

Monitoramento de ônibus circulares

Uma abordagem utilizando LoRaWAN e GPS

Julio Nunes Avelar

19 de Outubro de 2023

Resumo

Este texto descreve a implementação de um dispositivo de monitoramento de ônibus circulares, adotando a tecnologia LoRaWAN e o sistema GPS. O objetivo principal deste dispositivo é rastrear a posição dos ônibus em tempo real e transmitir esses dados. Além disso, busca-se substituir os dispositivos em uso na UNICAMP, que atualmente são baseados em GSM e GPS. A implementação desse tipo de dispositivo tem um impacto significativo no cotidiano dos usuários de transporte circular, uma vez que auxilia na tomada de decisões e otimiza o tempo de espera.

1 Introdução

Em um cenário em que as cidades se expandem cada vez mais, com um trânsito cada vez mais congestionado e as pessoas necessitando percorrer distâncias cada vez maiores, o transporte público assume um papel essencial na vida cotidiana. Sem ele, tarefas como ir para o trabalho, frequentar a faculdade e muitas outras se tornam extremamente complexas e dispendiosas. Dada a significativa importância do transporte público, monitorá-lo torna-se crucial. Ter acesso a informações como a localização em tempo real e o tempo estimado para chegar a um destino específico auxilia as pessoas a tomar decisões cruciais em seu dia a dia, como a escolha do ônibus certo, o horário de partida e o ponto de embarque.

Para desempenhar essa função de monitoramento, tecnologias como GPS e GSM são amplamente empregadas. O GPS fornece informações vitais, incluindo posicionamento

e velocidade, enquanto o GSM é responsável por transmitir esses dados. Embora essa abordagem tenha se mostrado eficaz ao longo dos anos, ela não está isenta de desvantagens, principalmente no que se refere à transmissão de informações via GSM. Esse método apresenta problemas, como a dependência significativa de redes de terceiros e alto consumo de energia.

Além das questões já mencionadas, com o advento de redes de última geração, como o 5G, e a ampla adoção do 4G em todo o mundo, o antigo padrão 2G, no qual o GSM se baseia, está gradualmente sendo desativado, com a previsão de desligamento completo em 2024, conforme o cronograma global.

Para superar essas limitações e substituir o uso do GSM nesses dispositivos, surgem abordagens que envolvem a adoção de padrões mais recentes, como o 4G e o 5G, e o uso de tecnologias de longo alcance alternativas, como o SigFox e o LoRaWAN. Este último se destaca entre as outras tecnologias devido à sua natureza open source, baixo custo de implementação, baixo consumo de energia e alcance estendido. Por essas razões, este projeto escolheu essa tecnologia para a implementação de um dispositivo de monitoramento de ônibus circular. Denominado "CircuLoRa", esse dispositivo faz parte do projeto Smart Campus da Prefeitura universitária da UNICAMP.

2 Objetivo

Além do objetivo citado anteriormente de substituir o padrão GSM, tal dispositivo tem como objetivo principal, substituir os dispositivos em operação atualmente nos circulares internos da UNICAMP, caso o mesmo se demonstre capaz.

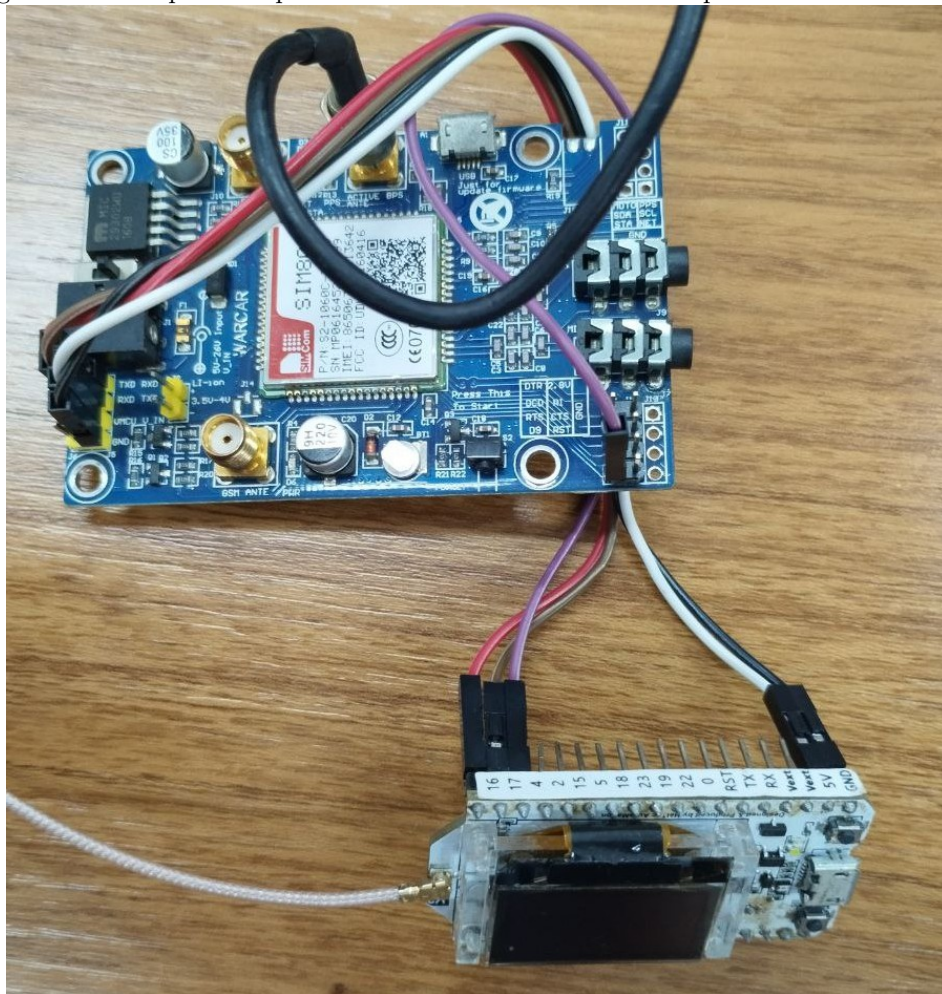
3 Tecnologias e dispositivos utilizados.

Além da tecnologia LoRaWAN, que desempenha um papel essencial na transmissão de dados, também está sendo feito o uso da tecnologia GPS, como mencionado anteriormente. Quanto ao hardware, optou-se por implementar o módulo LoRa SX1276. Como microcontrolador, foi escolhido o ESP32, com o software sendo capaz de ser utilizado tanto com ESP32s2 quanto com ESP32s3 e ESP32c3. Por fim, em relação ao módulo GPS, atualmente está sendo utilizado o SIM808, visando o reaproveitamento

dos módulos antigos com GSM. No entanto, está sendo considerada a possibilidade de utilizar módulos como o NEO-6m, NEO-7m e NEO-8m em futuras atualizações. Além disso, planeja-se a expansão do projeto para suportar microcontroladores mais acessíveis e menos potentes, como o Raspberry Pi Pico e STM32.

Para o desenvolvimento do software, foi utilizado o Espressif IoT Development Framework (ESPIDF), que oferece uma base sólida para nossa implementação. Além disso, foi utilizada uma adaptação da biblioteca LMIC para o ESPIDF, garantindo uma integração eficaz com o ESP32.

Imagem 1: Protótipo do dispositivo feito com módulo SIM808 e placa Heltec ESP32 LoRa

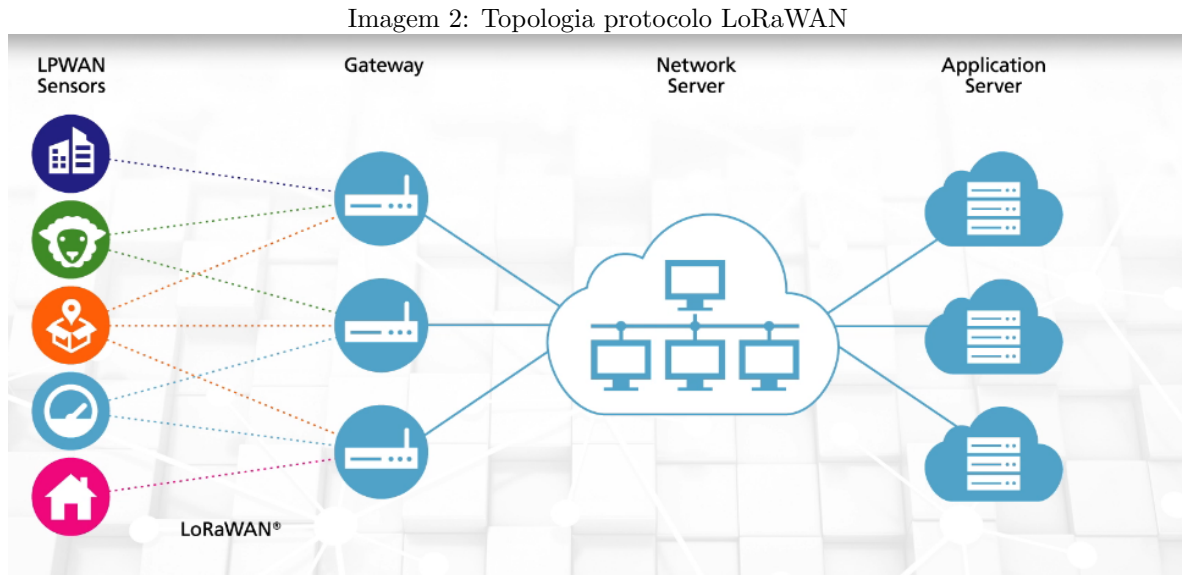


Fonte: Dos autores

4 Implementação

4.1 Arquitetura do sistema

O sistema é composto por quatro camadas essenciais que abrangem todo o conjunto. Essas camadas incluem o dispositivo, o gateway LoRaWAN, o servidor em nuvem e a aplicação final. A imagem abaixo oferece um resumo das camadas do sistema.



Fonte: Grupo de Teleinformática e Automação UFRJ

4.1.1 Dispositivo

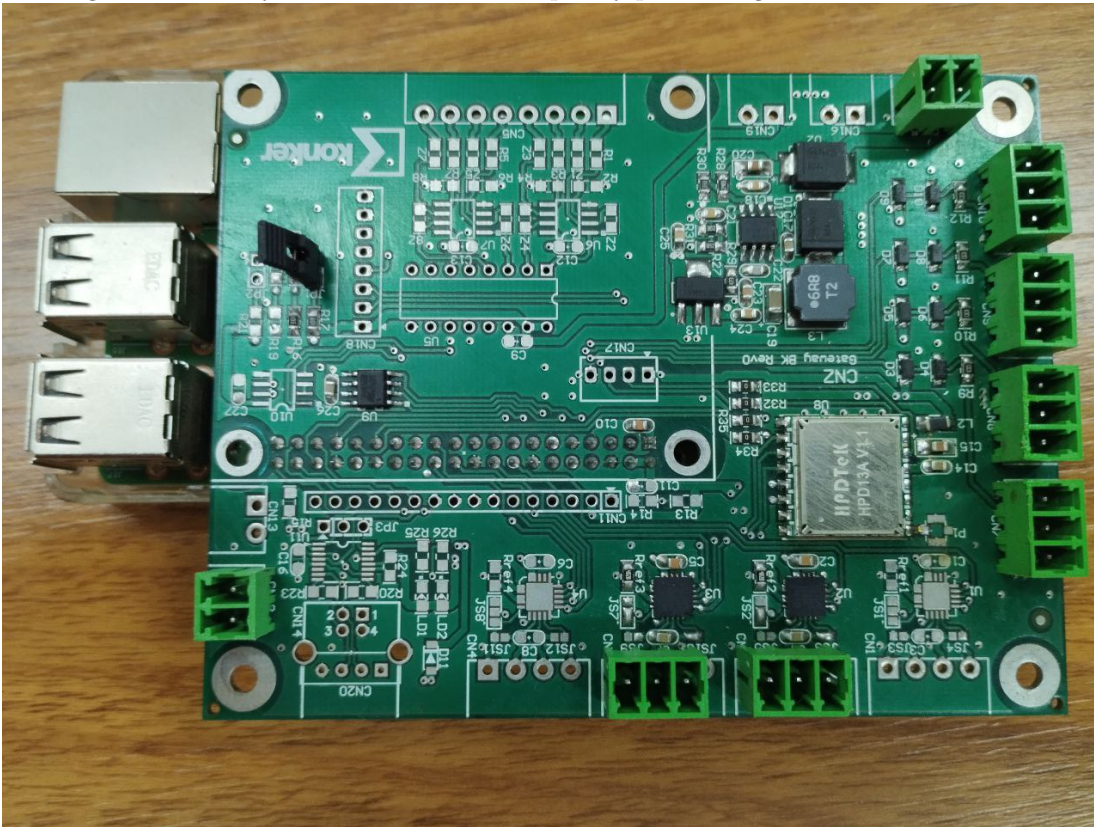
O dispositivo é o próprio CircuLoRa, responsável pela coleta e transmissão de dados. Seu funcionamento envolve o ESP32, que consulta o módulo SIM808 para obter informações como latitude, longitude, altitude e velocidade. Após essa coleta, ele envia esses dados via LoRaWAN utilizando o módulo SX1276 na faixa de frequência de 913,5 MHz a 914,9 MHz. A frequência de transmissão varia em incrementos de 200 KHz, ou seja, os dados são enviados nas frequências 913,5; 913,7; 913,9; ...; 914,9 MHz.

4.1.2 Gateway

O Gateway é responsável por receber todas as informações dos dispositivos, ele atua como um concentrador, coletando as informações de todos os dispositivos e as retransmitindo para a nuvem. O gateway utilizado neste projeto consiste em um Raspberry Pi

0w ou Raspberry Pi 3, em conjunto com um módulo SX1257, que é capaz de operar em todas essas frequências simultaneamente, transformando-o em um gateway multicanal.

Imagem 3: Gateway de canal único com raspberry pi 3 - Imagem meramente ilustrativa



Fonte: Dos autores

4.1.3 Servidor em Nuvem

A aplicação na nuvem é baseada no software de código aberto "Chirpstack". Ela é responsável pelo gerenciamento das aplicações finais, dispositivos e gateways. O Chirpstack gera informações como credenciais e encaminha os dados recebidos pelos gateways para as aplicações finais de acordo com os métodos especificados.

4.2 Aplicação Final

A aplicação final consiste em um serviço que utiliza e armazena os dados fornecidos pelas outras aplicações e dispositivos, fornecendo aos usuários finais alguma forma de interface para interação. O desenvolvimento dessa aplicação final fica a cargo de outros desenvolvedores, além da equipe responsável pelo dispositivo.

4.3 Funcionamento do Software do dispositivo

O software desenvolvido para o dispositivo opera com base em programação concorrente, onde várias tarefas funcionam simultaneamente, comunicando-se por meio de filas e sincronizando-se com a ajuda de semáforos. Essa implementação foi realizada com base no RTOS (Real Time Operating System, ou Sistema Operacional de Tempo Real, em tradução literal) FreeRTOS, que é nativamente incorporado ao ESPIDF.

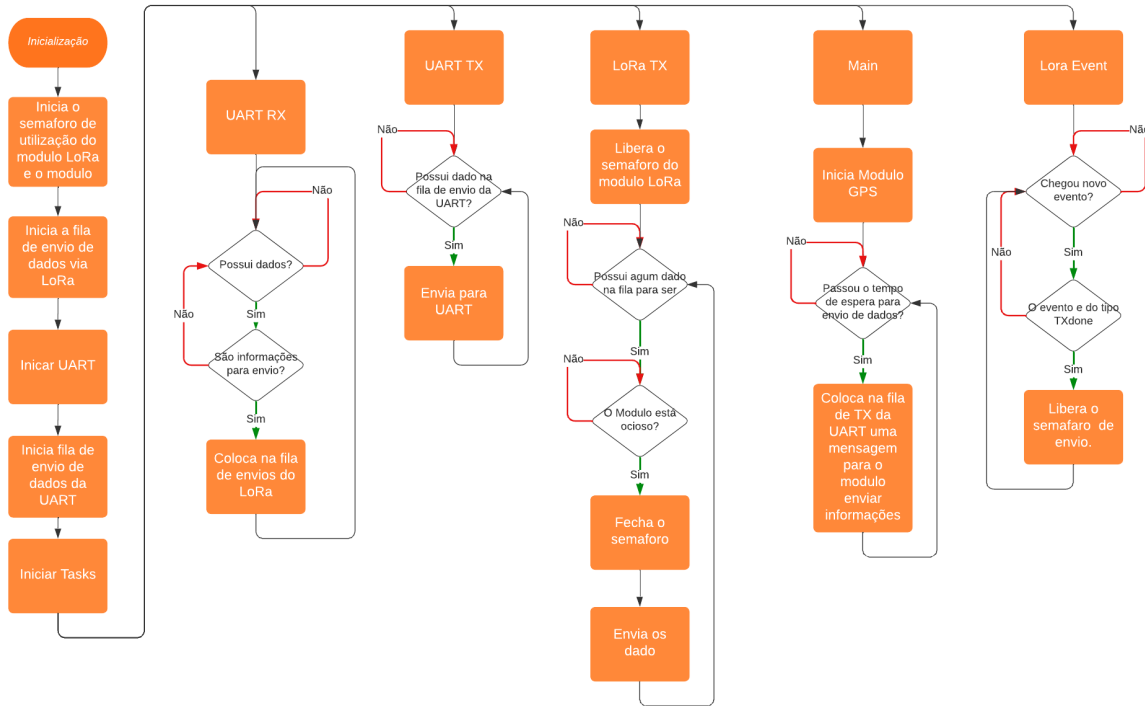
Ao iniciar, o dispositivo executa a inicialização do rádio LoRa e de um semáforo responsável pelo controle do estado ocioso do rádio. Resumidamente, quando uma mensagem é enviada, o semáforo é fechado e só é reaberto após a confirmação do envio. Após a inicialização do rádio LoRa, é criada uma fila pela qual a tarefa de envio de dados via LoRaWAN receberá os dados. Simultaneamente, é iniciada a comunicação via UART com o módulo GPS, bem como a fila para o envio de dados para a tarefa de transmissão (TX) da UART. Com o processo de inicialização concluído, o sistema inicia quatro tarefas responsáveis pela comunicação via UART com o GPS, pelo envio de dados via LoRaWAN e pela temporização entre cada envio de dados.

Das quatro tarefas iniciadas, uma delas é encarregada de monitorar o fluxo de dados na recepção (RX) da UART, verificando se os dados recebidos devem ser transmitidos via LoRaWAN. Caso positivo, essa tarefa insere os dados na fila de envio de dados do LoRa, e a tarefa de envio de dados do LoRa assume o controle sobre os dados. A tarefa de envio de dados via LoRa começa liberando o semáforo e, em seguida, entra em um loop infinito, aguardando a chegada de mensagens na fila. Quando uma nova mensagem é recebida, a tarefa aguarda até que o módulo LoRa esteja em um estado ocioso, momento em que aciona a função de envio dos dados. A tarefa de transmissão (TX) da UART segue um padrão semelhante à tarefa de envio de dados do LoRa. Ela permanece ociosa até que um novo dado chegue na fila para ser transmitido via UART.

Por fim, a tarefa principal tem a responsabilidade exclusiva de configurar o módulo GPS ao iniciar, enviando comandos de inicialização via UART por meio da fila de envio de dados da UART. Após a configuração inicial, envia mensagens via UART, solicitando informações do módulo GPS, a intervalos de tempo previamente configurados.

Resumidamente, essa é a visão geral do funcionamento do sistema, com um fluxograma do software disponível para consulta abaixo na imagem 4.

Imagem 4: Fluxograma com o funcionamento do software do dispositivo



Fonte: Dos autores

5 Resultados

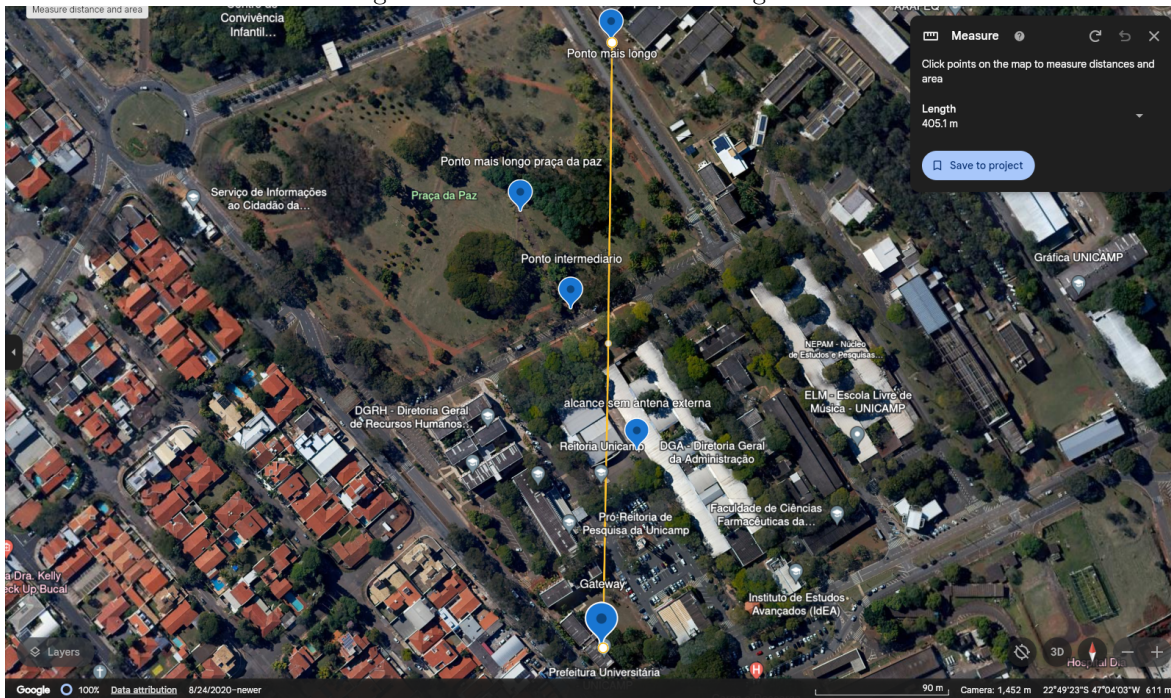
Durante o desenvolvimento do projeto, foram realizados testes para avaliar o desempenho do sistema. Essas avaliações envolveram o uso de um Gateway instalado nas dependências da Prefeitura Universitária da Unicamp e um nó composto por um microcontrolador ESP32 e um Módulo SX1276. As Imagens 3 e 1 oferecem visualizações detalhadas do Gateway e do nó, respectivamente.

Partindo da Prefeitura, foram conduzidas medições de distância, com o autor do projeto movendo-se com o nó enquanto a comunicação com o Gateway permanecia estável. O ponto mais distante registrado foi aproximadamente 400 metros do Gateway, sendo essa distância variável dependendo da localização do nó. Fatores como relevo e presença de árvores influenciaram o alcance do dispositivo.

É relevante destacar que tanto o Gateway quanto o dispositivo estavam operando em um ambiente urbano, cercado por construções de concreto e árvores. Além disso, o Gateway estava posicionado no terceiro andar do prédio da Prefeitura Universitária, com sua antena a aproximadamente 8 metros de altura em relação ao solo. O prédio está

localizado a 616 metros de altitude em relação ao nível do mar, e o ponto mais distante alcançado estava a 605 metros de altitude. A imagem abaixo mostra um panorama resumido dos principais pontos mensurados.

Imagem 5: Pontos marcados no Google Earth



Fonte: Dos autores

A condução de testes mais abrangentes não foi viável devido à ausência de um ou mais Gateways disponíveis em locais estratégicos, caracterizados por maior altitude e melhor visibilidade para as áreas onde o dispositivo realizará seu deslocamento. Essa limitação impede uma análise mais ampla do desempenho em diferentes cenários, e reconhecido a importância de considerar essa variabilidade em futuras etapas de avaliação do sistema. A inexistência de Gateways em posições mais elevadas e com melhor visibilidade restringe a capacidade de explorar plenamente o potencial do dispositivo em diversos ambientes.

É crucial destacar que a variabilidade no terreno, altitude e obstáculos naturais pode ter um impacto significativo no desempenho do sistema de comunicação. Em futuros testes essa limitação deve ser abordada para garantir uma avaliação mais abrangente e representativa do sistema em condições diversas.

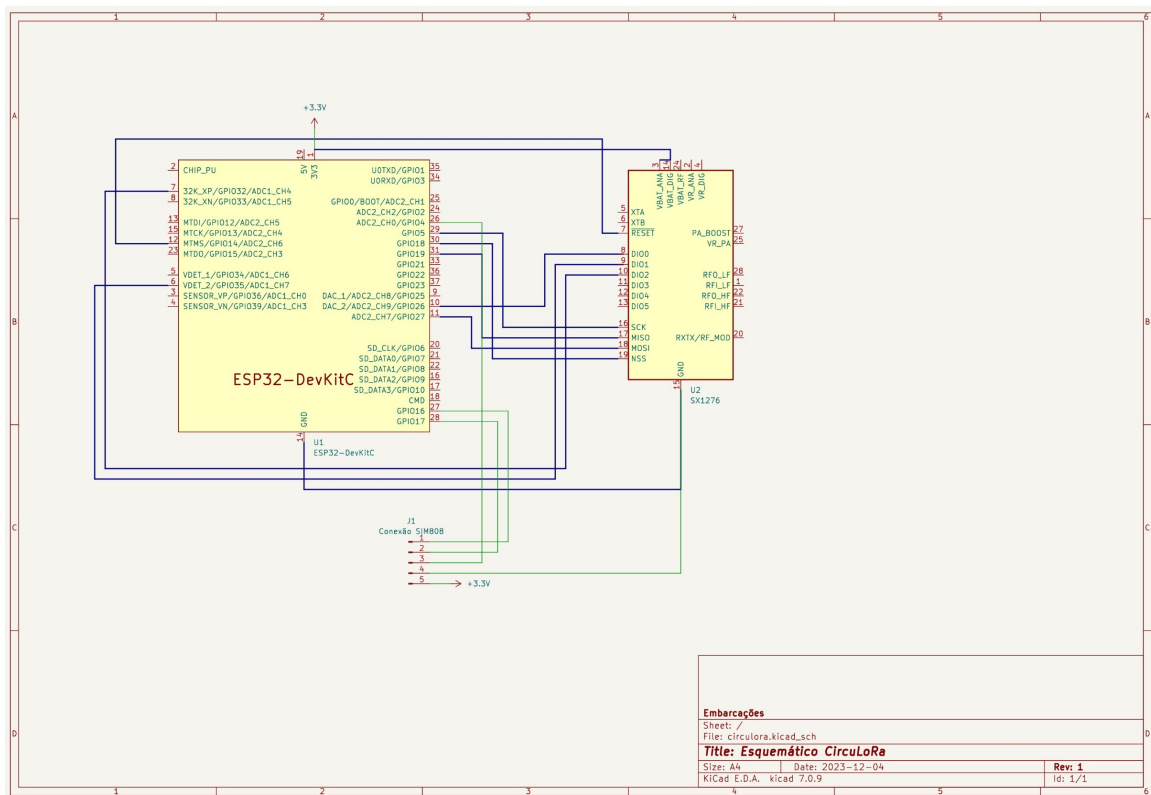
6 Conclusão

A abordagem de monitoramento de ônibus circulares utilizando LoRaWAN tem se mostrado bastante promissora, com resultados satisfatórios em testes preliminares realizados em um ambiente controlado. No entanto, é crucial realizar testes mais abrangentes, como simulações práticas em ônibus ou veículos que reproduzam ambientes próximos ao mundo real. A implementação do sistema é relativamente simples, exigindo apenas a instalação de um dispositivo em cada veículo circular e alguns poucos gateways para cobrir uma área significativa. No entanto, vale ressaltar que a expansão da rede requer o aumento proporcional do número de gateways.

Considerando todos esses aspectos, o CircuLoRa se destaca como uma solução promissora e eficaz, proporcionando uma experiência aprimorada para os usuários e auxiliando na tomada de decisões, como a escolha do ônibus correto e o ponto de espera.

7 Anexos

Anexo 1: Esquemático CircuLoRa



Anexo 2: Repositório do projeto: [CircuLoRa](#)