

DISPOSITIVO IOT DE BAIXO CUSTO PARA SENSORIAMENTO VEICULAR

Pedro Henrique Geanini Dicati, Vladimir Piccolo Barcelos

Instituto Federal de Educação, Ciência e Tecnologia do Mato Grosso do Sul – Três Lagoas/MS

pedro.dicati@estudante.ifms.edu.br, vladimir.barcelos@ifms.edu.br

Resumo

Neste projeto, desenvolveu-se um sistema de sensoriamento veicular usando um Raspberry Pi e o padrão GPS NMEA. Utilizando Python e um banco de dados MySQL, criou-se um *script* para coletar e processar dados em tempo real, relacionados à localização do veículo. Este *script* se conecta ao módulo GPS via pyserial, decodifica os dados NMEA com a pynmea2 e extrai informações essenciais, como latitude, longitude, altitude e data/hora. Para economizar espaço de armazenamento, foi implementado um filtro que verifica alterações significativas (mais de 10 metros) na localização antes de salvar os dados no banco de dados MySQL (MariaDB), otimizando o uso de recursos e garantindo o registro de informações relevantes. Em resumo, esse sistema tem o potencial de ser uma solução eficiente para o sensoriamento veicular, aplicável à diversas finalidades, desde monitoramento de frotas até análises de tráfego e mobilidade urbana, podendo ser ampliado no futuro com novas funcionalidades.

Palavras-chave: Sensoriamento veicular, GPS NMEA, Raspberry Pi.

Introdução

A revolução tecnológica trouxe consigo uma crescente demanda por soluções inovadoras e acessíveis, especialmente na área de sensoriamento veicular, visto que é uma área que possui um constante crescimento na indústria, porém envolve altos custos, tanto de produção quanto de instalação. Nesse contexto, aqui é apresentado a criação de um dispositivo IoT de baixo custo para sensoriamento veicular, utilizando o poderoso Raspberry Pi 3 e o padrão de GPS NMEA, assim como a persistência dos dados em um banco de dados, possibilitando análises e acompanhamento em tempo real dos dados que o veículo gera. Desta forma, unindo essas tecnologias, esta pesquisa colabora com a temática de Internet das Coisas (IoT) e com o setor automobilístico, abrindo portas para diversas aplicações práticas. Essa iniciativa exemplifica a busca contínua por soluções tecnológicas acessíveis e inovadoras que podem revolucionar a forma como interagimos com nossos veículos e ambientes urbanos, abrindo, desta forma, portas para a construção de uma sociedade interconectada e inteligente.

Metodologia

Esta pesquisa aborda a criação de um processo de captura de dados de geolocalização e realiza o processamento dos

mesmos. A presente pesquisa iniciou-se estudando o protocolo NMEA, onde entendeu-se como podia ser realizada a captura dos dados e como funcionava o módulo GPS utilizado no projeto.

No primeiro momento, utilizou-se um computador pessoal, com Python instalado e as devidas bibliotecas do mesmo, para fazer a captura dos dados que eram obtidos por meio de uma conexão serial do módulo GY-GPS6MV2 à um conversor serial para USB, ficando a conexão da seguinte forma:

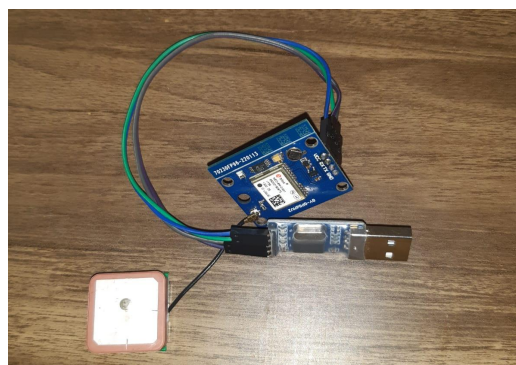


Figura 1. Captura dos dados do GPS

Baseado nisso, foi possível compreender a forma como os dados das localizações e demais informações eram geradas, então iniciou-se o desenvolvimento de como seria a construção da arquitetura, utilizando um Raspberry Pi 3 Model B+, para a conexão do módulo GPS, via serial. Com esta arquitetura, foi possível conectar o mesmo ao Raspberry e obter as informações via serial. A arquitetura de conexão com o Raspberry é mostrada na Figura 2 a seguir.



Figura 2. Conexão Raspberry Pi 3 com módulo GPS

A partir do sucesso dessa conexão, iniciou-se o desenvolvimento do *script* em Python para a captura dos dados e processamento dos mesmos. Como início, foi iniciado por obter os dados básicos de localização (latitude e longitude), posteriormente a isso tentou-se obter os demais dados (altitude, velocidade e data e hora). Neste passo, alguns percalços foram encontrados, principalmente a falta de recebimento de informações do módulo GPS, principalmente ligados ao protocolo NMEA, como variabilidade de localização em mesmo ponto, com isso, foi pensado em calcular a distância entre os pontos para evitar o armazenamento de informações muito próximas que não seriam relevantes para o projeto. Assim, para implementar uma distância mínima entre as capturas, foi implementado um limite de tempo, porém não foi obtido um grande sucesso, pois uma vez que o veículo estivesse parado, iria ser capturada a mesma localização, não solucionando o problema. Então, em pesquisas, encontrou-se a Fórmula de Haversine, usada para calcular distâncias entre dois pontos na superfície de uma esfera usando a latitude e a longitude dos dois pontos (KETTLE, 2017). A fórmula é a seguinte:

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\phi_2 - \phi_1}{2} \right) + \cos(\phi_1) \cos(\phi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

Figura 3. Fórmula de Haversine

Assim, conseguiu-se calcular a distância entre o ponto anterior com o ponto atual, possibilitando que a gravação no banco somente pontos que tivessem mais de 10 metros de distância entre si.

Paralelo ao desenvolvimento do *script* em Python para a captura dos dados, criou-se um banco de dados no MySQL, para armazenar os dados e permitir que fosse possível visitar os dados sempre que fosse preciso. Assim, construiu-se uma tabela com o nome **localizacoes_gps**, com as colunas sendo o ID da localização, ID do veículo, latitude e longitude, velocidade, altitude e data e hora para controle e mapeamento. Então ao conseguir capturar o dado do módulo do GPS e realizar a validação da distância dos pontos, persistimos esse dado nessa tabela do banco, permitindo que na próxima iteração, possamos utilizar o mesmo para o cálculo da distância, visto que este é o mais próximo, e também no momento de acompanhamento das informações, utilizar os dados armazenados na tabela, para o mapeamento das localizações para a visualização da rota.

Desta forma, o processo de desenvolvimento do projeto, ficou dividido em três marcos, primeiro o entendimento de como o dado chegava do módulo, e para isso também usamos um manual de referência da SiRF Technology para entender o que cada mensagem transmite de informação, segundo a captura das informações com o *script* Python e, por fim, a persistência dessas informações em um banco de

dados. Para a visualização, foi usada a ferramenta Streamlit, um *framework* Python que auxilia equipes a criarem e compartilharem aplicativos Web personalizados sem a necessidade de conhecer ferramentas de *front-end* ou de *deploy* de aplicações (PECHOTO, 2022), para a construção de um mapa, onde é traçado a rota que o veículo seguiu, lendo as localizações do banco de dados. Com o mapa sendo gerado, a visualização fica da seguinte forma:

Projeto Bahama - Dispositivo IoT de baixo custo para sensoriamento veicular

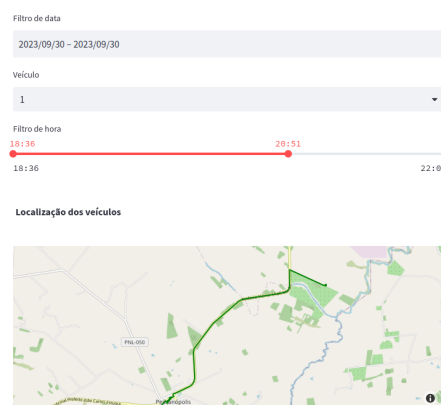


Figura 4. Visualização das localizações no Streamlit

Dessa forma, concluiu-se o desenvolvimento com o sucesso da implementação e captura das informações, abrindo novas portas para a criação de ferramentas e aplicações que possibilitem a melhoria do nosso cotidiano pessoal e também de empresas, com facilidade de captura de informações e visualização, além de possibilitar tornar mais coisas no nosso dia “inteligentes”, conseguindo interconectar e facilitar ainda mais o cotidiano do mundo.

Resultados e Discussão

Após o desenvolvimento e criação da aplicação e *script*, foi possível notar que é totalmente viável a construção de um dispositivo que consegue capturar informações em tempo real de veículos e armazená-las em um outro local. Alguns problemas e conclusões que abrem janelas para possíveis melhorias no projeto, como a variabilidade do sinal do padrão NMEA, que por muitas vezes pode ser perdido e retomado durante um trajeto por mal tempo climático ou até perda de conexão com os satélites, podendo ocasionar perdas de informações, porém, em um primeiro momento, se mostrou muito eficaz e uma solução de grande escalabilidade, visto seu baixo custo. A eficácia da obtenção do ponto geográfico em que o dispositivo está, demonstra-se pela exatidão de os pontos ficam dentro dos limites da rua, quando colocado dentro de um carro em movimento, conforme a imagem abaixo, onde o sistema foi testado na cidade de Fernandópolis/SP:

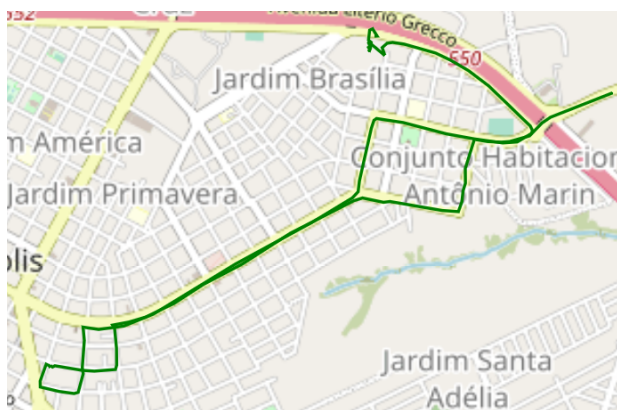


Figura 4. Visualização das localizações no Streamlit

Em relação a velocidade e escalabilidade do sistema desenvolvido, mostrou-se muito notório a eficiência da linguagem Python em trabalhar com módulos seriais e também processamento de informações de GPS, com uma velocidade de gravação e leitura excelente, possibilitando a visualização em tempo real das informações. Neste ponto é importante ressaltar, que a velocidade da Internet é fundamental para a otimização e boa experiência de tempo real do usuário, pois por ela que trafegam os dados para armazenamento dos dados e posteriormente visualização.

Em suma, os resultados obtidos demonstraram-se animadores, o que ressalta que um dispositivo IoT de baixo custo é totalmente viável, contudo, é importante ressaltar que alguns pontos, como a eficiência da obtenção das localizações, qualidade do dado e agilidade, devem ser levados em consideração para a continuidade do desenvolvimento desta solução inovadora que muito tem a revolucionar a indústria.

Considerações Finais

O desenvolvimento de um dispositivo de IoT para o sensoriamento de veículos mostra-se como uma tecnologia desejável da área automobilística, podendo proporcionar segurança, eficiência e rastreamento das atividades de forma adequada, tanto para empresas como para veículos pessoais.

A integração dos sistemas (captura das informações, armazenamento e visualização) se mostra ponto fundamental para eficácia da solução. A solução tem o potencial de integração com vários dispositivos e soluções em prol da criação de um sistema único, capaz de atuar como uma espécie de caixa preta veicular.

Sendo assim, como trabalhos, futuros pretende-se continuar o desenvolvendo deste projeto com a ampliação das funcionalidades, capturando não somente a localização, mas outras informações do veículo, como temperatura, aceleração, eficiência energética, defeitos que venham a surgir e demais informações que são pertinentes e que possam ser capturadas, para melhorar ainda mais essa

solução inicial. Essas informações serão obtidas por meio da interface OBD-II, diretamente da central do veículo. Com isso, associado a um sistema web que já se encontra em desenvolvimento, o protótipo terá potencial de ser uma solução completa para coletar, exibir e administrar dados veiculares, podendo ser utilizados em empresas de frotas, seguradoras, pessoas físicas, entre outros.

Agradecimentos

Primeiramente agradeço a Deus por me dar a oportunidade e proporcionar a sabedoria para conduzir esse projeto, mesmo perante tantas dificuldades encontradas pelo caminho. Também agradeço a Natália Celestino Farina, minha amada, que durante todo esse período, não deixou um minuto sequer de estar ao meu lado e me apoiar. Agradeço também ao meu orientador, Vladimir Piccolo Barcelos, sem ele esse projeto nunca teria saído do papel, muito menos teríamos avançado como conseguimos. Ademais, agradeço ao Instituto Federal de Mato Grosso do Sul, pela oportunidade e apoio financeiro para o desenvolvimento da pesquisa e, por fim, agradeço a todos os colegas e a família que sempre me motivaram e me ajudaram a estar onde estou hoje.

Referências

KETTLE, Simon. **Distance on a sphere: The Haversine Formula**. 2017. Disponível em: <https://community.esri.com/t5/coordinate-reference-systems-blog/distance-on-a-sphere-the-haversine-formula/ba-p/902128>. Acesso em: 13 mar. 2023.

SIRF TECHNOLOGY, INC.. **NMEA Reference Manual**. San Jose: Sirf Technology, 2007. Disponível em: <https://www.sparkfun.com/datasheets/GPS/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf>. Acesso em: 12 jan. 2023.

PECHOTO, Murilo Marin. **Streamlit: uma forma fácil de criar e compartilhar sua aplicação de dados**. 2022. Disponível em: <https://eldorado.org.br/blog/streamlit-uma-forma-facil-de-criar-e-compartilhar-sua-aplicacao-de-dados/>. Acesso em: 28 mar. 2023.

IOT DEVICE FOR LOW-COST VEHICULAR SENSING

Abstract: In this project, a vehicle sensing system was developed using a Raspberry Pi and the NMEA GPS standard. Using Python and a MySQL connection, a script was created to collect and process real-time data related to the vehicle's location. This script connects to the GPS module via pyserial, decodes the NMEA data with pynmea2 and extracts essential information such as latitude, longitude, altitude and date/time. To save storage space, a filter has been implemented that checks for significant

changes (more than 10 meters) in the location before saving the data in the MySQL database (MariaDB), optimizing the use of resources and ensuring that relevant information is recorded. In summary, this system has the potential to be an efficient solution for vehicle sensing, applicable to various purposes, from fleet monitoring to traffic analysis and urban mobility, and can be expanded in the future with new functionalities.

Keywords: *Vehicular sensing, NMEA GPS, Raspberry Pi.*