> |
| GPIO    | <i>General Purpose Input-Output</i>                        |
| IDE     | <i>Integrated Development Environment</i>                  |
| IEEE    | <i>Institute of Electrical and Electronics Engineers</i>   |
| IoT     | <i>Internet of Things</i>                                  |
| IP      | <i>Internet Protocol</i>                                   |
| ISM     | <i>Industrial Sientific and Medical</i>                    |
| ISO     | <i>International Organization for Standardization</i>      |
| JoinEUI | <i>Join Server Unique Identifier</i>                       |

|         |  |
|---------|--|
| LAN     | <i>Local Area Network</i>                          |
| LoRa    | <i>Long Range</i>                                  |
| NwkSKey | <i>Application Session Key</i>                     |
| OOK     | <i>On-Off Keying</i>                               |
| OTA     | <i>Over-The-Air</i>                                |
| OTAA    | <i>Over-The-Air Activation</i>                     |
| OSI     | <i>Open Systems Interconnection</i>                |
| PSK     | Chaveamento de Fase                                |
| QPSK    | Chaveamento de Fase em Quadratura                  |
| RAM     | <i>Random Access Memory</i>                        |
| RF      | Rádio Frequência                                   |
| SF      | <i>Spreading Factor</i>                            |
| TCP     | <i>Transmission Control Protocol</i>               |
| TDOA    | <i>Time Difference of Arrival</i>                  |
| UART    | <i>Universal Asynchronous Receiver/Transmitter</i> |
| USB     | <i>Universal Serial Bus</i>                        |
| WAN     | <i>Wide Area Network</i>                           |
| WuP     | <i>Wake-up Package</i>                             |
| WuR     | <i>Wake-up Radio</i>                               |
| WuRx    | <i>Wake-up Receiver</i>                            |
| WuTx    | <i>Wake-up Transmitter</i>                         |

## SUMÁRIO

|  |           |
|--|-----------|
| <b>INTRODUÇÃO</b>                                  | <b>16</b> |
| <b>1 REFERENCIAL TEÓRICO</b>                       | <b>20</b> |
| 1.1 Sistemas Embarcados                            | 20        |
| 1.2 Microcontroladores                             | 20        |
| 1.3 Internet das Coisas                            | 21        |
| 1.4 Rede de Comunicação de Dados                   | 22        |
| 1.4.1 Tecnologia de Transmissão de Dados           | 22        |
| 1.4.2 Escala de Transmissão de Dados               | 23        |
| 1.4.3 Tipos de Comunicação                         | 24        |
| 1.4.4 Arquitetura de Redes                         | 24        |
| 1.4.5 MQTT   | 25        |
| 1.5 Comunicação de Longo Alcance                   | 25        |
| 1.5.1 LPWAN  | 25        |
| 1.5.2 LoRa   | 26        |
| 1.5.2.1 A Camada Física do LoRa                    | 27        |
| 1.5.2.2 Largura de Banda                           | 28        |
| 1.5.2.3 Taxa de Codificação                        | 28        |
| 1.5.2.4 Fator de Espalhamento                      | 28        |
| 1.5.2.5 Data Rate                                  | 30        |
| 1.5.3 LoRaWAN                                      | 30        |
| 1.5.3.1 Servidor de Rede                           | 31        |
| 1.5.3.2 Servidor de Aplicação                      | 32        |
| 1.5.3.3 Gateway                                    | 32        |
| 1.5.3.4 Dispositivos Finais                        | 32        |
| 1.5.3.5 Tipos de Ativação                          | 32        |
| 1.5.3.6 Formato dos Frames                         | 34        |
| 1.5.3.7 Classe de Dispositivos                     | 35        |
| 1.6 Wake-Up Radio                                  | 37        |
| 1.7 Modulação OOK                                  | 38        |
| <b>2 METODOLOGIA</b>                               | <b>41</b> |
| 2.1 Arquitetura do Sistema                         | 41        |
| 2.2 Método para Definição do <i>Hardware</i>       | 42        |
| 2.3 Método para Desenvolvimento do <i>Firmware</i> | 42        |
| 2.3.1 Método para Geração de Eventos               | 43        |
| 2.4 Validação do Sistema                           | 43        |
| <b>3 IMPLEMENTAÇÃO</b>                             | <b>44</b> |

|          |   |           |
|----------|---|-----------|
| 3.1      | Definição do Hardware do Gateway e End-Node . . . . .     | 44        |
| 3.1.1    | Módulos Heltec Wi-Fi LoRa (V2) . . . . .                  | 44        |
| 3.1.2    | Transmissor de Rádio Secundário . . . . .                 | 45        |
| 3.1.3    | Receptor de Rádio Secundário . . . . .                    | 46        |
| 3.2      | Desenvolvimento do Firmware dos Módulos . . . . .         | 47        |
| 3.2.1    | Firmware do End-Node . . . . .                            | 47        |
| 3.2.2    | Firmware do Gateway . . . . .                             | 48        |
| 3.3      | Broker MQTT . . . . .                                     | 51        |
| 3.4      | Cadastro dos Dispositivos no Servidor de Rede . . . . .   | 51        |
| 3.4.1    | Registrando uma Aplicação no TTN . . . . .                | 52        |
| 3.4.2    | Registrando o End-Node no TTN . . . . .                   | 53        |
| 3.4.3    | Registrando o Gateway no TTN . . . . .                    | 55        |
| 3.5      | Mensuração do Consumo Energético do Dispositivo . . . . . | 56        |
| <b>4</b> | <b>RESULTADOS E DISCUSSÕES . . . . .</b>                  | <b>58</b> |
|          | <b>CONCLUSÕES . . . . .</b>                               | <b>61</b> |
|          | <b>REFERÊNCIAS . . . . .</b>                              | <b>62</b> |

## INTRODUÇÃO

A Internet das Coisas (IoT) se refere à conexão e troca de dados entre dispositivos e/ou sensores. Atualmente, com o aumento das tecnologias IoT, um número crescente de aplicações práticas pode ser encontrado em muitas áreas, incluindo segurança, rastreamento de ativos, agricultura, medição inteligente, *Smart Cities* e *Smart Homes* (RATASUK; MANGALVEDHE; GHOSH, 2015). Muitas aplicações IoT têm requisitos específicos tais como: longo alcance, baixa taxa de dados, baixo consumo de energia e custo-benefício. As tecnologias sem fio como ZigBee e *Bluetooth Low Energy* (BLE) são amplamente utilizadas, porém não são adaptadas para cenários que requerem transmissão de longo alcance. Soluções baseadas em comunicações celulares (por exemplo, 2G, 3G, 4G) podem fornecer uma cobertura maior, entretanto quando não utilizadas em aplicações de banda estreita, podem possuir um consumo de energia excessivo. Portanto, as exigências das aplicações IoT têm impulsionado o surgimento de novas tecnologias de comunicação sem fio, tal como as redes de área ampla de baixa potência (*Low Power WAN*, LPWAN).

LoRaWAN é um protocolo de rede de baixa potência e área ampla (LPWAN) desenvolvido pela LoRa Alliance (ALLIANCE, 2015). As redes LoRaWAN são tipicamente dispostas em uma topologia estrela de estrelas em que os *gateways* retransmitem mensagens entre os dispositivos finais e um servidor de rede central, o servidor de rede encaminha os pacotes de cada dispositivo da rede para o servidor de aplicação associado (COMMITTEE, 2017c). Os *gateways* são conectados ao servidor de rede através de conexões IP padrão e atuam como uma ponte transparente, simplesmente convertendo pacotes RF para pacotes IP e vice-versa. A comunicação sem fio aproveita as características de longo alcance da camada física LoRa, permitindo uma conexão de um único ponto entre o *end-device* e um ou vários *gateways*.

A especificação LoRaWAN define três classes diferentes de *end-nodes* para atender às diferentes necessidades refletidas na ampla gama de aplicações, sendo elas Classe A, B e C. Todas as classes são capazes de comunicação bidirecional e há suporte para grupos de endereçamento *multicast* para fazer uso eficiente do espectro durante tarefas como atualização de *Firmware Over-The-Air* (OTA) ou outras mensagens de distribuição em massa (ALLIANCE, 2015).

A rede de Classe A é a classe padrão que deve ser suportada por todos os *end-nodes* de uma rede LoRaWAN. A comunicação dessa sempre é iniciada pelo dispositivo final e é totalmente assíncrona. O dispositivo final é capaz de entrar no modo de repouso de baixa potência pelo tempo definido por sua própria aplicação, não há necessidade de rede para o despertador periódico (ALLIANCE, 2015). Isto faz da classe A o modo de operação de baixa potência, enquanto ainda permite a comunicação de *uplink* a qualquer momento, ideal para aplicações cujo dispositivo final possui bateria reduzida (LAVRIC; POPA, 2017). Na rede de Classe B, além das janelas de recepção da Classe A, é aberta uma janela de

recepção extra aberta em horários programados, esta janela é chamada de *ping slot* (DJIDI et al., 2021). Isso fornece à rede a capacidade de enviar comunicações *downlink* com uma latência específica. A rede de Classe C é a caracterizada pelos dispositivos finais sempre prontos para enviar os pacotes de informações, pois o servidor de rede pode iniciar uma transmissão *downlink* a qualquer momento na suposição de que o receptor do dispositivo final está aberto, portanto, sem latência (BOUGUERA et al., 2018). Dessa forma, essa classe é adequada para aplicações onde a alimentação dos dispositivos é feita pela rede elétrica convencional e constante.

Em várias aplicações de LPWANs, o ambiente e as restrições podem não permitir a alimentação do dispositivo pelo meio convencional, isto é, rede elétrica. Assim, é utilizado o armazenamento de energia por meio de baterias. Entretanto, o custo da bateria precisa ser baixo e se a bateria perder energia, a substituição da bateria pode não ser possível em períodos curtos. A LPWAN deve operar com um limite de ciclo de funcionamento rigoroso e muito baixo para que a vida útil do dispositivo possa ser aumentada. Portanto, a operação com energia ultrabaixa é um requisito crucial para dispositivos IoT alimentados por bateria.

Os sistemas de rádio despertador (WuR) apareceram como uma solução para aumentar a eficiência das tecnologias de comunicação, empregando uma interface secundária de rádio de baixa potência, que está sempre no estado ativo e muda o transceptor principal (usado para comunicação de dados principais) desde o modo de economia de energia até o modo de operação ativa (LOPEZ-AGUILERA et al., 2019).

Este objetivo de baixo consumo, impõe restrições à modulação utilizada para o rádio despertador. Se a modulação utilizada possuir uma alta complexidade, como por exemplo a modulação por Chaveamento de Fase (PSK) e Chaveamento de Fase em Quadratura (QPSK), os circuitos de demodulação do WuRx serão mais complexos e exigirão mais potência (MAGNO et al., 2016). O esquema de modulação mais popular usado em implementações de rádio desperta é o *On-Off Keying* (OOK). Esta modulação de baixa complexidade permite uma simplificação drástica do circuito WuR e oferece oportunidades de redução de energia (JELICIC et al., 2012).

Em redes LoRaWAN para que os dispositivos do sistema possam realizar uma comunicação instantânea ou com intervalo de transmissão dos dados em tempo reduzido, é necessário abrir mão do baixo consumo energético, pelo tempo que os dispositivos se manterão ativos trocando mensagens. Uma solução para a redução do consumo energético é o aumento do intervalo de transferência, o que dificulta a coleta de dados em eventos assíncrono. Assim, nota-se a viabilidade em manter os dispositivos em modo de baixo consumo até que enviem seus pacotes de dados, conforme preconizado na especificação de dispositivos LoRaWAN classe A.

Desta forma, este trabalho tem por hipótese a ideia de que por meio do uso da

tecnologia de rádio despertar é possível utilizar uma estratégia para uma comunicação orientada a eventos em uma rede LoRaWAN Classe A capaz de acordar os *end-nodes* a fim de se obter a comunicação de dados de forma assíncrona e um baixo consumo dos dispositivos simultaneamente.

A justificativa é o aprofundamento nas disciplinas do curso de engenharia elétrica, bem como a proposta visa possibilitar ao sistema uma comunicação assíncrona mantendo o baixo consumo dos dispositivos utilizando uma estratégia baseada na tecnologia de *Wake-Up Radio* para uma comunicação orientada a eventos em uma rede LoRaWAN Classe A.

O estudo envolve conceitos estudados durante todo o curso de graduação, possibilitando o envolvimento de três das quatro subáreas do bacharelado em engenharia elétrica, sendo elas: Sistemas Elétricos de Energia; Sistemas e Controle Eletrônico; Eletrônica Industrial e Telecomunicações. As principais disciplinas diretamente relacionadas ao projeto incluem Linguagem de Programação I e II, Eletrônica Analógica I, II e III, Eletrônica Digital I e II, Sistemas Microprocessados, Microcontroladores, Sistemas de Telecomunicações, Eficiência Energética, Redes de Comunicações de Dados I e Comunicação Digital.

Neste contexto, objetivo deste trabalho é desenvolver uma estratégia baseado na tecnologia de *Wake-Up Radio* para comunicação orientada a eventos em uma rede LoRaWAN Classe A. Para isso será necessário realizar estudos sobre estratégias de rádio despertar no contexto de redes LoRaWAN, a montagem de um circuito para acionamento do *end-device* e, por fim, realizar a análise do consumo energético do *end-node*, após a implementação do *Wake-Up Radio*.

Este trabalho está dividido em quatro seções primárias citadas a seguir: Referencial Teórico, Metodologia, Implementação e, por último, Análise e Interpretação dos Resultados.

O primeiro capítulo está destinado ao Referencial Teórico, este consiste na descrição e revisão dos conteúdos necessários para a implementação do projeto. Inicialmente, são introduzidos os conceitos de sistemas embarcados, microcontroladores e paradigma IoT. Em seguida, são abordados conceitos rede de comunicação de dados, comunicação a longa distância e modulação digital que possuem relevância para implementação deste trabalho.

O segundo capítulo define os métodos, as técnicas e os procedimentos utilizados para a concepção do projeto. É explanado qual a modalidade da pesquisa, sua natureza seu objetivo e sua abordagem.

O terceiro capítulo consiste na implementação do projeto, conforme as descrições realizadas nos capítulos anteriores. Este capítulo tem por objetivo apresentar toda metodologia e recursos necessários para implementação do sistema proposto.

O quarto e último capítulo apresenta a análise dos resultados do projeto em conformidade com os objetivos do trabalho. São apresentados os dados coletados durante

as medições de consumo energético do *end-node*.

Por fim, será apresentado uma conclusão com o intuito de finalizar o trabalho com as considerações finais e sugestões para trabalhos da mesma linha. Em seguida pode-se encontrar as referências bibliográficas citadas ao decorrer deste trabalho.

## 1 REFERENCIAL TEÓRICO

Neste capítulo serão apresentados os aspectos teóricos necessários para o melhor entendimento dos conceitos e definições a serem utilizadas ao longo do projeto. Inicialmente, serão abordados os conceitos fundamentais de Sistemas Embarcados, Microcontroladores e Internet das Coisas. Em seguida, serão apresentados aspectos característicos de redes de transmissão de dados e de uma LPWAN. Serão apresentados também elementos da arquitetura de uma rede LoRa, classe de dispositivos, *wake-up radio* e modulação digital OOK.

### 1.1 SISTEMAS EMBARCADOS

Os sistemas embarcados pode ser definidos como circuitos integrados com capacidade computacional, estes dispositivos agem sob o controle de um programa armazenado na memória, executando operações aritméticas, lógica booleana, além de controlar a entrada e saída de dados, permitindo a comunicação com outros dispositivos e periféricos (PEREIRA et al., 2011).

Como os microprocessadores precisam necessariamente de outros dispositivos para seu funcionamento, tais como memória e controladores de entrada e saída de dados, o mercado criou uma necessidade de um sistema de processamento, que já incorpora os dispositivos necessários para seu funcionamento. Dessa forma, o sistema embarcado acaba por ser definido como a combinação de unidades microcontroladas ou microprocessadas com seus respectivos softwares embarcados (CHASE; ALMEIDA, 2007).

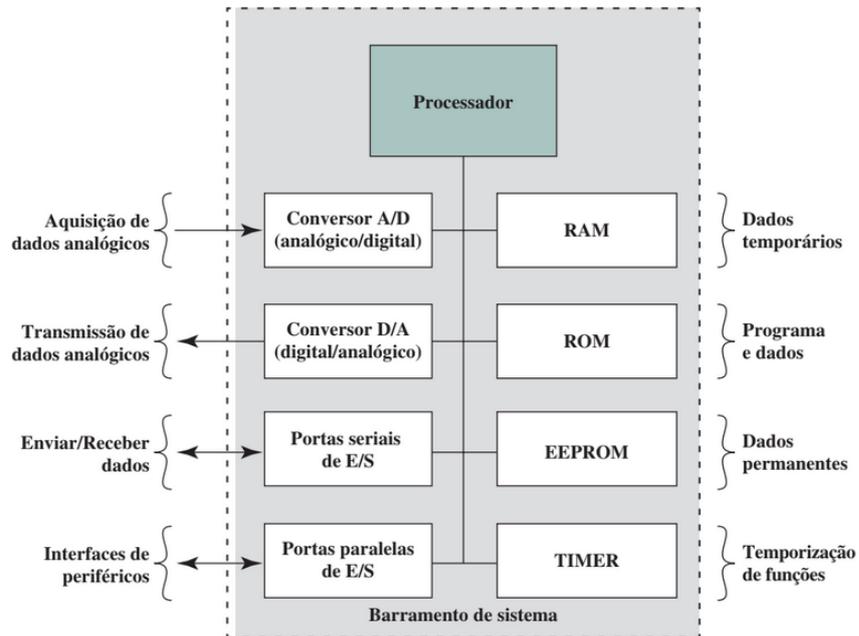
Um consumo de energia elevado inviabiliza o uso de baterias e contribui para o aumento do consumo de energia, quando ligado diretamente à rede. Dessa forma, uma característica muito importante para um sistema embarcado é o baixo consumo de energia (OLIVEIRA, 2017).

### 1.2 MICROCONTROLADORES

Os microcontroladores surgiram ao longo do desenvolvimento dos microprocessadores, em meados de 1970 (OLIVEIRA, 2017). Os microprocessadores são circuitos que têm como principal função a lógica de processamento de instruções, ou seja, agrega em si a função de tomada de decisão computacional (STALLINGS, 2017).

Microcontroladores são circuitos computacionais compostos de elementos como processador, portas de entrada e saída (*I/O*), memória RAM (*Random Access Memory*), memória EEPROM (*Electrically Erasable Programmable Read-Only Memory*), interface de comunicação e interface de rede (OLIVEIRA, 2017). A figura 1 apresenta este e outros elementos tipicamente encontrados em microcontroladores.

Figura 1 – Elementos Típicos de um Microcontrolador

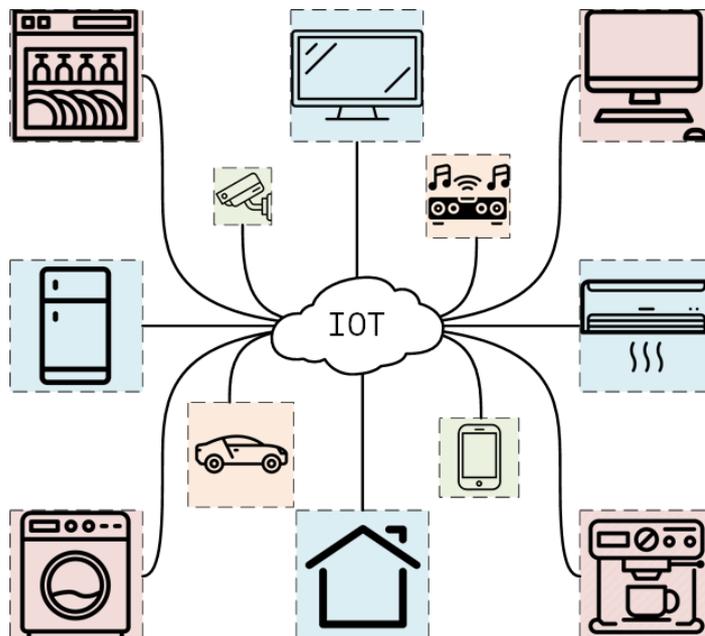


Fonte: (STALLINGS, 2017)

### 1.3 INTERNET DAS COISAS

A Internet das Coisas (*Internt of Things*, IoT) se tornou muito popular por possibilitar que diversos dispositivos presentes um sistema tenham conexão com a internet, seja por meio de uma rede cabeada ou sem fio. Dessa forma conceitos de IoT estão sendo utilizados em diversas áreas como no transporte, agropecuária, indústria, saúde e em conceitos de *Smart Cities* (GERMANI et al., 2019). A figura 2 apresenta o conceito de IoT, que interliga diversos dispositivos comuns ao nosso cotidiano por meio da internet.

Figura 2 – Apresentação Paradigma de IoT



Fonte: Autoria própria

Seguindo a evolução da tecnologia, cada vez mais a capacidade computacional, armazenamento e bateria se tornam mais acessíveis. Esta tendência está permitindo o desenvolvimento de dispositivos eletrônicos de pequena escala com capacidade de identificação/comunicação/computação, que poderiam ser incorporados a outros dispositivos, sistemas e instalações (CHEN et al., 2014).

## 1.4 REDE DE COMUNICAÇÃO DE DADOS

A comunicação de dados se refere a trocas de dados entre dois dispositivos por intermédio de algum tipo de meio de comunicação, seja por meio de cabos condutores ou tecnologias sem fio. Uma rede de comunicação é todo um conjunto de dispositivos, conhecidos como nós, conectados por meio de *links* de comunicação. Um nó pode ser um computador, um telefone ou qualquer outro dispositivo conectado a rede capaz de realiza o envio e/ou recepção de dados para outros nós (FOROUZAN, 2009).

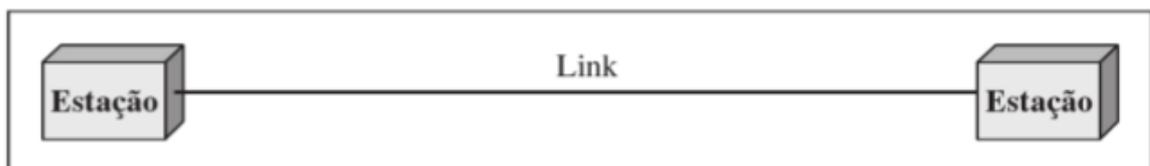
Não existem taxionomia de aceitação geral na qual todas as redes de computadores possam ser classificadas, entretanto duas se destacam das demais, são elas a Tecnologia e a Escala de Transmissão (TANENBAUM; WETHERALL, 2011).

### 1.4.1 Tecnologia de Transmissão de Dados

Para ocorrer a comunicação em uma rede é necessário que dois ou mais dispositivos estejam conectados de alguma maneira ao mesmo *link* ao mesmo tempo. Existem dois tipos de conexão para os nós de forma a possibilitar a transmissão dos dados: enlaces ponto a ponto e enlaces difusão (*broadcast*) (FOROUZAN, 2009).

Uma conexão ponto a ponto fornece um link dedicado entre dois dispositivos ou em redes de comunicação par a par. A rede do tipo *peer-to-peer* funciona de forma que para ir da origem ao destino, um pacote deve visitar uma ou mais máquinas intermediárias (MACEDO et al., 2018). A figura 3 apresenta uma rede de conexão ponto a ponto.

Figura 3 – Topologia de Rede de Enlace Ponto a Ponto

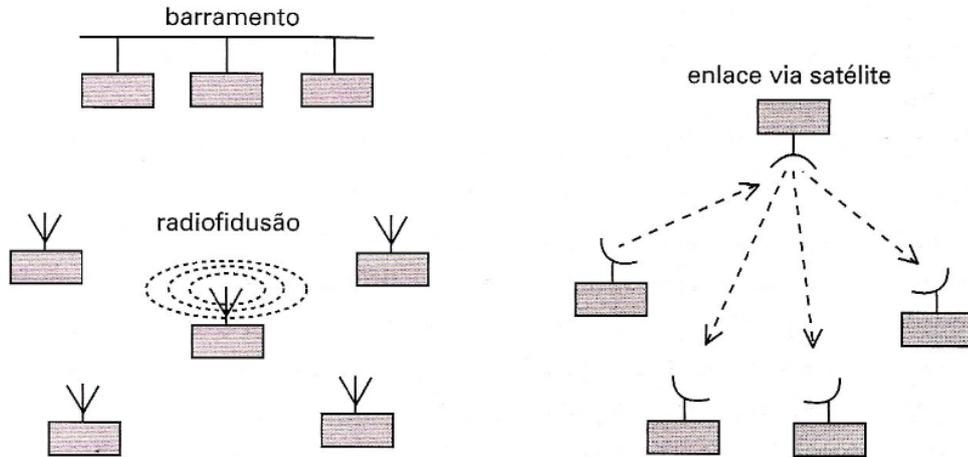


Fonte: (FOROUZAN, 2009)

As redes *broadcast* têm apenas um canal de comunicação que é compartilhado entre todos os dispositivos da rede. Os pacotes enviados por qualquer máquina são recebidos por todas as outras. Entretanto, um campo de endereço dentro do pacote especifica o destinatário pretendido. Assim, quando um nó recebe um pacote, ele apenas será

processado se o endereço corresponder à máquina receptora, caso contrário será ignorado (TANENBAUM; WETHERALL, 2011). A figura 4 apresenta exemplos de topologias de redes baseadas em enlaces por difusão.

Figura 4 – Topologias de Redes de Enlace por Difusão



Fonte: (ALBINI, 2015)

#### 1.4.2 Escala de Transmissão de Dados

Para essa técnica de classificação, a distância é uma métrica importante, pois em cada caso são utilizadas tecnologias diferentes conforme a escala. Dessa forma, as redes de transmissão de dados podem ser classificadas em: Rede pessoal, são redes destinadas a uma única pessoa; Rede local, é uma rede privada que opera dentro e próximo de um único prédio ou residência; Rede metropolitana, é uma rede que estende-se ao longo de uma cidade; Rede de longas distâncias, trata-se de uma rede que abrange uma grande área geográfica, como um país ou um continente (TANENBAUM; WETHERALL, 2011). A figura 5 apresenta as classificações de escalas de redes conforme o seu tamanho físico aproximado.

Figura 5 – Escalas de Redes de Transmissão de Dados

| Distância do interprocessador | Processadores localizados no mesmo | Exemplo                    |
|-------------------------------|------------------------------------|----------------------------|
| 1 m                           | Metro quadrado                     | Área pessoal               |
| 10 m                          | Cômodo                             | } Rede local               |
| 100 m                         | Prédio                             |                            |
| 1 km                          | Campus                             |                            |
| 10 km                         | Cidade                             | Rede metropolitana         |
| 100 km                        | País                               | } Rede a longas distâncias |
| 1.000 km                      | Continente                         |                            |
| 10.000 km                     | Planeta                            | A Internet                 |

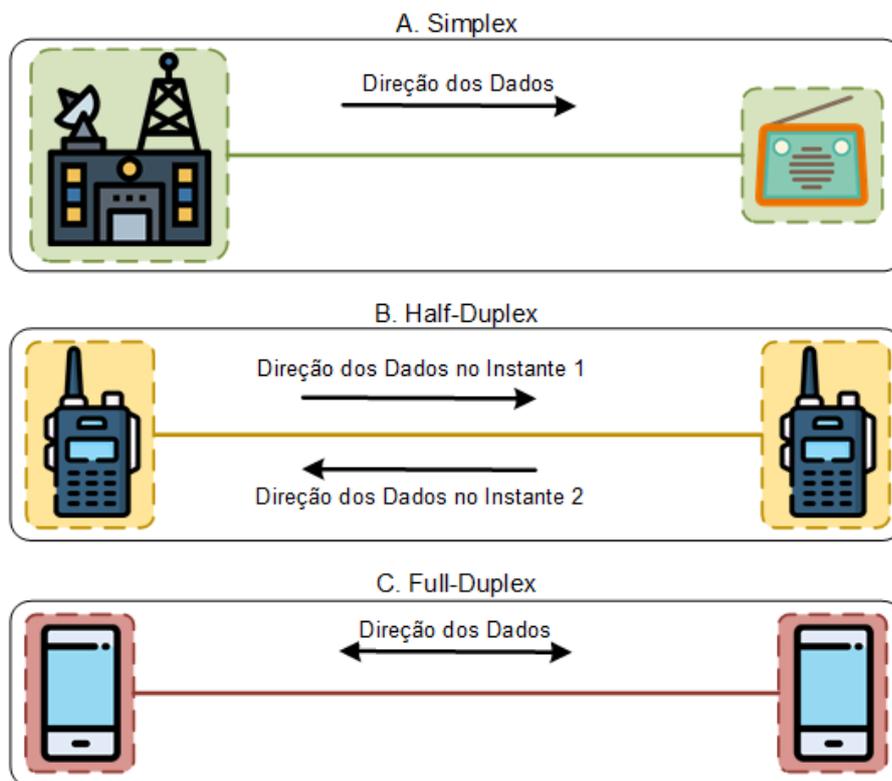
Fonte: (TANENBAUM; WETHERALL, 2011)

### 1.4.3 Tipos de Comunicação

Para que seja possível estabelecer a comunicação entre dois nós são necessários caminhos de transmissão entre os elementos transmissores e os receptores por meio de canais. Os canais de comunicação entre dispositivos pode ser simplex, half-duplex ou full-duplex (FOROUZAN, 2009).

Os enlaces simplex consiste em enlaces que permitem o tráfego em apenas um sentido. No modo half-duplex, a comunicação pode ocorrer em ambos os sentidos, contudo não de forma simultânea. Os enlaces full-duplex podem ser usados nos dois sentidos ao mesmo tempo (TANENBAUM; WETHERALL, 2011). A figura 6 demonstra os tipos de comunicação e suas respectivas formas de fluxos de dados.

Figura 6 – Fluxo de Dados (Simplex, Half-Duplex e Full-Duplex)



Fonte: Autoria própria

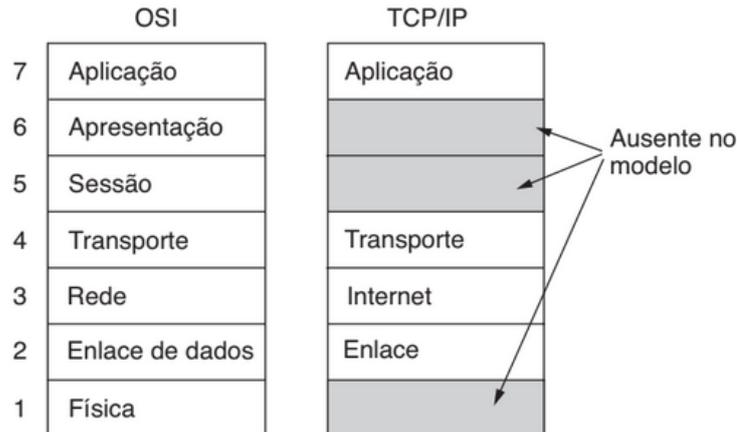
### 1.4.4 Arquitetura de Redes

As redes de computadores são criadas por diversas entidades. Dessa forma são necessários padrões de forma que essas diferentes redes possam se comunicar. Os dois padrões mais conhecidos são o modelo OSI e o modelo TCP/IP (FOROUZAN, 2009).

O modelo OSI baseia-se em uma proposta desenvolvida pela ISO (*International Organization for Standardization*) para padronização internacional dos protocolos, este modelo possui 7 camadas e cada uma possui sua própria lista de protocolos e serviços. O

modelo TCP/IP é composto por cinco camadas e tem características opostas ao modelo OSI: o modelo propriamente dito não é muito utilizado, porém seus protocolos são mais implementados (TANENBAUM; WETHERALL, 2011). A figura 7 apresenta as camadas de cada arquitetura de rede.

Figura 7 – Camadas da Arquitetura OSI e TCP/IP



Fonte: (TANENBAUM; WETHERALL, 2011)

#### 1.4.5 MQTT

O MQTT é um protocolo de comunicação caracterizado por sua leveza e maneira simples de distribuir a informação em rede, que foi desenvolvido pela IBM na década de 90 e projetado especificamente para comunicação M2M (*Machine-to-Machine*) (SONI; MAKWANA, 2017). Este protocolo utiliza as camadas TCP/IP e possui o cabeçalho do pacote de dados com o tamanho mínimo de 2 bytes, o que implica em um consumo menor de energia pelo dispositivo (ATMOKO; RIANINI; HASIN, 2017).

Além disso, o protocolo funciona em um sistema servidor-cliente, onde o servidor é chamado de *broker*. Os clientes não enviam mensagens diretamente um ao outro, em vez disso contam com o intermédio do *broker*. Cada mensagem MQTT contém um tópico o qual os clientes podem se inscrever ou publicar. Este tópico funciona como um filtro, pois dessa forma o *broker* consegue verificar os dispositivos que estão interessados no recebimento das mensagens deste tópico específico (DINCULEANĂ; CHENG, 2019).

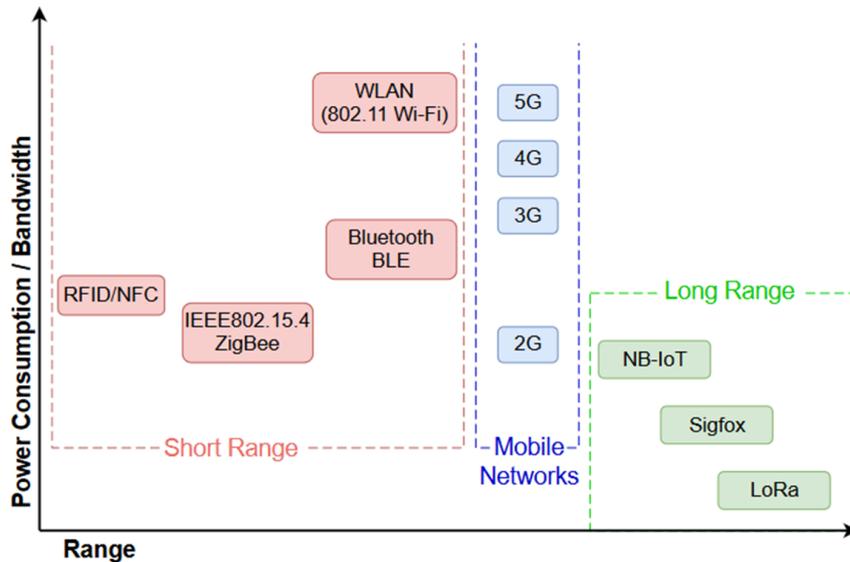
## 1.5 COMUNICAÇÃO DE LONGO ALCANCE

### 1.5.1 LPWAN

LPWAN é uma classe de rede de comunicação sem fio para soluções de IoT com características como grandes áreas de cobertura, baixas taxas de transmissão de dados com pequenos tamanhos de pacotes de dados e longa vida útil da bateria. Os dispositivos

podem abranger desde distâncias muito curtas a remotas, de posições estacionárias a móveis, de conexões de baixa energia baseadas em bateria a conexões baseadas em energia comercial, e uma gama de ambientes amigáveis a ambientes hostis. A faixa de cobertura para algumas aplicações é altamente localizada, por exemplo, implantações estacionárias internas (CHAUDHARI; ZENNARO; BORKAR, 2020). A figura 8 demonstra a o consumo energético em relação ao alcance de algumas tecnologias de comunicação sem fio.

Figura 8 – Relação Consumo x Alcance de tecnologias de comunicação sem fio



Fonte: (TORRES; PINTO; LOPES, 2021)

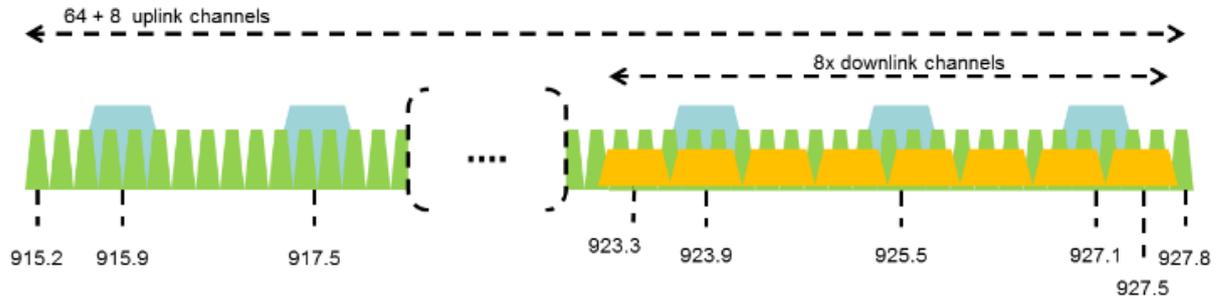
As redes LPWAN tem a característica de operação em bandas de frequências não licenciadas, estas bandas são reservadas para o estudo e implementação na área industrial, científica e médica (ISM) (HUANG; CHEN, 2013). Elas compreendem as faixas de 902 MHz a 907,5 MHz, 915 MHz a 928 MHz, 2,4 a 2,48 GHz e 5,72 a 5,85 GHz (CUNHA, 2021).

### 1.5.2 LoRa

A palavra LoRa vem do inglês *Long Range*, é uma tecnologia que permite comunicação a longas distâncias com consumo mínimo de energia. Ela proporciona uma comunicação com alcance de até 10 a 40 km em zonas rurais e de 1 a 5 km em zonas urbanas (CENTENARO et al., 2016). Essa técnica é utilizada na comunicação militar e espacial permitindo cobrir longas distâncias de comunicação e alcançar robustez à interferência (ALLIANCE, 2015).

O órgão regulamentador para as comunicações sem fio brasileira é a Agência Nacional de Telecomunicações (ANATEL). No Brasil é adotado o padrão australiano (AU915), cuja faixa de frequência estabelecida é de 915 a 928MHz (ROSSATO; SPANHOL; CAMARGO, 2020). A figura 9 apresenta a subdivisão dos canais do plano AU915.

Figura 9 – Frequências dos Canais do Plano AU915

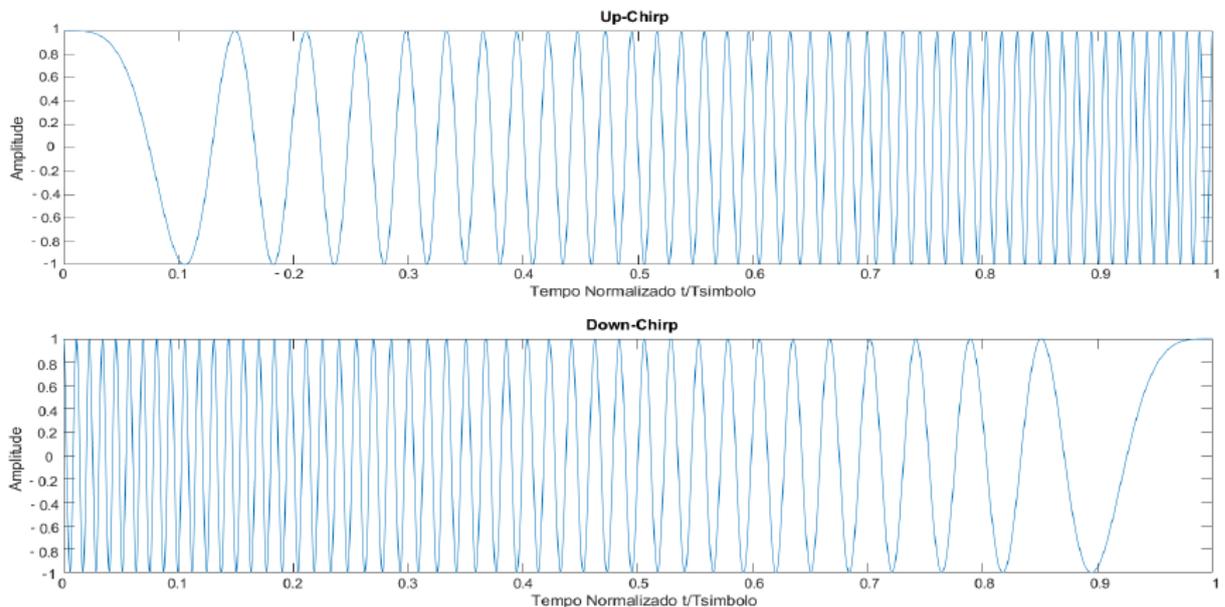


Fonte: (COMMITTEE, 2017b)

### 1.5.2.1 A Camada Física do LoRa

O LoRa é uma tecnologia que realiza modulação de radiofrequência baseada na técnica de modulação por espalhamento espectral de chirp (*Chirp Spread Spectrum*), que permite comunicação a longas distâncias com baixo consumo de energia e alta robustez a efeitos de interferências (SANTOS; CUNHA, 2019). Durante sua transmissão, quando a frequência vai de uma frequência baixa para alta se tem um *Up-Chirp* e se ela vai de uma frequência alta para baixa temos um *Down-Chirp*. A figura 10 apresenta a representação de um *Up-Chirp* e *Down-Chirp* no domínio do tempo.

Figura 10 – Up e Down Chirp no Domínio do Tempo



Fonte: (SANTOS; CUNHA, 2019)

O motivo do uso dessa modulação é explicado pela teorema da capacidade do canal de Claude E. Shannon. Pelo teorema, a capacidade de transmissão de um canal depende da relação sinal/ruído e da largura de banda do canal conforme apresentado da equação 1:

$$c = B * \log_2\left(1 + \frac{S}{N}\right) \quad (1)$$

Onde se tem:

- $c \rightarrow$  Capacidade do Canal em bits/s
- $B \rightarrow$  Largura de Banda em Hz
- $\frac{S}{N} \rightarrow$  Relação Sinal Ruído em dB

Para utilizar a modulação LoRa é necessário configurar três parâmetros no dispositivo final, sendo elas Largura de Banda, Fator de Espalhamento e Taxa de Código. Essas configurações determinam a quantidade de bits por segundo (*bit rate*), o tamanho máximo do pacote de informações e o tempo que o pacote permanece no ar, sendo este fator importante para que se tenha um maior alcance ou enviar uma quantidade maior de bits na mensagem (PEREIRA et al., 2011).

#### 1.5.2.2 Largura de Banda

A largura de banda ou *bandwidth* representa a faixa de frequência na banda de transmissão e pode assumir três valores determinados, 125 kHz, 250 kHz ou 500 kHz. Para se obter um maior alcance um menor valor de frequência deve ser utilizado (BOUGUERA et al., 2018).

#### 1.5.2.3 Taxa de Codificação

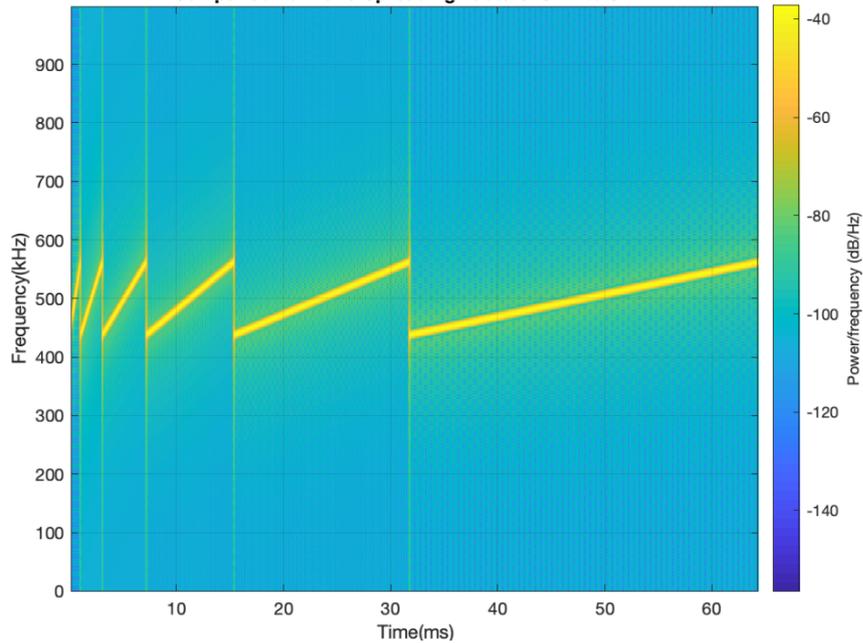
Taxa de Codificação ou *Coding Rate* indica o número de bits destinados para dados de redundância na mensagem, a fim de realizar a recuperação de erros, estipulados com os respectivos valores de CR: 4/5, 4/6, 4/7 e 4/8. Incrementar o CR aumenta a proteção contra interferências, mas também o tempo do bit no ar (BOUGUERA et al., 2018). A equação 2 define a taxa de codificação.

$$CR = \frac{4}{4+n}, \text{ com } n \in \{1, 2, 3, 4\} \quad (2)$$

#### 1.5.2.4 Fator de Espalhamento

Apresentando valores possíveis de 7 à 12, o fator de espalhamento ou *Spreading Factor* influencia na taxa de dados tempo no ar, consumo de energia e alcance da comunicação. O fator de espalhamento determina o número de *chirps* que a portadora dos dados enviará a cada segundo, ou seja é a representação do número de bits de dados codificados em um único símbolo (KIM; LEE; JEON, 2020). A figura 11 apresenta a comparação do fator de espalhamento de SF7 à SF12.

Figura 11 – Comparação do Fator de Espalhamento: SF7 à SF12



Fonte: (KIM; LEE; KIM, 2019)

O número de *chirps* por símbolo pode ser calculado como mostrado na equação 3.

$$M = 2^{SF} \quad (3)$$

Onde se tem:

- $M \rightarrow$  Número de Chirps por Símbolo
- $SF \rightarrow$  Fator de Espalhamento, sendo  $SF \in \{7, 8, 9, 10, 11, 12\}$

Quanto maior o valor para o fator de espalhamento, maior será a capacidade de cobertura do sinal e menor será a taxa de transmissão (SANTOS; CUNHA, 2019). A taxa de bits pode ser calculada por meio da equação 4.

$$R_b = SF * \frac{2^{SF}}{BW} * CR \quad (4)$$

Onde se tem:

- $R_b \rightarrow$  Taxa de Bits em bits/símbolo
- $SF \rightarrow$  Fator de Espalhamento
- $BW \rightarrow$  Largura de Banda
- $CR \rightarrow$  Taxa de Codificação

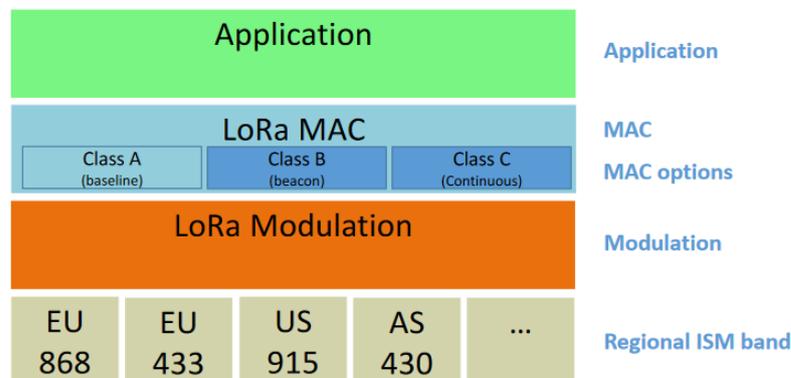
### 1.5.2.5 Data Rate

O *Data Rate* trata-se de um índice numerado de 0 à 15, o mesmo é associado a combinação específica de fator de espalhamento e largura de banda, onde cada combinação resulta em uma taxa de dados diferente. As configurações de DR nos dispositivos finais podem ser gerenciadas automaticamente pelo *network server* por meio do recurso *Adaptive Data Rate* (ADR). Quando ativo, a taxa de dados de cada *end-device* e a potência de transmissão são ajustadas com o objetivo de otimizar o consumo do dispositivo (CAVALLI et al., 2021).

### 1.5.3 LoRaWAN

LoRaWAN é um protocolo aberto de comunicação que funciona sobre a camada física LoRa. O LoRaWAN implementa os detalhes de funcionamento, segurança, qualidade do serviço, ajustes de potência visando maximizar a duração da bateria dos módulos, e os tipos de aplicações tanto do lado do módulo quanto do servidor (ALLIANCE, 2015). A figura 12 apresenta as camadas do protocolo LoRaWAN.

Figura 12 – Camadas do Protocolo LoRaWAN



Fonte: (COMMITTEE, 2017c)

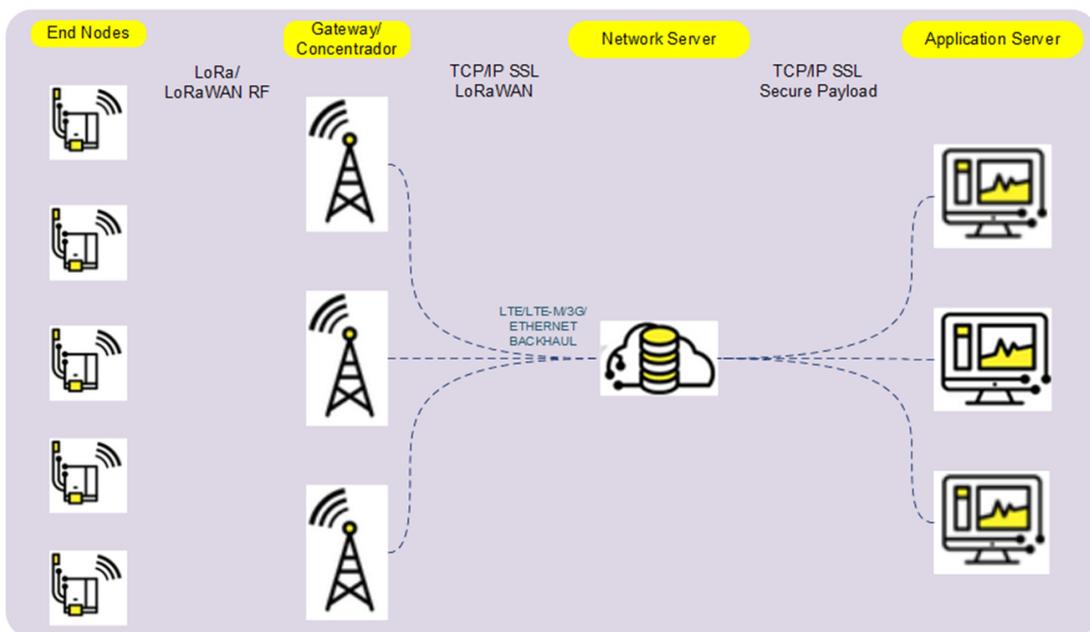
Usando LoRaWAN, cada mensagem transmitida por um dispositivo final é recebida por todas as estações rádio base na faixa. Ao explorar esta recepção redundante, LoRaWAN melhora a taxa de sucesso das mensagens recebidas. Entretanto, a obtenção desta característica requer várias estações rádio base na vizinhança, o que pode aumentar o custo de implantação da rede. As recepções duplicadas resultantes são filtradas no sistema *backend* (servidor de rede) que também tem a inteligência necessária para verificar a segurança, enviar agradecimentos ao dispositivo final e enviar a mensagem para o servidor de aplicação correspondente.

Além disso, múltiplas recepções da mesma mensagem por diferentes estações base são exploradas pelo LoRaWAN para a localização dos dispositivos finais. Para este fim, é utilizada a técnica de localização baseada na diferença de tempo de chegada (*Time*

*Difference of Arrival – TDOA*), suportada por sincronização de tempo muito precisa entre múltiplas estações rádio base. Além disso, múltiplas recepções da mesma mensagem em estações rádio base diferentes evitam a entrega na rede LoRaWAN (ou seja, se um nó é móvel ou está em movimento, não é necessário enviar novamente a mensagem entre as estações rádio base).

Esse protocolo fornece um mecanismo que permite que diversos dispositivos se comuniquem com um único *gateway* através do MAC (Medium Access Control). Os elementos da arquitetura se organizam de forma que os dispositivos finais enviam informações para o *gateway*, que conecta os dispositivos na rede por meio de LoRaWAN, o servidor de rede gerencia a comunicação dos dispositivos com o servidor de aplicação através da internet e o servidor de aplicação exibe as informações vindas dos dispositivos para o usuário final (ALLIANCE, 2015). A estrutura desta rede é complexa, pois sua arquitetura forma uma topologia estrela de estrelas, como mostra a figura 13.

Figura 13 – Arquitetura de uma rede LoRa



Fonte: Autoria própria

### 1.5.3.1 Servidor de Rede

O servidor de rede é responsável pelo controle da rede LoRaWAN e comandos MAC para gerenciamento da sessão, reconhecimento, eliminação de pacotes duplicados e ADR. O *network server* se comunica com múltiplas entidades, tais como gateways, servidor de aplicações e banco de dados, também é responsável por descriptografar os pacotes de uplink validando ou descartando em caso de inconsistência (MEHIC et al., 2022). Para soluções de servidores de rede há várias opções, dentre elas duas destaca:

- ChirpStack: Fornece componentes de código aberto para LoRaWAN. O servidor apresenta uma interface web para gerenciamento de dispositivos e APIs para integração.
- The Things Network: fornece um conjunto de ferramentas abertas e uma infraestrutura de rede LoRaWAN pronta para uso. Realiza o cadastro de gateways, dispositivos finais e faz a integração com servidores de aplicação

### 1.5.3.2 Servidor de Aplicação

O servidor de aplicação é responsável pelo gerenciamento de todas as chaves da rede (geração e armazenamento), recebimento dos pacotes vindos do *network server* e apresentação dos dados monitorados pelos dispositivos finais (PASTÓRIO et al., 2021).

### 1.5.3.3 Gateway

O *gateway*, concentrador ou ponto de acesso, é responsável por estabelecer a conexão, receber e encaminhar mensagens com dispositivos finais via radio, usando a tecnologia de transmissão LoRa e o protocolo LoRaWAN. O mesmo repassa as informações recebidas para o servidor de rede via TCP/IP (Wi-Fi, 3G, 4G, entre outros) (SELLER, 2021).

### 1.5.3.4 Dispositivos Finais

Módulos, end-nodes, end-points ou end-devices, são conhecidos como dispositivos eletrônicos capazes de realizar alguma ação, leitura de dados, entre outras possibilidades (FAN et al., 2022).

### 1.5.3.5 Tipos de Ativação

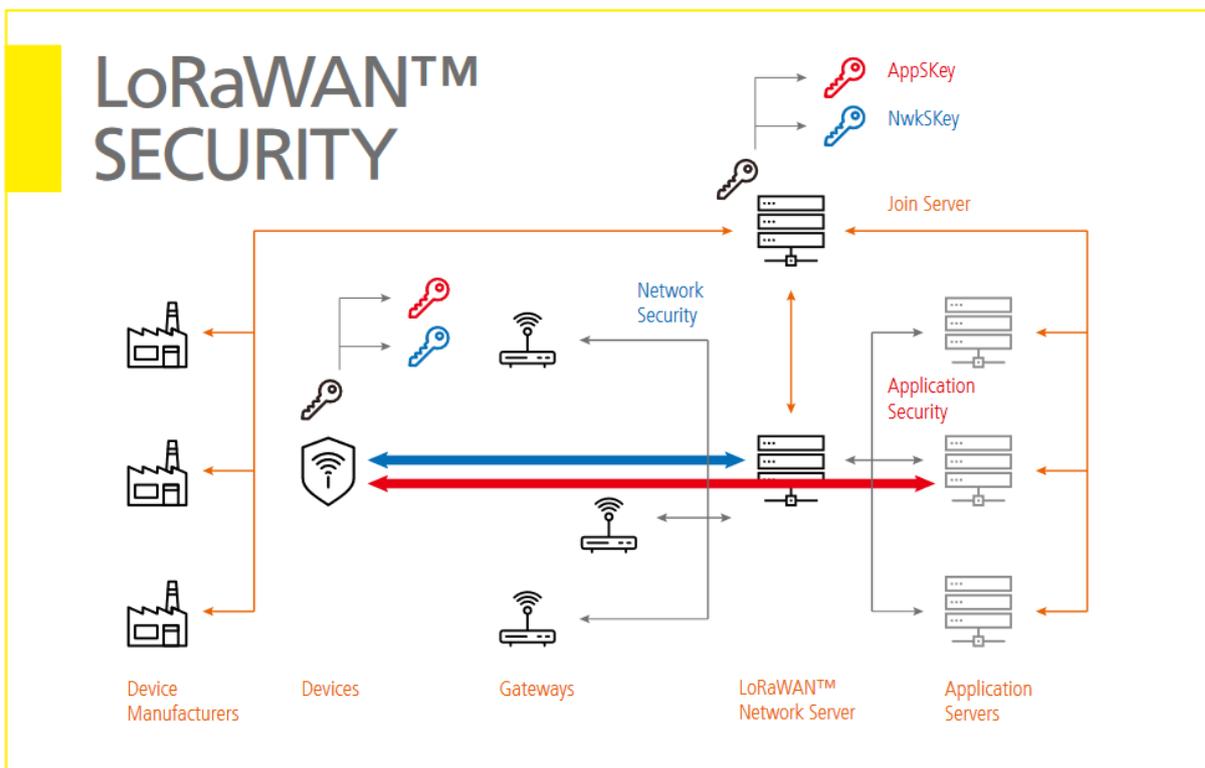
LoRaWAN oferece criptografia do tipo *Advanced Encryption Standard* de 128 bits (AES-128) em dois níveis: o primeiro localizado na camada de rede que estabelece a segurança e confidencialidade dos dados entre *end-node* e *network server* por meio da chave *Network Session Key* (NwkSKey), e o segundo encontra-se na camada de aplicação, que estabelece a criptografia de ponta-a-ponta entre o dispositivo final e a aplicação por meio da chave *Application Session Key* (AppSKey). A ativação pode ser realizada por dois tipos de autenticação: Over The Air Activation (OTAA) e Activation By Personalization (ABP) (PASTÓRIO et al., 2021).

A forma de ativação OTAA é complexa, pois os dispositivos são integrados a rede por meio de um processo chamado *join*. Os parâmetros de NwkSKey, AppSKey e DevAddr

não são fixos, são gerados automaticamente e redefinidos sempre que o dispositivo realiza um novo *join*. O processo é iniciado pelo end-node, por meio do envio de uma mensagem de *Join Request*, após a validação pelo *network server* a mensagem é respondida com um *Join Accept* (MEHIC et al., 2022).

Para realizar a ativação é necessário que três informações sejam pré-configuradas na memória do end-device, sendo elas: *End-Device Unique Identifier* (DevEUI), *Join-Server Unique Identifier* (JoinEUI), e *Application Key* (AppKey) (CAVALLI et al., 2021). O DevEUI trata-se do identificador único de fábrica que deve ser conhecido pelo *network server* no qual o dispositivo final pretende ser integrado, já o JoinEUI é um identificador global da aplicação cuja função é, junto ao AppKey, derivar as chaves NwkSKey e AppSKey (HAN; WANG, 2018). Ambos são identificadores de 64 bits que seguem o padrão do Institute of Electrical and Electronics Engineers (IEEE) EUI64 (CAVALLI et al., 2021). A figura 14 apresenta a distribuição das chaves de ativação para OTAA.

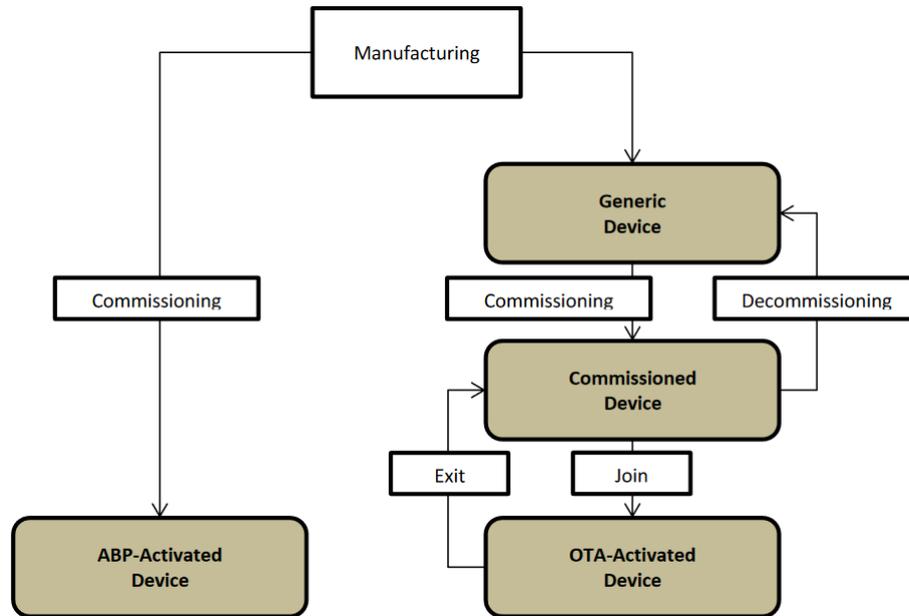
Figura 14 – Diagrama de Distribuição das Chaves de Ativação para OTAA



Fonte: (GEMALTO; SEMTECH, 2017)

A ativação ABP tem a vantagem de ser mais simples, pois não realiza o processo de join e as chaves de segurança e o endereço são definidos de maneira personalizada no momento do início do funcionamento e permanecem as mesmas durante toda a vida útil do dispositivo (MEHIC et al., 2022). Tratam-se de duas chaves de criptografia, NwkSKey e AppSKey, e um endereço de 32 bits chamado *Device Address* (DevAddr), que identifica o *end-device* na rede (CAVALLI et al., 2021). A figura 15 ilustra os processos de ativação.

Figura 15 – Diagrama de funcionamento das Ativações ABP e OTAA

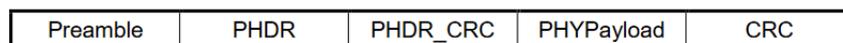


Fonte: (COMMITTEE, 2017a)

### 1.5.3.6 Formato dos Frames

Em uma rede LoRaWAN, as mensagens de *uplink* são enviadas de um dispositivo final ao servidor de rede, por meio de gateways que a receberam. Esses pacotes são compostos por um preâmbulo de sincronização (*preamble*), um cabeçalho da camada física LoRa (*LoRa Physical Header* ou PHDR), um cabeçalho CRC próprio de 16 bits denominado de PHDR\_CRC. Desse modo, ao receber o pacote o *gateway* valida primeiro este cabeçalho antes de seguir para a demodulação da informação. Para garantir a integridade da PHYPayload da mensagem é incluído um CRC de 16 bits sobre esta (COMMITTEE, 2017c). A figura 16 apresenta a composição de uma mensagem *uplink* da camada física.

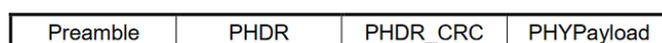
Figura 16 – Composição de uma Mensagem de Uplink



Fonte: (COMMITTEE, 2017c)

Cada mensagem de *downlink* é enviada pelo servidor de rede de um único *gateway* para apenas um dispositivo final. As mensagens *downlink* usam o modo explícito do pacote de rádio no qual o cabeçalho físico LoRa (PHDR) e um cabeçalho CRC (PHDR\_CRC) estão incluídos (COMMITTEE, 2017c). A figura 17 apresenta a composição de uma mensagem *downlink* da camada física.

Figura 17 – Composição de uma Mensagem de Downlink

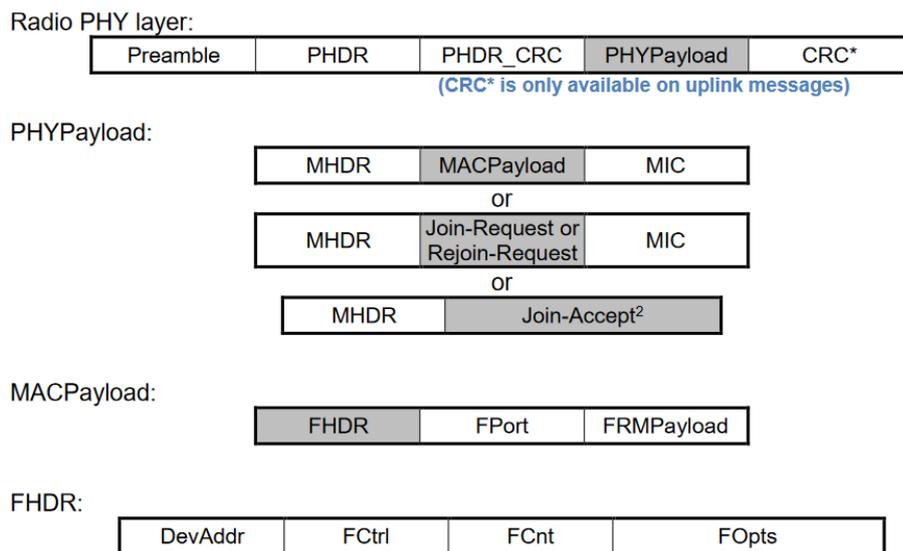


Fonte: (COMMITTEE, 2017c)

A PHYPayload é formada por um cabeçalho MAC (*MAC Header*, MHDR), que consiste em um cabeçalho que identifica se o pacote é de *uplink* ou *downlink*, se é confirmado ou não confirmado, se é um tipo de *Join Request* ou *Join Accept*, uma carga útil denominada de MACPayload que varia de acordo com as informações presentes no MHDR e um código de integridade de mensagem (MIC), utilizado tanto na verificação da integridade quanto na autenticação da mensagem no servidor da rede (CAVALLI et al., 2021).

Todas as mensagens LoRaWAN uplink e downlink carregam um *payload* da camada física que começa com um cabeçalho MAC de um octeto (MHDR), seguido por um *payload* MAC (MACPayload), e terminando com um código de integridade de mensagem de 4 octetos (MIC). A figura 18 apresenta o formato de todos os elementos de uma mensagem LoRaWAN.

Figura 18 – Estrutura de uma Mensagem LoRaWAN



Fonte: (COMMITTEE, 2017c)

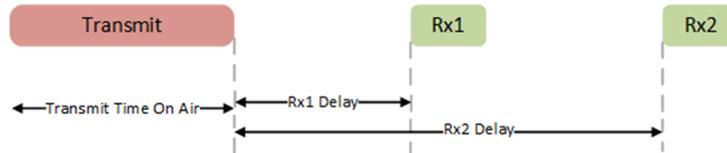
### 1.5.3.7 Classe de Dispositivos

A especificação LoRaWAN define três classes diferentes de end-nodes para atender às diferentes necessidades refletidas na ampla gama de aplicações sendo elas: classe A, B, e C.

A classe A é a classe padrão que deve ser suportada por todos os end-nodes de uma rede LoRaWAN. A comunicação da classe A sempre é iniciada pelo dispositivo final e é totalmente assíncrona. Cada transmissão *uplink* pode ser enviada a qualquer momento e é seguida por duas breves janelas de *downlink*. Como a comunicação *downlink* deve sempre seguir uma transmissão *uplink* com um horário definido pela aplicação do dispositivo final, a comunicação *downlink* deve ser guardada no servidor de rede até o próximo evento de *uplink* (COMMITTEE, 2017c). Esta classe tem o menor consumo de energia do dispositivo final, mas a maior latência de *downlink* do servidor (BOUGUERA et al., 2018). A figura 19

exemplifica o funcionamento desta classe para o melhor entendimento do seu processo de envio de dados.

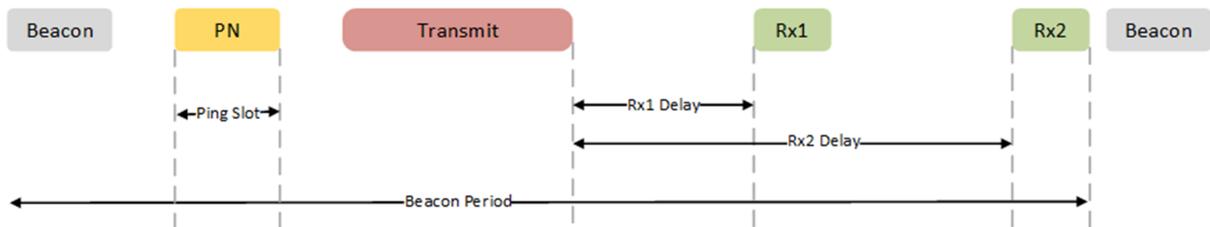
Figura 19 – Janela de transmissão de uma rede LoRaWAN Classe A



Fonte: Autoria própria

Na classe B, além das janelas de recepção da Classe A, é aberta uma janela de recepção extra aberta em horários programados, esta janela é chamada de *ping slot* (DJIDI et al., 2021). Para isso, o dispositivo final recebe um sinal sincronizado, enviado pelo *gateway*. Isto fornece à rede a capacidade de enviar comunicações *downlink* com uma latência específica, mas às custas de algum consumo de energia adicional no dispositivo final. A latência é programável em até 128 segundos para atender diferentes aplicações e o consumo de energia adicional é baixo o suficiente para ainda ser válido para aplicações alimentadas por bateria (ALLIANCE, 2015). O funcionamento do processo de envio de dados desta classe é apresentado na figura 20.

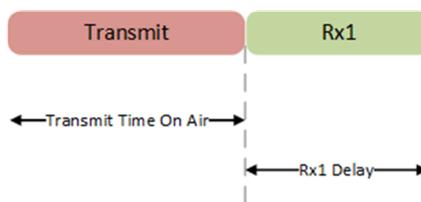
Figura 20 – Janela de transmissão de uma rede LoRaWAN Classe B



Fonte: Autoria própria

A classe C é caracterizada pelos dispositivos finais sempre prontos para enviar os pacotes de informações, pois o servidor de rede pode iniciar uma transmissão *downlink* a qualquer momento na suposição de que o receptor do dispositivo final está aberto, portanto, sem latência (BOUGUERA et al., 2018). Dessa forma, essa classe é adequada para aplicações onde a alimentação dos dispositivos é feita pela rede elétrica convencional e constante. A figura 21 apresenta um exemplo de como funciona o processo de envio de transmissão de pacotes em redes LoRaWAN de classe C.

Figura 21 – Janela de transmissão de uma rede LoRaWAN Classe C



Fonte: Autoria própria

Dessa forma, a tabela 1 apresenta resumidamente as principais características de cada uma das classes supracitadas.

Tabela 1 – Características de cada classes de Dispositivos

| Classe A                                    | Classe B  | Classe C   |
|---|---|--|
| Alimentação por bateria                     | Baixa Latência  | Sem Latência   |
| Bidirecional                                | Bidirecional com slots de recebimento programado                      | Bidirecional   |
| Mensagens <i>Unicast</i>                    | Mensagens <i>Unicast</i> e <i>Multicast</i>                           | Mensagens <i>Unicast</i> e <i>Multicast</i>          |
| Pacotes pequenos e em um longo intervalo    | Pacotes pequenos e em um longo intervalo, duração de envio específico | Pacotes pequenos                                     |
| Os <i>end-devices</i> iniciam a comunicação | Janela extra de recepção  | O servidor solicita a transmissão a qualquer momento |

Fonte: Autoria própria

## 1.6 WAKE-UP RADIO

Dispositivo IoT não sabe quando esperar receber dados e, portanto, deve realizar uma varredura de drenagem de energia para detectar a presença de um pacote. Neste cenário, se tem a possibilidade do uso de um receptor em modo de consumo muito baixo que acordaria a tempo de todos os dados serem recebidos.

Os WuRx são receptores de rádio que permanecem ligados, dessa forma precisam funcionar com consumo de energia ultra baixo, o que exige uma arquitetura muito simples com apenas alguns componentes discretos são suficientes para construir o circuito de detecção de sinais (PIYARE et al., 2018). Por ter um consumo de energia muito baixo, o WuRx pode estar permanentemente ativo sem perdas significativas em termos de duração da bateria (LOPEZ-AGUILERA et al., 2019).

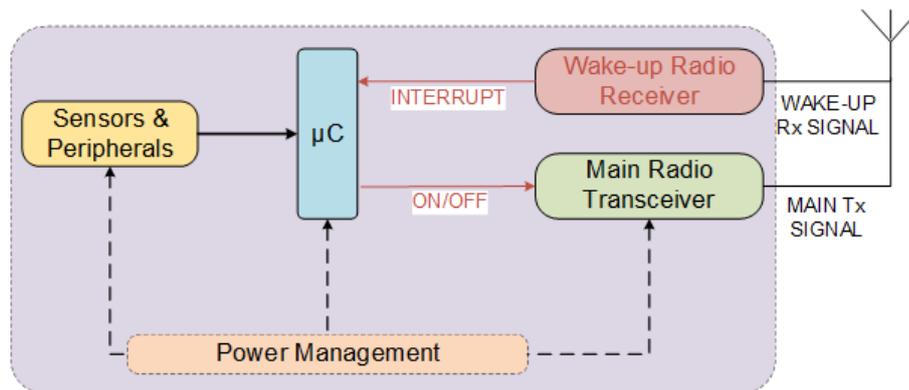
Este objetivo de baixo consumo, impõe restrições à modulação utilizada para o rádio despertador. Se a modulação utilizada possuir uma alta complexidade, como por exemplo a modulação por Chaveamento de Fase (PSK) e Chaveamento de Fase em Quadratura (QPSK), os circuitos de demodulação do WuRx serão mais complexos e exigirão mais potência (MAGNO et al., 2016). O esquema de modulação mais popular usado em implementações de rádio desperta é o *On-Off Keying* (OOK). Esta modulação simples permite uma simplificação drástica do circuito WuR e oferece oportunidades de redução de energia (JELICIC et al., 2012).

Em implementações de WuTx em uma rede LoRaWAN classe A, o *end-device* possui um rádio transmissor secundário para enviar o aviso de envio de pacote para o concentrador, enquanto ele possui receptor secundário. Quando o WuRx recebe um pacote de *Wake-Up* (WuP), ocorre o alerta do *gateway* sobre a transmissão e ele recebe as informações dos seus sensores.

Para a implementação do WuRx em uma rede LoRaWAN de classe A, o concentrador deve acordar o dispositivo final para que ele envie o pacote de dados. Dessa forma, o *end-node* permanece em baixo consumo até que seja detectado um sinal no canal de comunicação, enviado pelo *gateway*. Assim, o *end-device* sai de seu estado de baixo consumo e envia os dados de seus sensores para o concentrador por meio de pacotes de mensagens. Essa implementação proporciona a economia de energia do *end-node*, que normalmente fica acordado o tempo todo esperando a solicitação de envio dos pacotes de informações para o *gateway*.

A figura 22 apresenta um exemplo de arquitetura de hardware, onde um *end-node* possui um receptor de rádio secundário para o recebimento dos pacotes de *wake-up*.

Figura 22 – Exemplo de arquitetura de um *end-node* para *wake-up radio*



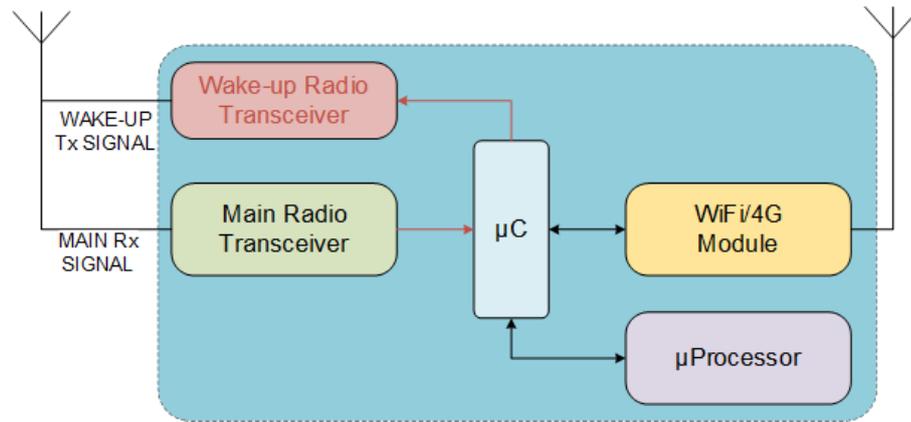
Fonte: Autoria própria

Já a figura 23 apresenta um *gateway* com rádio transmissor secundário, que realiza o envio dos pacotes de *wake-up* para o *end-node*.

## 1.7 MODULAÇÃO OOK

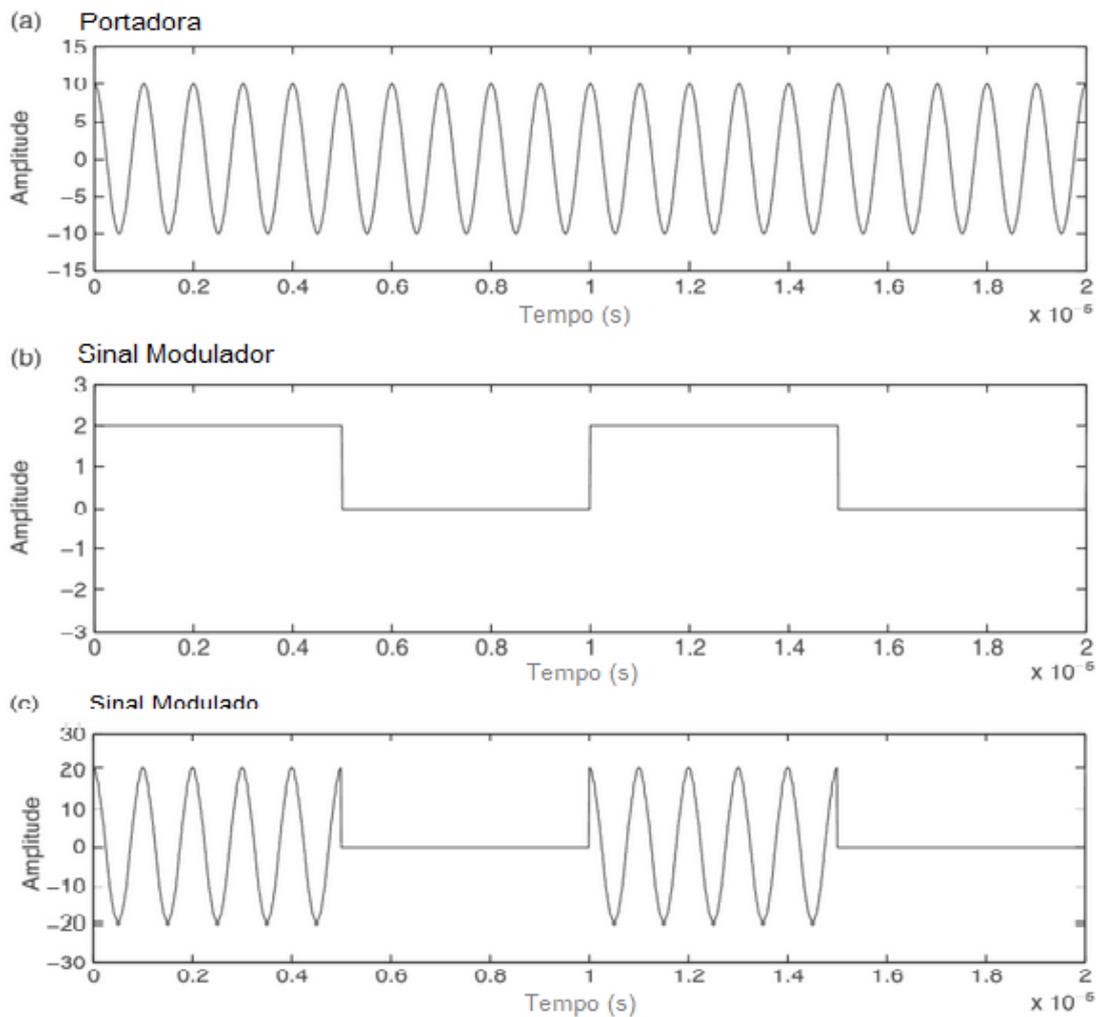
A modulação por chaveamento de amplitude ou ASK (*Amplitude Shift Keying*) consiste na alteração da amplitude da onda portadora em função do sinal a ser transmitido. Esta modulação tem como principais características a simplicidade no processo de modulação e demodulação, eficiência de largura de banda e baixa imunidade ao ruído. Devido às características citadas, esta modulação é indicada em ambientes de pouco ruído para interferir na recepção do sinal ou quando há requisitos de baixo custo para o sistema (SANTOS; LAMPA, 2018). A figura 24 ilustra o sinal da portadora, o sinal modulador e o sinal modulado para modulação ASK.

Figura 23 – Exemplo de arquitetura de um *gateway* para *wake-up radio*



Fonte: Autoria própria

Figura 24 – Sinal em Modulação ASK



Fonte: Adaptado de (ZHU; NANDI, 2015)

Para se gerar um sinal com modulação 2-ASK basta alterar o valor da amplitude do sinal de saída em função do bit a transmitir. Com base nesta condição, os símbolos na modulação binária M-ASK são definidos pela equação 5:

$$S_i(t) = \sqrt{\frac{2E_i(t)}{T_b}} \cos(2\pi f_c t + \phi_c), \quad i = 1, 2, \dots, M \quad (5)$$

Onde se tem:

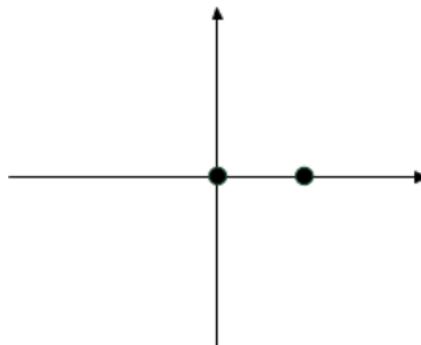
- $i \rightarrow$  Número de símbolos da constelação
- $E \rightarrow$  Energia por bit do sinal transmitido
- $T \rightarrow$  Duração do bit
- $f_c \rightarrow$  Frequência da portadora
- $\phi_c \rightarrow$  Fase da portadora

Um caso particular da modulação 2-ASK é a modulação OOK. Esta técnica transmite um pulso senoidal para representar a um determinado bit e nos intervalos de tempo do bit complementar não transmite pulsos. Neste caso particular, o bit 1 é codificado de forma similar à modulação ASK descrita na equação 5 e o bit 0 é codificado por um valor de amplitude zero com duração de tempo de um bit. Dessa forma, a expressão matemática que representa a modulação OOK é descrita pela equação 6.

$$S(t) = \begin{cases} \sqrt{\frac{2E_b}{T_b}} \cos(2\pi f_c t); & \text{bit 1} \\ 0; & \text{bit 0} \end{cases} \quad (6)$$

Assim, a constelação associada a modulação OOK é ilustrada conforme apresentado na figura 25.

Figura 25 – Sinal em Modulação ASK



Fonte:(FONTES, 2012)

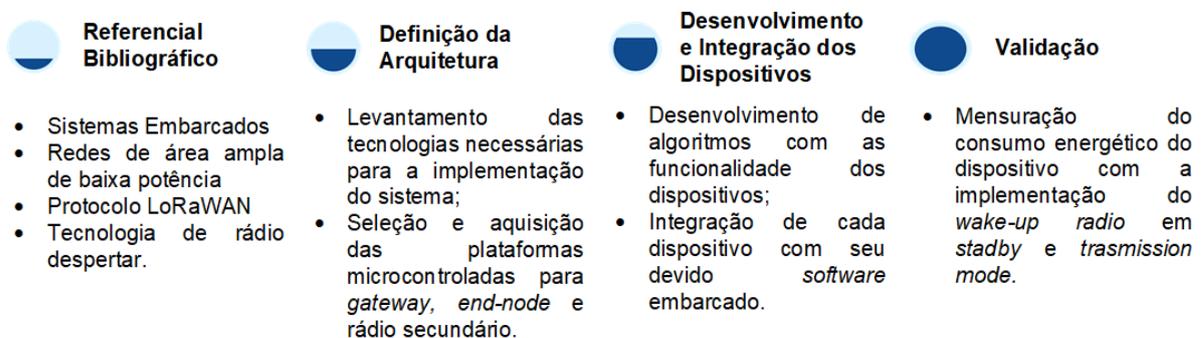
## 2 METODOLOGIA

O presente trabalho se trata de uma pesquisa aplicada, cujo objetivo é a realização de uma pesquisa exploratória e explicativa sobre o material bibliográfico. São utilizados os procedimentos técnicos de pesquisa bibliográfica e experimental. O método de abordagem utilizado será o hipotético-dedutivo e a elaboração segue o método de procedimento monográfico. A coleta de dados é feita através da observação direta intensiva e documentação indireta, sendo estes dados quantitativos.

O trabalho consiste no desenvolvimento de um sistema embarcado, baseado na tecnologia de *Wake-Up Radio*, como estratégia para comunicação orientada a eventos em uma rede LoRaWAN Classe A. O procedimento adotado para realização do projeto, se iniciou com um estudo bibliográfico relacionado a sistemas embarcados, IoT, redes de área ampla de baixa potência, protocolo LoRaWAN, tecnologia de rádio despertar e trabalhos associados ao tema.

Após a etapa de levantamento bibliográfico foi possível realizar a definição da arquitetura do sistema de forma a eleger as funcionalidades necessárias para os dispositivos, bem como delimitar quais plataformas embarcadas seriam possíveis utilizar para a implementação. Assim, a figura 26 apresenta de forma resumida as etapas das atividades realizadas para que o objetivo do projeto fosse alcançado.

Figura 26 – Diagrama de Desenvolvimento do Projeto

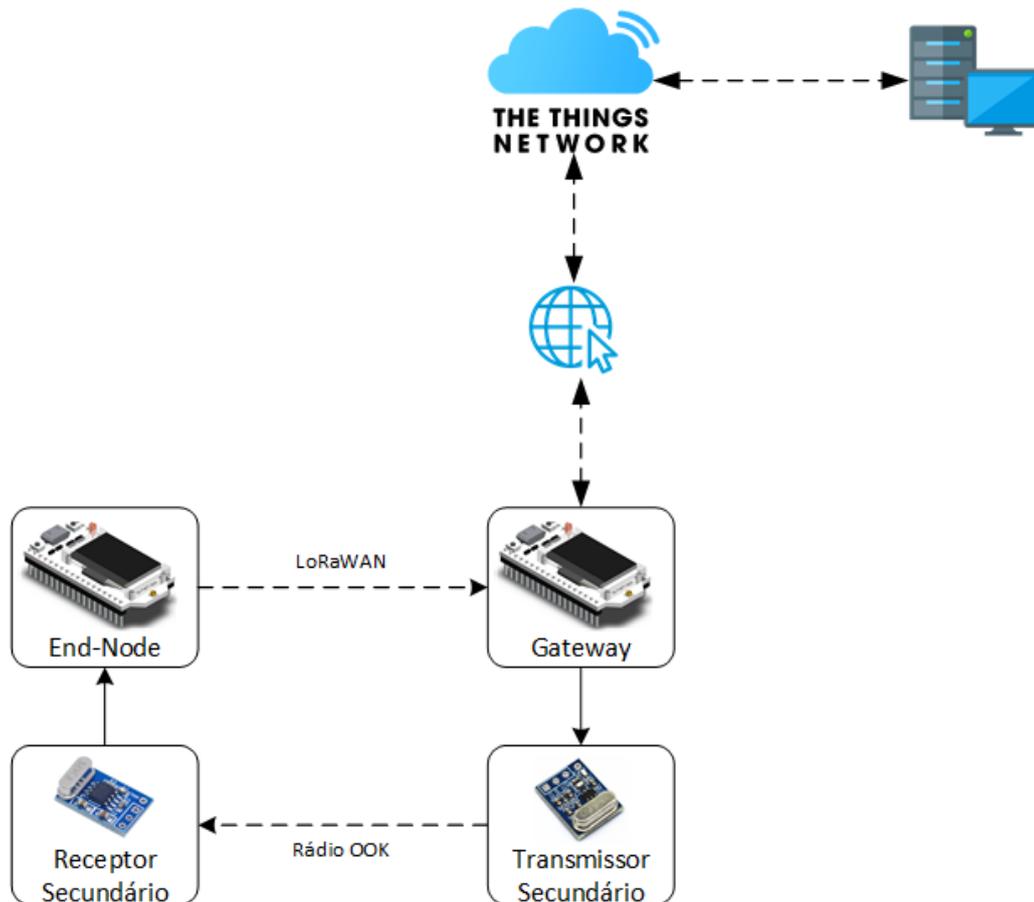


Fonte: Autoria própria

### 2.1 ARQUITETURA DO SISTEMA

Com o levantamento bibliográfico foi possível definir as matrizes tecnológicas utilizadas no processo de implementação de uma rede LoRaWAN, cuja topologia é constituída por *end-nodes*, *gateway* e servidor de rede. Uma vez especificado os elementos necessários para a composição de uma rede LoRaWAN, definiu-se os dispositivos de rádio necessários para a realização do rádio despertar. Assim, a figura 27 apresenta a arquitetura geral do sistema desenvolvido.

Figura 27 – Arquitetura Geral do Sistema



Fonte: Autoria própria

## 2.2 MÉTODO PARA DEFINIÇÃO DO *HARDWARE*

Após o levantamento das necessidades tecnológicas, pôde-se concluir que seria necessária a aquisição de um módulo que interface de comunicação Wi-Fi e LoRa. Em seguida, foi iniciada a etapa de pesquisa de módulos de comunicação para o uso como rádio secundário. As opções foram avaliadas e selecionadas com base no custo e tempo estimado de entrega.

## 2.3 MÉTODO PARA DESENVOLVIMENTO DO *FIRMWARE*

Com a aquisição dos módulos necessários e a definição de como seriam gerados e interpretados os eventos, foi iniciado o processo de definição das principais funções para o desenvolvimento dos algoritmos a serem embarcados nos mesmos. Dessa forma, inicialmente foram desenvolvidos os algoritmos para comunicação via LoRaWAN do *gateway* e o *end-node*. Durante a fase de implementação, foi necessária a realização de etapas de testes preliminares de forma a garantir que a comunicação LoRa dos módulos ocorreria como o esperado. Em seguida, foram desenvolvidos em códigos separados algoritmos para

que um módulo realizassem o envio de um pacote de dados via rádio utilizando modulação OOK para outro dispositivo. Posteriormente, foi realizada a integração das funções para o uso de um único software embarcado em cada microcontrolador.

### 2.3.1 Método para Geração de Eventos

O MQTT foi utilizado para que a aplicação possa realizar iterações com o *gateway* de forma a gerar os eventos para o rádio despertar, optou-se pelo uso do protocolo de comunicação MQTT (*Message Queuing Telemetry Transport*) com transmissão de dados via Wi-Fi a 2,4GHz de frequência.

Assim, o sistema realiza o envio de uma mensagem ao *gateway*, de forma que o mesmo envie um pacote de *wake-up* para o *end-node* por meio de um rádio secundário. O node, ao reconhecer este pacote, deve enviar para o *gateway* os dados de um sensor coletados naquele instante, por meio de LoRaWAN. O *end-node* ficará no aguardo de um novo pacote de *wake-up*, que será enviado automaticamente pelo sistema ou de forma manual.

## 2.4 VALIDAÇÃO DO SISTEMA

Para a validação do projeto, foi mensurado o consumo energético do *end-node* utilizando um osciloscópio. Essas medições foram realizadas com o dispositivo em três estados, sendo eles: aguardando o recebimento de um pacote de *wake-up* (*Standby mode*), durante o recebimento de uma mensagem por meio de seu rádio secundário (*Receive Mode*) e durante a transmissão de dados por meio do rádio principal (*Transmission Mode*).

### 3 IMPLEMENTAÇÃO

O presente capítulo descreve as etapas utilizadas para o desenvolvimento do sistema proposto.

#### 3.1 DEFINIÇÃO DO HARDWARE DO GATEWAY E END-NODE

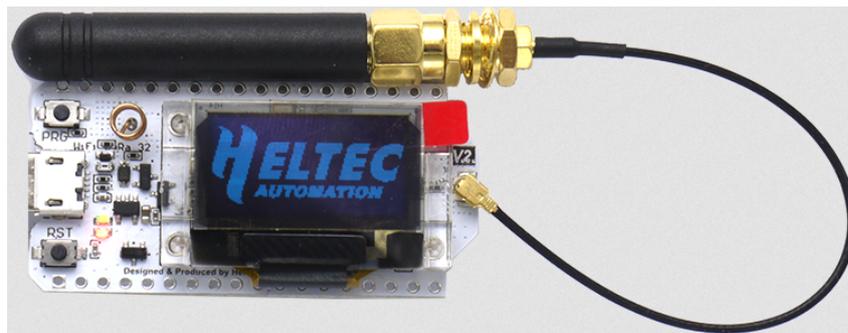
Após o levantamento das necessidades tecnológicas, pôde-se concluir que seria necessária a aquisição de um módulo que interface de comunicação Wi-Fi e LoRa para ser utilizado com *gateway* para a rede LoRaWAN implementada. Para o *end-node* optou-se pelo uso de um dispositivo similar ao *gateway*, pois o mesmo precisaria, também, possuir comunicação LoRa.

Em seguida, foi iniciada a etapa de pesquisa de módulos de comunicação para o uso como rádio secundário. As opções foram avaliadas e selecionadas com base no custo e tempo estimado de entrega. Vale ressaltar que a principal diferença entre o *hardware* dos dispositivos está no rádio secundário e suas conexões físicas a placa, pois o concentrador possui o rádio transmissor OOK, enquanto o *end-device* possui o rádio receptor.

##### 3.1.1 Módulos Heltec Wi-Fi LoRa (V2)

O módulo Heltec Wi-Fi LoRa (V2) é uma de uma placa desenvolvida pela empresa Heltec Automation para aplicações IoT, contendo um chip ESP32 que se trata de um microcontrolador desenvolvido pela Espressif que possui baixo custo e baixo consumo de energia em um chip (SoC) da série com recursos de Wi-Fi e Bluetooth de modo duplo (ESPRESSIF, 2022). Além do microcontrolador da Espressif, o módulo possui um transceptor SX1276, para comunicação LoRa, e display OLED de 0.96 polegadas. Dessa forma, foram adquiridos módulos Heltec Wi-Fi LoRa (V2), pois o mesmo possui todas as características necessárias para a realização da implementação do projeto. A figura 28 apresenta o *hardware* do módulo selecionado.

Figura 28 – Módulo Heltec Wi-Fi Lora V2



Fonte: (HELTEC, 2018)

Este dispositivo é capaz de realizar transmissões sem fio nas faixas de frequências 433, 868 e 915 MHz, com um baixo consumo de energia, longo alcance e grande capacidade anti-interferência (SEMTECH, 2020). As principais características do módulo são apresentadas na tabela 2.

Tabela 2 – Especificações técnicas do Heltec Wi-Fi LoRa V2

| Parâmetros          | Referências                           |
|---------------------|---------------------------------------|
| CPU                 | Xtensa Dual-core 32-bit LX6 600 DMIPS |
| Wi-Fi               | 802.11 b/g/n                          |
| Bluetooth           | v4.2 BR/EDR e BLE                     |
| LoRa                | LoRaWAN ou Node-to-Node               |
| SRAM                | 520 KB                                |
| Nº de GPIOs         | 22 pinos                              |
| Nº de ADC (12 Bits) | 18 canais                             |
| Nº de DAC (8 Bits)  | 2 canais                              |

Fonte: Adaptado (HELTEC, 2018)

Para o uso do dispositivo para comunicação via LoRaWAN, é utilizado o transceptor SX1276, presente na placa, que proporciona um longo alcance comunicação e alta imunidade a interferências enquanto minimiza o consumo de corrente. Suas características são apresentadas na tabela 3.

Tabela 3 – Especificações técnicas do SX1276

| Parâmetros                                 | Referências    |
|--|----------------|
| Faixa de Frequência                        | 137 - 1020 MHz |
| Fator de Espalhamento                      | 6 - 12         |
| Largura de Banda                           | 7.8 - 500 kHz  |
| Tensão de Alimentação                      | 1.8 - 3.7V     |
| Consumo de Corrente em <i>Sleep Mode</i>   | 0.2 $\mu$ A    |
| Consumo de Corrente em <i>Standby Mode</i> | 1.6mA          |

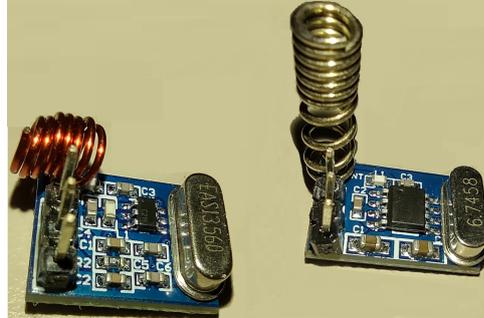
Fonte: Adaptado (SEMTECH, 2020)

### 3.1.2 Transmissor de Rádio Secundário

Para possibilitar transmissão de pacotes de *wake-up* para o *end-node*, foi selecionado o módulo SYN115 e para a recepção de dados foi escolhido o módulo SYN480R. Ambos são rádios de alto desempenho e de fácil uso para aplicações sem fio que opera com modulação

ASK/OOK cujo fabricante é a empresa Synoxo. A figura 29 apresenta os módulo de rádio SYN115 e SYN 480R, respectivamente.

Figura 29 – Módulo de Rádio SYN115 e SYN 480R



Fonte: Autoria própria

Para possibilitar transmissão de pacotes de *wake-up* para o *end-node*, o *gateway* possui o módulo SYN115. Este dispositivo é um transmissor de rádio de alto desempenho e de fácil uso para aplicações sem fio que opera com modulação ASK/OOK (*Amplitude Shift Keying/On-Off Keyed*), cuja faixa de frequência vai de 300 a 450 MHz, baixa (SYNOXO, 2010a). A tabela 4 apresenta o esquema de conexões entre o módulo rádio transmissor e o módulo utilizado com *gateway*.

Tabela 4 – Conexões entre o módulo SYN115 e o módulo Heltec Wi-Fi LoRa V2

| Módulo SYN115 | Módulo Heltec |
|---------------|---------------|
| VCC           | 5V            |
| GND           | GND           |
| DAT           | 23            |

Fonte: Autoria própria

### 3.1.3 Receptor de Rádio Secundário

O *end-node* possui o módulo SYN480R, para realizar a recepção de pacotes destinados a realização do radio despertar. Este dispositivo consiste em um de rádio receptor que opera com modulação ASK/OOK, cuja frequência de operação vai de 300 a 450 MHz, baixa (SYNOXO, 2010b). A tabela 5 apresenta o esquema de conexões entre o módulo rádio receptor e o módulo utilizado com *end-node*.

Tabela 5 – Conexões entre o módulo SYN480R e o módulo Heltec Wi-Fi LoRa V2

| Módulo SYN480R | Módulo Heltec |
|----------------|---------------|
| VCC            | 5V            |
| GND            | GND           |
| DAT            | 38            |

Fonte: Autoria própria

## 3.2 DESENVOLVIMENTO DO FIRMWARE DOS MÓDULOS

Os *firmwares* de cada Heltec ESP32 LoRa foi desenvolvido nas linguagens C e C++ por meio do ambiente integrado de desenvolvimento (IDE) Arduino, com o auxílio da biblioteca da Heltec. Este software possui plataforma *open source*, é disponibilizado para sistemas operacionais Windows, MAC e Linux, além de ser um ambiente de programação amigável e flexível para elaboração de algoritmos para embarcados (FEZARI; DAHOUD, 2018). Todos os algoritmos desenvolvidos para a implementação do *software* embarcado dos dispositivos estão disponíveis para consulta em um repositório *online* (AZEVEDO, 2022).

A tabela 6 apresenta as bibliotecas importadas para o desenvolvimento utilizadas para a implementação do *software* dos dispositivos.

Tabela 6 – Bibliotecas utilizadas para a implementação do *firmware* do *gateway* e *end-node*

| Biblioteca   | Descrição   |
|--------------|---|
| ArduinoJson  | Biblioteca em C++ utilizada pra organizar dados em formato JSON   |
| gBase64      | Biblioteca de codificação e decodificação de base64   |
| Heltec_ESP32 | Biblioteca que fornece suporte para módulos Heltec Wi-Fi Lora 32  |
| PubSubClient | Biblioteca que permite utilizar o ESP32 como cliente no protocolo MQTT                                  |
| Radio Head   | Biblioteca biblioteca para o envio e recepção de pacote de mensagens através de uma variedade de rádios |
| TTN_ESP32    | Biblioteca que fornece suporte para implementação de LoRaWAN com módulo ESP32                           |
| Wi-Fi        | Biblioteca que possibilita o uso da interface de comunicação Wi-Fi do ESP32                             |

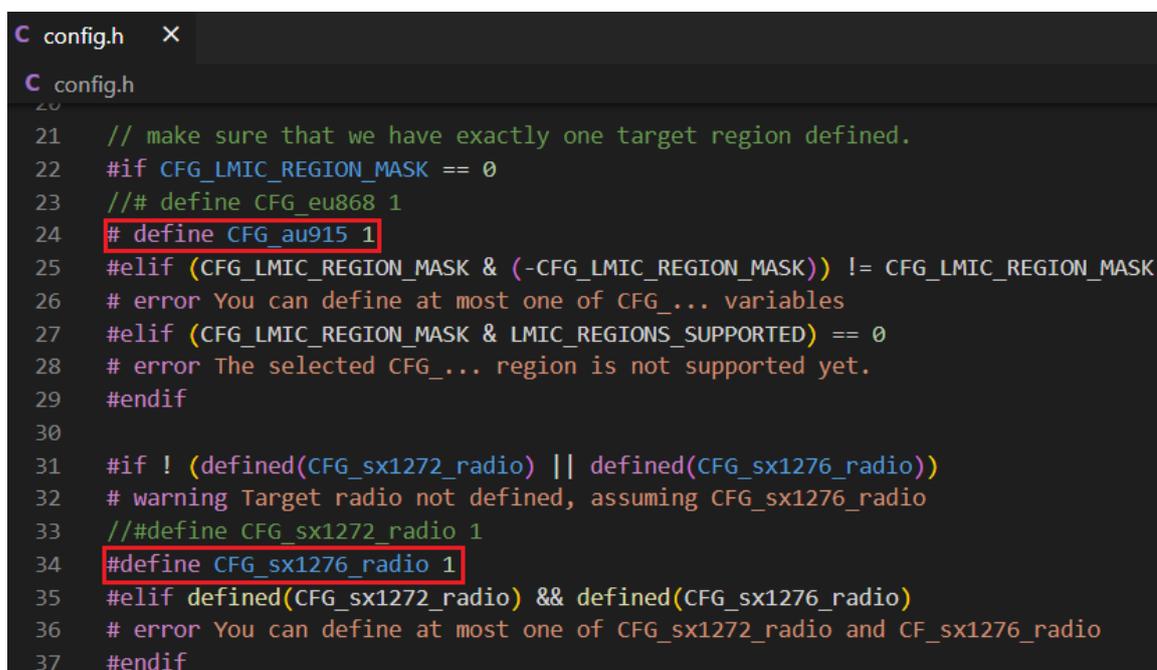
Fonte: Autoria própria

### 3.2.1 Firmware do End-Node

A biblioteca LMIC (*LoraWAN MAC in C*), fornecida pela IBM por meio da biblioteca TTN\_ESP32, permite o uso de módulos Semtech SX1272 e SX1276, além de possuir suporte para os planos regionais EU868, US915, AU915, AS923, KR e IN. Esta biblioteca possui exemplos de programas que implementam o funcionamento de um *end-node* por meio dos métodos de ativação OTAA ou ABP. Dessa forma, o *software* embarcado para o *end-device* foi desenvolvido utilizando como base o exemplo de código TTN-ABP.ino, implementando uma rede LoRaWAN com ativação ABP.

Por se tratar de uma biblioteca com suporte a diversos planos de frequências, é recomendado pelo fabricante a realizar a definição da região e rádio a serem utilizados. Essa definição é realizada no arquivo `config.h` presente no diretório da biblioteca instalada e conforme apresentado na figura 30.

Figura 30 – Definição na Biblioteca LMIC da Região e Rádio Utilizados



```

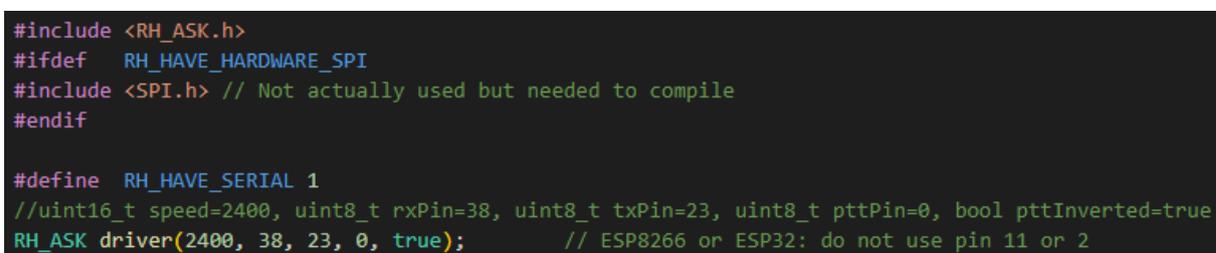
C config.h x
C config.h
20
21 // make sure that we have exactly one target region defined.
22 #if CFG_LMIC_REGION_MASK == 0
23 //# define CFG_eu868 1
24 # define CFG_au915 1
25 #elif (CFG_LMIC_REGION_MASK & (-CFG_LMIC_REGION_MASK)) != CFG_LMIC_REGION_MASK
26 # error You can define at most one of CFG_... variables
27 #elif (CFG_LMIC_REGION_MASK & LMIC_REGIONS_SUPPORTED) == 0
28 # error The selected CFG_... region is not supported yet.
29 #endif
30
31 #if ! (defined(CFG_sx1272_radio) || defined(CFG_sx1276_radio))
32 # warning Target radio not defined, assuming CFG_sx1276_radio
33 //#define CFG_sx1272_radio 1
34 #define CFG_sx1276_radio 1
35 #elif defined(CFG_sx1272_radio) && defined(CFG_sx1276_radio)
36 # error You can define at most one of CFG_sx1272_radio and CF_sx1276_radio
37 #endif

```

Fonte: Autoria própria

Outra biblioteca utilizada foi a RadioHead, que é orientada a objetos e proporciona a recepção e envio de pacotes de mensagens através de uma variedade de rádios, o que possibilitou a comunicação via OOK entre os rádios secundários. A figura 31 apresenta a importação da biblioteca no código principal do *end-node* e a configuração das GPIOs do dispositivo.

Figura 31 – Função para Habilitar Comunicação via OOK



```

#include <RH_ASK.h>
#ifdef RH_HAVE_HARDWARE_SPI
#include <SPI.h> // Not actually used but needed to compile
#endif

#define RH_HAVE_SERIAL 1
//uint16_t speed=2400, uint8_t rxPin=38, uint8_t txPin=23, uint8_t pttPin=0, bool pttInverted=true
RH_ASK driver(2400, 38, 23, 0, true); // ESP8266 or ESP32: do not use pin 11 or 2

```

Fonte: Autoria própria

### 3.2.2 Firmware do Gateway

Inicialmente foram realizadas pesquisas quanto ao desenvolvimento de um *gateway* de apenas um canal, por se tratar de uma implementação de baixo custo. Dessa forma, os

algoritmos que proporcionam o funcionamento do dispositivo foram baseados no ESP-1ch-Gateway, disponível em <https://github.com/things4u/ESP-1ch-Gateway>.

Para proporcionar o funcionamento do dispositivo conforme as especificações necessárias, foram realizada a configuração nos arquivos configGway.h, onde o plano de frequência utilizado como AU915, o rádio utilizado foi definido como o SX1276, o *pinout* adequado para o módulo Heltec ESP32 LoRa (V2), seguindo como base as infamações do arquivo loraModem.h, e a endereço do servidor de rede. A figura 32 apresenta a configuração alguns dos parâmetros citados.

Figura 32 – Definição dos Parâmetros do Gateway

```
C configGway.h
78 // Define the frequency band the gateway will listen on.
79 // You can find the definitions in "loraModem.h" and frequencies in
80 // See https://www.thethingsnetwork.org/docs/lorawan/frequency-plans.html
81 #define AU925_928 1
82
83 // Define the correct radio type that you are using
84 #define CFG_sx1276_radio
85 // #define CFG_sx1272_radio
86
87 // We support a few pin-out configurations out-of-the-box: HALLARD, COMPRESULT and
88 // Heltec/TTGO ESP32.
89 // If you use one of these, just set the parameter to the right value.
90 // If your pin definitions are different, update the loraModem.h file to reflect the
91 // hardware settings.
92 // 1: HALLARD
93 // 2: COMRESULT pin out
94 // 3: ESP32, Wemos pin out (Not used)
95 // 4: ESP32, Heltec and TTGO pin out (should work for Heltec, 433 and Oled too).
96 // 5: Other, define your own in loraModem.h (does not include GPS Code)
97 #if !defined _PIN_OUT
98 #   define _PIN_OUT 5
99 #endif
100
101 // The spreading factor is the most important parameter to set for a single channel
102 // gateway. It specifies the speed/datarate in which the gateway and node communicate.
103 // As the name says, in principle the single channel gateway listens to one channel/frequency
104 // and to one spreading factor only.
105 // This parameters contains the default value of SF, the actual version can be set with
106 // the webserver and it will be stored in SPIFF
107 // NOTE: The frequency is set in the loraModem.h file and is default 868100000 Hz.
108 #if !defined _SPREADING
109 #   define _SPREADING SF9
110 #endif
111
112 // Define the CLASS mode of the gateway
113 // A: Baseline Class
114 // B: Beacon/Battery Class
115 // C: Continuous Listen Class
116 #define _CLASS "A"
117
118 // TTN Version 3:
119 #   define _TTNSERVER "au1.cloud.thethings.network"
120 #   define _TTNPORT 1700 // Standard port for TTN
121 #endif
```

Fonte: Autoria própria

Foram realizadas, também, alterações no arquivo configNode.h, onde foram inseridas as informações para possibilitar que o dispositivo seja capaz de se conectar a rede Wi-Fi disponível, como mostra a figura 33 .

Figura 33 – Definição da Rede Wi-Fi Utilizada

```

C configNode.h
84
85 // Wifi definitions
86 // WPA is an array with SSID and password records. Set WPA size to number of entries in array
87 // When using the WiFiManager, we will overwrite the first entry with the
88 // accesspoint we last connected to with wifiManager
89 struct wpas {
90     char login[32]; // Maximum Buffer Size (and allocated memory)
91     char passw[64];
92 };
93
94
95 // Please fill in at least ONE valid SSID and password from your own WiFi network
96 // below. This is needed to get the gateway working
97 //
98 wpas wpa[] = {
99     {"", ""}
100 //{"", ""}
101 };

```

Fonte: Autoria própria

No código principal do dispositivo, ESP-sc-gway.ino, foi adicionada a biblioteca RadioHead, para o uso do rádio secundário para possibilitar o envio dos pacotes de *wake-up*, e, também, a biblioteca PubSubClient, para habilitar o recebimento e envio de mensagens via protocolo MQTT.

Em seguida foi implementada a comunicação com um *broker* MQTT, para realizar a geração de evento de forma que o *gateway* envie um pacote de *wake-up* para o *end-node*. Logo, as informações de endereço do *broker* e uma função de *callback* para realizar a leitura e processamento das mensagens recebidas pelos tópicos definidos foram adicionadas ao código principal do dispositivo, como mostra a figura 34.

Figura 34 – Função para Processamento de Mensagens MQTT

```

ESP-sc-gway.ino
250 const char* mqttServer = "broker.hivemq.com";
251 const int mqttPort = 1883;
252
253
254 WiFiClient espClient;
255 PubSubClient client(espClient);
256
257
258 // =====
259 // MAIN PROGRAM CODE (MQTT)
260 // =====
261
262 void callback(char* topico, byte* recado, unsigned int tamanho)
263 {
264     const char *msg = "data";
265
266     if (recado[0] == '1')
267     {
268         driver.send((uint8_t *)msg, strlen(msg));
269         driver.waitPacketSent();
270         Serial.println("Message sent");
271         client.publish("heltec/tcc_pub", "Message sent");
272     }
273     else
274     {
275         Serial.println("It wan't possible to sent the message, please verify the command");
276         client.publish("heltec/tcc_pub", "It wan't possible to sent the message, please verify the command");
277     }
278 }

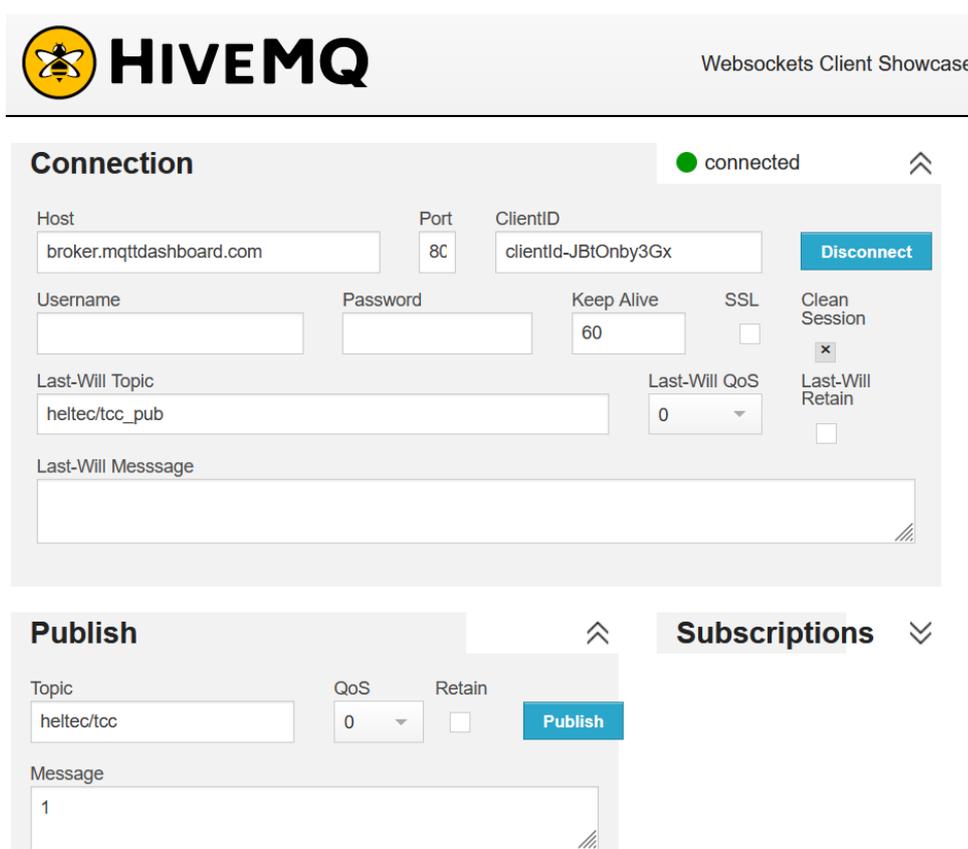
```

Fonte: Autoria própria

### 3.3 BROKER MQTT

Para o envio e recebimento de mensagens via protocolo MQTT, foi utilizado um broker MQTT online de código aberto chamado HiveMQ que pode ser acessado por meio do link <http://www.hivemq.com/demos/websocket-client/>. Trata-se de uma plataforma MQTT em nuvem gratuita e gerenciada pela HiveMQ Cloud. A figura 35 apresenta a tela de exibição e envio das mensagens via MQTT.

Figura 35 – Dashboard do Broker MQTT Utilizado



The screenshot displays the HiveMQ Websockets Client Showcase interface. At the top left is the HiveMQ logo, and at the top right is the text 'Websockets Client Showcase'. The main content is organized into three panels:

- Connection:** Shows a green 'connected' status. Fields include Host (broker.mqtdashboard.com), Port (80), and ClientID (clientId-JBtOnby3Gx). There is a 'Disconnect' button. Other fields include Username, Password, Keep Alive (60), SSL (unchecked), Clean Session (checked), Last-Will Topic (heltec/tcc\_pub), Last-Will QoS (0), and Last-Will Retain (unchecked). A text area for Last-Will Message is also present.
- Publish:** Shows a Topic (heltec/tcc), QoS (0), and Retain (unchecked). There is a 'Publish' button. A text area for Message contains the number '1'.
- Subscriptions:** This section is currently empty and has a downward arrow icon.

Fonte: Autoria própria

### 3.4 CADASTRO DOS DISPOSITIVOS NO SERVIDOR DE REDE

Neste tópico são apresentados os passos utilizados para a criação da aplicação, registro do *end-device* e *gateway* no servidor de rede *The Things Network*. Para enviar os dados do *end-node* para o servidor de rede utilizado, foi necessária a criação de uma aplicação e realizar o procedimento de registro dos dispositivos no console. Dessa forma, foi criada uma conta no site da *The Things Network*, por ser o *network server* utilizado para o desenvolvimento do projeto. A figura 36 apresenta a tela inicial do site com destaque para a localização da opção de *login*.

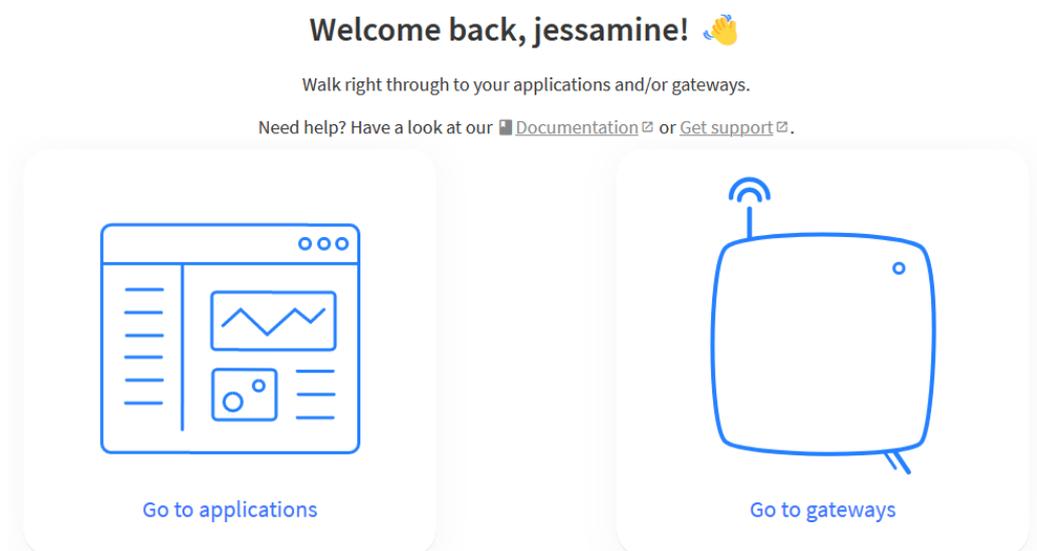
Figura 36 – Tela de Inicial do Console



Fonte: Autoria própria

Após a realização do *login*, é apresentada uma tela inicial, como mostra a figura 37, onde o usuário pode escolher entre acessar aplicações ou o *gateway* registrados no servidor de rede em nuvem.

Figura 37 – Tela de Inicial do Console

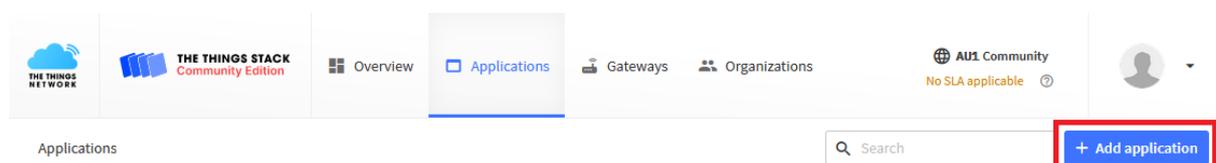


Fonte: Autoria própria

### 3.4.1 Registrando uma Aplicação no TTN

Ao selecionar a opção *Go to applications* é exibida uma tela onde se tem a opção de consultar as aplicações já registradas ou realizar a adição de uma nova aplicação por meio da opção *Add applications*, como mostra a figura 38.

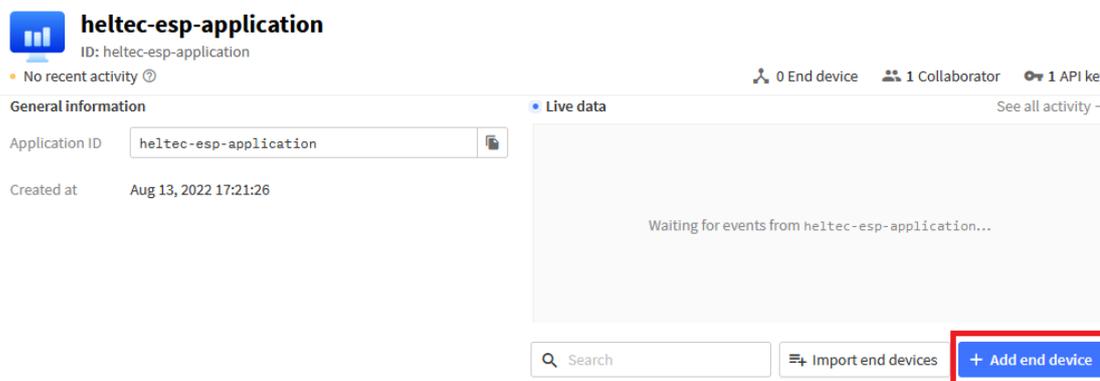
Figura 38 – Tela para Adição de uma Nova Aplicação



Fonte: Autoria própria

Para adicionar uma aplicação foi necessário o preenchimento dos campos de *Application ID*, *Application Name* e *Description*. Ao finalizar a criação, é exibida uma página onde se pode ter uma visão geral desta aplicação e, também, realizar o cadastro dos dispositivos finais desta aplicação, como mostra a figura 39.

Figura 39 – Tela de Visão Geral da Aplicação

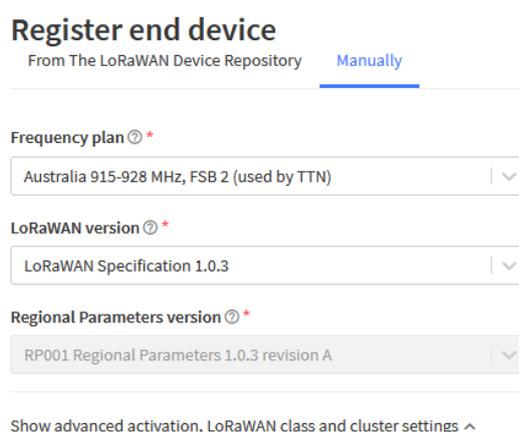


Fonte: Autoria própria

### 3.4.2 Registrando o End-Node no TTN

Para o cadastros do *end-device* foi selecionada a opção *Manually*, onde são exibidas as opções de planos de frequências e versão do LoRaWAN. Dessa forma, para o campo *Frequency Plan*, foi selecionada a opção Australia 915-928MHz, FSB 2 (used by TTN), pois a mesma se encaixa na configuração do padrão AU915 e o *LoRaWAN version* foi selecionado o *LoRaWAN Specification 1.0.3*, como apresentado na figura 40.

Figura 40 – Tela de Inicial do Cadastro de um End-Node



Fonte: Autoria própria

Ao selecionar a opção de exibir configurações avançadas é aberto um novo menu onde se pode selecionar os modos de ativação e a classe da rede LoRaWAN do dispositivo. Como modo de ativação foi selecionado o ABP, dessa forma são abertos campos para a geração automática de *DevEUI*, *Device address*, *APPSKey*, *NwkSKey*, como mostra a figura 41.

Figura 41 – Configurações Avançadas do Cadastro de um Novo End-Node

**Activation mode** ⓘ \*

Over the air activation (OTAA)  
 Activation by personalization (ABP)  
 Define multicast group (ABP & Multicast)

**Additional LoRaWAN class capabilities** ⓘ

None (class A only) | v

**DevEUI** ⓘ

.. .. . Generate 10/50 used

**Device address** ⓘ \*

.. .. .

**AppSKey** ⓘ \*

.. .. . Generate

**NwkSKey** ⓘ \*

.. .. . Generate

**End device ID** ⓘ \*

my-new-device

This value is automatically prefilled using the DevEUI

Register end device

Fonte: Autoria própria

Assim, ao finalizar o cadastro é possível visualizar as informações gerais do dispositivo no console TTN, como mostra a figura 42.

Figura 42 – Tela de Visão Geral do End-Node

**General information**

End device ID eui-70 [redacted] 78

Frequency plan Australia 915-928 MHz, FSB 2 (used by T...

LoRaWAN version LoRaWAN Specification 1.0.3

Regional Parameters version TS001 Technical Specification 1.0.3

**Activation information**

AppEUI n/a

DevEUI [redacted] <>

**Session information**

Device address 26 [redacted] 85 <>

NwkSKey .. .. .

SNwKSIntKey .. .. .

NwkSEncKey .. .. .

AppSKey .. .. .

Fonte: Autoria própria

Para que as informações transmitidas pelo *end-node* sejam exibidas no console TTN é necessário adicionar ao *firmware* do dispositivo as informações da aplicação e da sessão. Assim, é necessário capturar os dados *Device address*, *APPSKey*, *NwkSKey*, apresentados

tela de visão geral do *end-node*, para realizar o preenchimento dos campos no *firmware* do dispositivo, como ilustrado na figura 43.

Figura 43 – Registros das Chaves de Ativação no End-Node

```

/*****
 * Go to your TTN console register a device then the copy fields
 * and replace the devAddr, nwkSKey and appSKey as strings
 *****/
const char* devAddr = "26[redacted]5"; // Device Address
const char* nwkSKey = "65[redacted]1D"; // Network Session Key
const char* appSKey = "26[redacted]AE"; // Application Session Key

```

Fonte: Autoria própria

### 3.4.3 Registrando o Gateway no TTN

Voltando a tela inicial do console, ao selecionar a opção *Go to gateways* é exibida uma tela onde se tem a opção de consultar os concentradores já registradas ou realizar a adição de um por meio da opção *Register gateway*, como mostra a figura 44.

Figura 44 – Tela para Adição de um Novo Gateway



Fonte: Autoria própria

Para adicionar um *gateway* foi necessário o preenchimento dos campos, *Gateway EUI Gateway Name* e *Frequency Plan*. O campo *Gateway ID* é automaticamente preenchido ao inserir o *Gateway EUI*, que corresponde ao endereço MAC dispositivo. Para o *Frequency Plan*, mais uma vez, foi utilizado o Australia 915-928MHz, FSB 2 (used by TTN), pois, como observado anteriormente, o mesmo se encaixa na configuração do padrão AU915. A figura 45 apresenta a tela de cadastro do dispositivo.

Figura 45 – Tela de Inicial do Cadastro de um Gateway

 A screenshot of the 'Register gateway' form. The form has a title 'Register gateway' and a subtitle 'Gateway EUI'. Below the subtitle, there is a text input field for 'Gateway EUI' with a 'Reset' button to its right. Below that, there is a text input field for 'Gateway ID' with a red asterisk indicating it is required. Below that, there is a text input field for 'Gateway name' with the placeholder text 'My new gateway'. Below that, there is a dropdown menu for 'Frequency plan' with the selected option 'Australia 915-928 MHz, FSB 2 (used by TTN)'. At the bottom of the form, there is a blue button with the text 'Register gateway'.

Fonte: Autoria própria

A informação de *Gateway EUI* pode ser obtida ao embarcar o *software* no dispositivo e aguardar a sua inicialização, como mostra a figura 46.

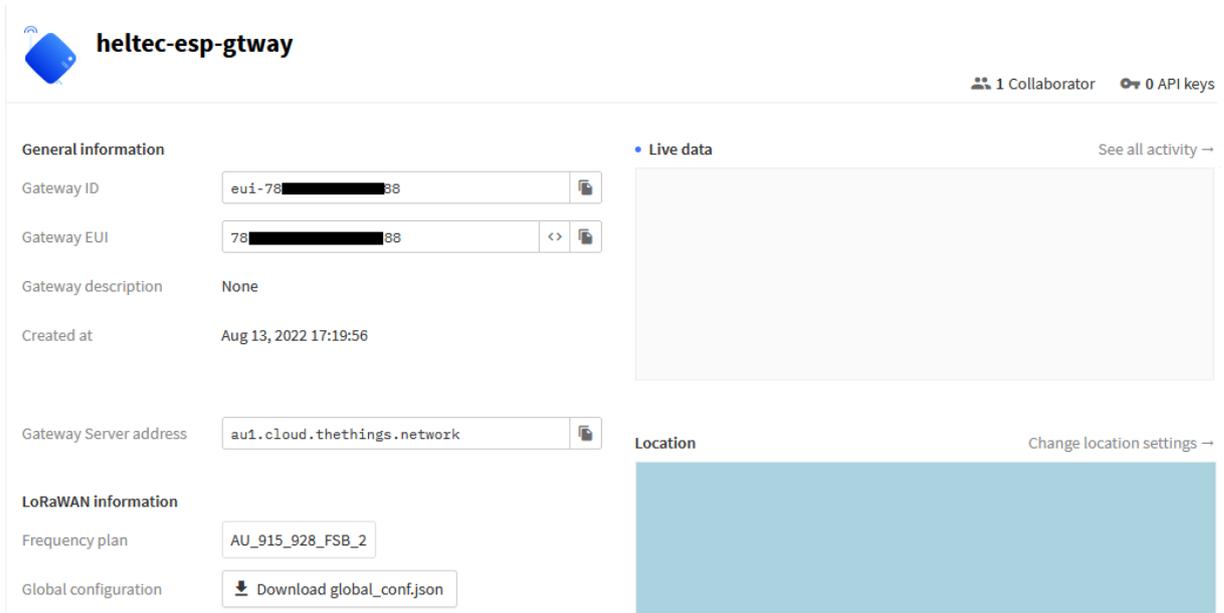
Figura 46 – Registros Exibidos da Porta Serial do Gateway ao Inicializar

```
0:0:1. WlanConnect SSID=██████████
WlanStatus:: CONNECTED ssid=██████████
Host=esp32██████████ WiFi Connected to ██████████ on IP=██████████
Local UDP port=1700
UDP Connection successful
Gateway ID: ██████████, Listening at SF9 on 916800000 Hz.
resolveHost:: OK=nl.pool.ntp.org IP=██████████
resolveHost:: OK=aul.cloud.thethings.network IP=██████████
WWW Server started on port 80
RegInvertIQ :: 27
RegInvertIQ2:: 1D
--- Setup() ended, Starting loop() ---
Conecting to MQTT...
Conected to MQTT
```

Fonte: Autoria própria

Após o fim do preenchimento dos campos é exibida uma página onde se pode ter uma visão geral deste dispositivo, como mostra a figura 47.

Figura 47 – Tela de Visão Geral do Gateway



The screenshot displays the configuration page for a Heltec ESP-Gateway. The interface includes a header with the device name 'heltec-esp-gtway', a collaborator count of 1, and 0 API keys. The main content is organized into several sections:

- General information:** Contains fields for Gateway ID (eui-78██████████38), Gateway EUI (78██████████88), Gateway description (None), and Created at (Aug 13, 2022 17:19:56).
- Gateway Server address:** Set to 'aul.cloud.thethings.network'.
- LoRaWAN information:** Includes Frequency plan (AU\_915\_928\_FSB\_2) and a button to download 'global\_conf.json'.
- Live data:** A section for real-time data, currently empty, with a 'See all activity' link.
- Location:** A section for location settings, currently empty, with a 'Change location settings' link.

Fonte: Autoria própria

### 3.5 MENSURAÇÃO DO CONSUMO ENERGÉTICO DO DISPOSITIVO

Para possibilitar a aferição do consumo energético do *end-node* seria necessário utilizar um dispositivo capaz de medir a corrente consumida pelo dispositivo ao longo de um tempo específico. Dessa forma, foi utilizado um osciloscópio Tektronix MDO4104-3

para realizar o consumo de corrente para se estimar o consumo energético do dispositivo como um todo. No osciloscópio, foi utilizada uma ponta de prova específica, a tektronix TCP0020, que trata-se de um sensor de corrente de alta precisão. A figura 48 ilustra o equipamento para a mensuração do consumo energético do dispositivo, junto a ponta de prova utilizada.

Figura 48 – Osciloscópio e Ponta de Prova Utilizada para Aferições

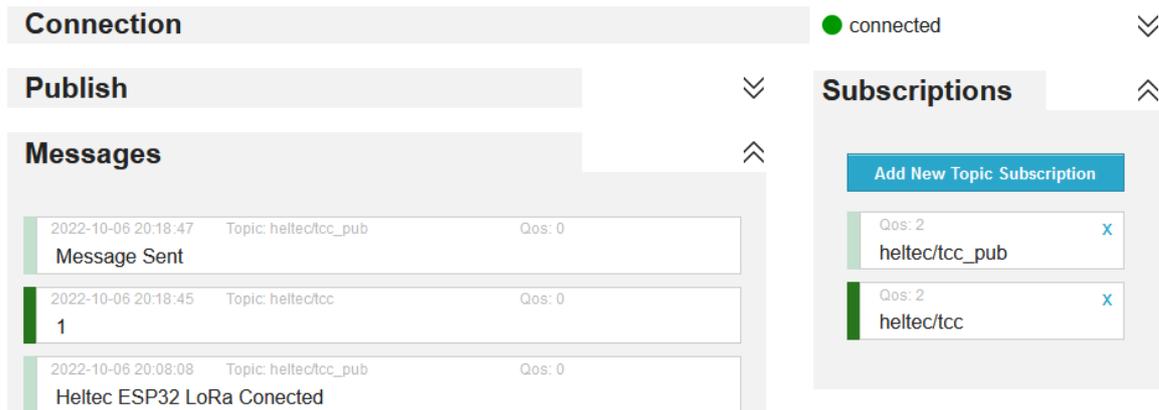


Fonte: (TEKTRONIX, 2021)

## 4 RESULTADOS E DISCUSSÕES

O processo de envio de mensagens via MQTT de forma manual pode ser realizada pelo *dashboard* presente no HiveMQ. Assim, ao realizar o envio de uma mensagem por meio do MQTT, o *gateway* verifica se a mensagem corresponde ao esperado para o envio do *Wake-up Package* e responde com o status do envio do pacote, como apresentado na figura 49.

Figura 49 – Troca de mensagens via MQTT



Fonte: Autoria própria

Dessa forma, ao receber uma mensagem específica, o *gateway* realiza o envio de um pacote de *wake-up* para o node por meio do rádio secundário e recebe o pacote via LoRaWAN. A figura 50 apresenta o log do dispositivo durante esse processo.

Figura 50 – Gateway inicializado e recebendo pacotes via MQTT e LoRaWAN

```
0:0:1. WlanConnect SSID=[REDACTED]
WlanStatus:: CONNECTED ssid=[REDACTED]
Host=esp32[REDACTED] WiFi Connected to [REDACTED] on IP=[REDACTED]
Local UDP port=1700
UDP Connection successful
Gateway ID: [REDACTED], Listening at SF9 on 916800000 Hz.
resolveHost:: OK=nl.pool.ntp.org IP=[REDACTED]
resolveHost:: OK=aul.cloud.thethings.network IP=[REDACTED]
WWW Server started on port 80
RegInvertiq :: 27
RegInvertiq2:: 1D
--- Setup() ended, Starting loop() ---

Connecting to MQTT...
Conected to MQTT
Message sent
v _STRICT_1CH==1 done, freq=916800000, HEX=54 165 66 0
v txLoraModem hi:: micr=55.444965, tmst=55.545209, wait=100244,
v txLoraModem lo:: , size=25, old=[ 60 85 fe 0d 26 8d 01 00 06
v OK, stateMachine TXDONE: rcvd=55.744270, done= 0.0
^ TX_ACK:: readUDP: protocol version 2+
```

Fonte: Autoria própria

Após a recepção de um pacote de *wake-up*, o *end-node* realiza o envio de uma mensagens via LoRaWAN, como apresentado na figura 51.

Figura 51 – End-node realizando o envio mensagem via LoRaWAN

```

Starting
-----Status-----
Device EUI: 0000000000000000
Application EUI: 0000000000000000
netid: 13
devaddr: 2[REDACTED]85
NwkSKey: 65[REDACTED]1D
AppSKey: 26[REDACTED]AE
data rate: 5
tx power: 30dB
freq: 916800000Hz
-----
Temp: 18.200001

```

Fonte: Autoria própria

Durante esse processo de comunicação, também são exibidas informações no *network server*. A figura 52 apresenta as informações exibidas no log da aplicação, enquanto a 53 apresenta o log do gateway no console do *The Things Network*.

Figura 52 – Log da Aplicação no console TTN

| Time     | Type                           | Details  |
|----------|--------------------------------|--|
| 20:18:47 | Schedule data downlink for...  | DevAddr: 26[REDACTED]85 <> MAC payload: C3 12 00 63 DD CA 30 97 ... <> Rx1 Delay: 1              |
| 20:18:46 | Forward uplink data message    | DevAddr: 26[REDACTED]85 <> Payload: 01 67 00 E6 <> FPort: 1 Data rate: SF7BW125 SNR: 9 RSSI: -42 |
| 20:18:46 | Successfully processed data... | DevAddr: 26[REDACTED]85 <>   |

Fonte: Autoria própria

Figura 53 – Log do Gateway no console TTN

| Time     | Type                   | Data preview  |
|----------|------------------------|---|
| 20:18:53 | Receive gateway status | Metrics: { ackr: 0, txin: 0, txok: 0, rxin: 30, rxok: 30, rxfw: 1 }               |
| 20:18:47 | Send downlink message  | Tx Power: 30 Data rate: SF7BW500  |
| 20:18:46 | Receive uplink message | DevAddr: 26[REDACTED]85 <> FCnt: 24 FPort: 1 Data rate: SF7BW125 SNR: 9 RSSI: -42 |

Fonte: Autoria própria

Para a validação do processo de comunicação e verificação da perda de pacotes, foi utilizado um *script* em linguagem python para realizar o envio da mensagem que realiza a geração dos eventos de comunicação via MQTT em um intervalo de 10 minutos, durante um período de 4 horas. A tabela 7, apresenta a relação das mensagens enviadas e recebidas pelos dispositivos nesse intervalo de tempo.

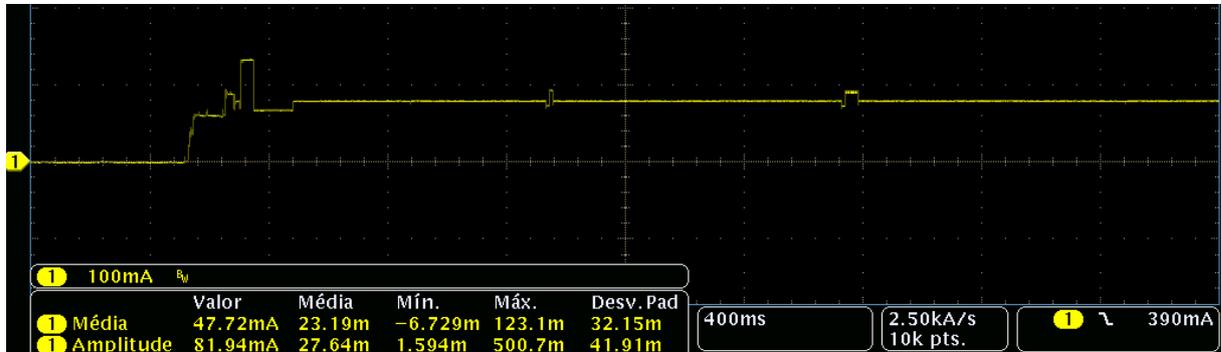
Tabela 7 – Comunicação

| Tipo de Comunicação | Mensagens Enviadas | Mensagens Recebidas |
|---------------------|--------------------|---------------------|
| via MQTT            | 24                 | 21                  |
| via OOK             | 21                 | 21                  |
| via LoRaWAN         | 21                 | 21                  |

Fonte: Autoria própria

Por meio da figura 54 é possível visualizar o consumo de corrente do *end-node* em *standby*, mensurado com o auxílio de um osciloscópio.

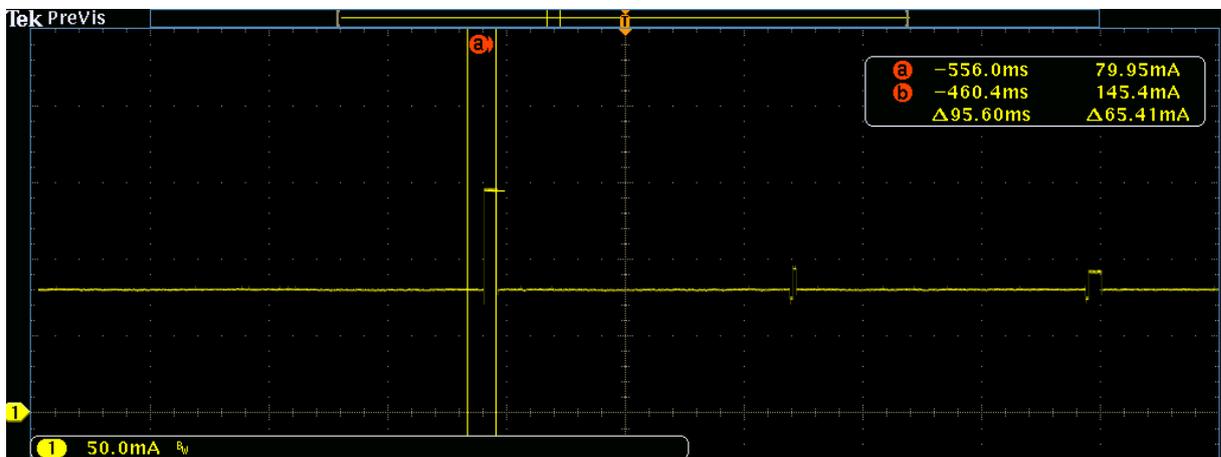
Figura 54 – Medição em Standby



Fonte: Autoria própria

O SYN480R realiza de forma contínua leituras de rádio até que a mensagem recebida corresponda ao pacote de *wake-up*. Com as medições, foi possível notar que quando dispositivo entra em modo de transmissão, ocorrem três picos de corrente sendo o primeiro cerca de 145,4mA, como podem ser observados por meio da figura 55, e os dois seguintes tem valores próximos as 90mA.

Figura 55 – Medição do primeiro pico da transmissão LoRa



Fonte: Autoria própria

Por meio da tabela 7, pode ser notado que durante esse processo de comunicação ocorreu uma baixa taxa de perda de pacotes dados enviados que podem ser ocasionadas por fatores como a perda de conexão a internet, pois as perdas apresentadas se concentram na comunicação entre o *broker* MQTT e o *gateway*.

Por meio do gráfico apresentadona figura 54, pôde-se notar que o consumo do dispositivo se mantém estável após a inicialização, isso simplifica a estimativa do consumo energético do dispositivo ao longo do tempo e possibilita o dimensionamento mais preciso de uma bateria para o dispositivo.

## CONCLUSÕES

Neste trabalho foi realizada a prova de conceito de uma estratégia de *Wake-Up Radio* para comunicação orientada a eventos em uma rede LoRaWAN Classe A. Para isto, foram feitas revisões bibliográficas sobre conceitos básicos de Internet das Coisas e sobre as redes LPWAN, além de um estudo detalhado sobre a modulação LoRa e o protocolo LoRaWAN.

A partir da análise dos dados obtidos nos testes, foi observado que o objetivo do desenvolvimento de uma comunicação iniciada por meio de um evento foi concluído com sucesso, pois ao realizar o envio de uma mensagem via MQTT para o *gateway*, o mesmo realiza o envio de um pacote via rádio com modulação OOK para o *end-node* que, após interpretar o pacote recebido, envia pacotes de informações para o *gateway* por meio do protocolo LoRaWAN. Todo esse processo pode ser visto tanto pela serial dos dispositivos quanto pelo *network server*.

Além disso, conforme definido nos objetivos específicos, foram realizados estudos sobre estratégias de rádio despertar no contexto de redes LoRaWAN e a análise do consumo energético do *end-node*, após a implementação do *Wake-Up Radio*.

Uma implementação futura para o sistema seria o uso do *node-red* para o desenvolvimento de uma API onde as mensagens MQTT e as informações do *network server* possam ser visualizadas em apenas uma tela. Esta ferramenta é utilizada para desenvolvimento baseada em fluxo para programação visual desenvolvida originalmente pela IBM.

Utilizando as implementações de *hardware* e *firmware* aqui desenvolvidas, é notório que existe possibilidade da simplificação dos circuitos de forma a se obter uma maior redução consumo energético dos dispositivos, por meio do uso de chips com maior eficiência energética que realizam as mesmas funções e, também, uma redução de custos, para o caso escalabilidade.

## REFERÊNCIAS

- ALBINI, L. C. P. *Redes de Computadores I*. 2015. Disponível em: <[https://www.inf.ufpr.br/albini/apostila/Apostila\\_Redex1\\_Beta.pdf](https://www.inf.ufpr.br/albini/apostila/Apostila_Redex1_Beta.pdf)>.
- ALLIANCE, L. *What is LoRaWAN?* 2015. Disponível em: <<https://lora-alliance.org/wp-content/uploads/2020/11/what-is-lorawan.pdf>>.
- ATMOKO, R.; Riantini, R.; HASIN, M. Iot real time data acquisition using mqtt protocol. In: IOP PUBLISHING. *Journal of Physics: Conference Series*. [S.l.], 2017. v. 853, n. 1, p. 12003.
- AZEVEDO, J. M. de L. *GitHub*. 2022. Disponível em: <<https://github.com/jessaminelima/TCC>>.
- BOUGUERA, T. et al. Energy consumption model for sensor nodes based on lora and lorawan. *Sensors*, MDPI, v. 18, n. 7, p. 2104, 2018.
- CAVALLI, D. T. et al. Análise de técnicas de machine learning aplicadas para geolocalização lora. Universidade de Passo Fundo, 2021.
- CENTENARO, M. et al. Long-range communications in unlicensed bands: The rising stars in the iot and smart city scenarios. *IEEE Wireless Communications*, IEEE, v. 23, n. 5, p. 60–67, 2016.
- CHASE, O.; ALMEIDA, F. Sistemas embarcados. *Mídia Eletrônica. Página na internet*: <[www.sbajovem.org/chase](http://www.sbajovem.org/chase)>, capturado em, v. 10, n. 11, p. 13, 2007.
- CHAUDHARI, B. S.; ZENNARO, M.; BORKAR, S. Lpwan technologies: Emerging application characteristics, requirements, and design considerations. *Future Internet*, Multidisciplinary Digital Publishing Institute, v. 12, n. 3, p. 46, 2020.
- CHEN, S. et al. A vision of iot: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, IEEE, v. 1, n. 4, p. 349–359, 2014.
- COMMITTEE, L. *LoRaWAN Backend Interfaces 1.0 Specification*. 2017. Disponível em: <<https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm-backend-interfaces-v1.0.pdf>>.
- COMMITTEE, L. *LoRaWAN Regional Parameters v1.1*. 2017. Disponível em: <<https://lora-alliance.org/wp-content/uploads/2020/11/lorawan-regional-parameters-v1.1ra.pdf>>.
- COMMITTEE, L. *LoRaWAN Specification v1.1*. 2017. Disponível em: <[https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm\\_specification\\_v1.1.pdf](https://lora-alliance.org/wp-content/uploads/2020/11/lorawantm_specification_v1.1.pdf)>.
- CUNHA, M. P. da. *Análise e Caracterização da Cobertura LoRa para Ambiente Indoor na Frequência de 915 MHz*. 2021. Disponível em: <<http://www.ppgeet.uff.br/site/wp-content/uploads/2021/08/Michel-Pessoa.pdf>>.
- DINCULEANĂ, D.; CHENG, X. Vulnerabilities and limitations of mqtt protocol used between iot devices. *Applied Sciences*, MDPI, v. 9, n. 5, p. 848, 2019.
- DJIDI, N. E. H. et al. How can wake-up radio reduce lora downlink latency for energy harvesting sensor nodes? *Sensors*, MDPI, v. 21, n. 3, p. 733, 2021.

- ESPRESSIF. *ESP32 Technical Reference Manual v. 4.7*. 2022. Disponível em: <[https://www.espressif.com/sites/default/files/documentation/esp32\\_technical\\_reference\\_manual\\_en.pdf](https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf)>.
- FAN, C.-I. et al. A multiple end-devices authentication scheme for lorawan. *Electronics*, MDPI, v. 11, n. 5, p. 797, 2022.
- FEZARI, M.; DAHOUD, A. A. Integrated development environment “ide” for arduino. *WSN applications*, ResearchGate, p. 1–12, 2018.
- FONTES, A. I. R. *Classificação Automática de Modulação Digital com uso de Correntropia para Ambientes de Rádio Cognitivo*. Dissertação (Mestrado) — Universidade Federal do Rio Grande do Norte, 2012.
- FOROUZAN, B. *Comunicação de Dados e Redes de Computadores*. McGraw Hill Brasil, 2009. ISBN 9788563308474. Disponível em: <<https://books.google.com.br/books?id=FIaDr9ZtwXgC>>.
- GEMALTO, A.; SEMTECH. *A WHITE PAPER PREPARED FOR THE LoRa ALLIANCE*. 2017. Disponível em: <[https://lora-alliance.org/wp-content/uploads/2020/11/lorawan\\_security\\_whitepaper.pdf](https://lora-alliance.org/wp-content/uploads/2020/11/lorawan_security_whitepaper.pdf)>.
- GERMANI, L. et al. An iot architecture for continuous livestock monitoring using lora lpwan. *Electronics*, Multidisciplinary Digital Publishing Institute, v. 8, n. 12, p. 1435, 2019.
- HAN, J.; WANG, J. An enhanced key management scheme for lorawan. *Cryptography*, MDPI, v. 2, n. 4, p. 34, 2018.
- HELTEC. *HELTEC WiFi LoRa 32 V2 Product Information*. 2018. Disponível em: <<https://heltec.org/project/wifi-lora-32/>>.
- HUANG, C.-F.; CHEN, W.-Y. A wideband microstrip array based on air substrate for wireless base-station applications. In: IEEE. *2013 7th European Conference on Antennas and Propagation (EuCAP)*. [S.l.], 2013. p. 1268–1271.
- JELICIC, V. et al. Analytic comparison of wake-up receivers for wsns and benefits over the wake-on radio scheme. In: *Proceedings of the 7th ACM workshop on Performance monitoring and measurement of heterogeneous wireless and wired networks*. [S.l.: s.n.], 2012. p. 99–106.
- KIM, D.-H.; LEE, E.-K.; KIM, J. Experiencing lora network establishment on a smart energy campus testbed. *Sustainability*, MDPI, v. 11, n. 7, p. 1917, 2019.
- KIM, S.; LEE, H.; JEON, S. An adaptive spreading factor selection scheme for a single channel lora modem. *Sensors*, MDPI, v. 20, n. 4, p. 1008, 2020.
- LAVRIC, A.; POPA, V. Internet of things and lora™ low-power wide-area networks: a survey. In: IEEE. *2017 International Symposium on Signals, Circuits and Systems (ISSCS)*. [S.l.], 2017. p. 1–5.
- LOPEZ-AGUILERA, E. et al. Ieee 802.11-enabled wake-up radio: Use cases and applications. *Sensors*, MDPI, v. 20, n. 1, p. 66, 2019.

- MACEDO, R. T. et al. *Redes de computadores*. 1st. ed. [S.l.]: Santa Catarina: Universidade Federal de Santa Maria, 2018. ISBN 9788583412250.
- MAGNO, M. et al. Design, implementation, and performance evaluation of a flexible low-latency nanowatt wake-up radio receiver. *IEEE Transactions on Industrial Informatics*, IEEE, v. 12, n. 2, p. 633–644, 2016.
- MEHIC, M. et al. Lorawan end nodes: Security and energy efficiency analysis. *Alexandria Engineering Journal*, Elsevier, v. 61, n. 11, p. 8997–9009, 2022.
- OLIVEIRA, S. de. *Internet das coisas com ESP8266, Arduino e Raspberry PI*. [S.l.]: Novatec Editora, 2017.
- PASTÓRIO, A. et al. Fundamentos de lorawan—teoria e prática. *Sociedade Brasileira de Computação*, 2021.
- PEREIRA, L. A. M. et al. Software embarcado, o crescimento e as novas tendências deste mercado. *Revista de Ciências Exatas e Tecnologia*, v. 6, n. 6, p. 85–94, 2011.
- PIYARE, R. et al. On-demand lora: Asynchronous tdma for energy efficient and low latency communication in iot. *Sensors*, MDPI, v. 18, n. 11, p. 3718, 2018.
- RATASUK, R.; MANGALVEDHE, N.; GHOSH, A. Overview of lte enhancements for cellular iot. In: IEEE. *2015 IEEE 26th annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*. [S.l.], 2015. p. 2293–2297.
- ROSSATO, J.; SPANHOL, F. A.; CAMARGO, E. T. de. Implantação e avaliação de uma rede sem-fio de longo alcance e baixa potência para cidades inteligentes. In: SBC. *Anais do IV Workshop de Computação Urbana*. [S.l.], 2020. p. 192–205.
- SANTOS, J. I. d.; LAMPA, P. H. D. M. *Desenvolvimento de um gerador de ondas arbitrárias utilizando a técnica de síntese digital direta e com recursos de modulação*. Dissertação (B.S. thesis) — Universidade Tecnológica Federal do Paraná, 2018.
- SANTOS, N. S. G. d.; CUNHA, M. W. R. d. Análise de redes lorawan: desempenho indoor, outdoor, híbrido, aplicações e simulações utilizando ns3. 2019.
- SELLER, O. Lorawan security. *Journal of ICT Standardization*, p. 47–60, 2021.
- SEMTECH. *SX1276/77/78/79 - 137 MHz to 1020 MHz Low Power Long Range Transceiver*. 2020. Disponível em: <<https://www.semtech.com/products/wireless-rf/lora-core/sx1276#documentation>>.
- SONI, D.; MAKWANA, A. A survey on mqtt: a protocol of internet of things (iot). In: *International conference on telecommunication, power analysis and computing techniques (ICTPACT-2017)*. [S.l.: s.n.], 2017. v. 20, p. 173–177.
- STALLINGS, W. *Arquitetura e Organização de Computadores*. 10th. ed. [S.l.]: São Paulo: Pearson Education do Brasil, 2017. ISBN 9788543020532.
- SYNOXO. *SYN113/SYN115 Datasheet (300-450MHz ASK Transmitter) Version 1.0*. 2010. Disponível em: <<https://www.msseletronica.com.br/arquivos/1381.pdf>>.
- SYNOXO. *SYN470R Datasheet (300-450MHz ASK Receiver) Version 1.3*. 2010. Disponível em: <[https://datasheet.lcsc.com/szlcsc/SYN480R-FS24\\_C15561.pdf](https://datasheet.lcsc.com/szlcsc/SYN480R-FS24_C15561.pdf)>.

TANENBAUM, A. S.; WETHERALL, D. *Rede de Computadores*. 5th. ed. [S.l.]: Pearson Education do Brasil, 2011. ISBN 9788576059240.

TEKTRONIX. *Mixed Domain Oscilloscopes MDO4000C Series Datasheet*. 2021. Disponível em: <<https://download.tek.com/datasheet/MDO4000C-Datasheet-EN-US-48W-60277-10.pdf>>.

TORRES, N.; PINTO, P.; LOPES, S. I. Security vulnerabilities in lpwans—an attack vector analysis for the iot ecosystem. *Applied Sciences*, MDPI, v. 11, n. 7, p. 3176, 2021.

ZHU, Z.; NANDI, A. K. *Automatic modulation classification: principles, algorithms and applications*. [S.l.]: John Wiley & Sons, 2015.