

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
DE MINAS GERAIS

GRADUAÇÃO EM
ENGENHARIA MECATRÔNICA



Wireless Sampling Network

Erick Nathan Martins Alves
Gabriel Alvarenga Nogueira
Kleber Rodrigo da Silva Junior
Miguel Lukas Rodrigues

Professor(a): Dr. Alan Mendes Marotta

7 de julho de 2023

1 Introdução

O uso de veículos automotores é algo indispensável nos dias de hoje, principalmente por proporcionarem um menor tempo de deslocamento em um dado trajeto. Seja em veículos próprios ou através do transporte público, todos precisam, diariamente, de se locomover de onde residem ao local de trabalho, por exemplo. Por isso, busca-se cada vez mais que essa tarefa seja feita com maior segurança, de forma mais eficiente e com maior conforto.

Desde a etapa de concepção até a etapa de construção de um determinado veículo, diversos parâmetros são analisados e medidos a fim de proporcionar aos seus ocupantes uma experiência agradável de utilização. Uma das principais variáveis analisadas para isso é a vibração mecânica transmitida aos indivíduos. Um veículo que atenua pouco as vibrações induzidas pelo pavimento para o condutor é considerado qualitativamente desconfortável e, por isso, tende a ter menor aceitação por parte do mercado consumidor.

Nesse contexto, nota-se a importância da análise dessa variável de projeto no desenvolvimento de um automóvel. Percebe-se, contudo, que apenas a verificação qualitativa não é suficiente para auxiliar na tomada de decisão da engenharia em relação à dinâmica veicular. Surge, portanto, a necessidade de instrumentação adequada capaz de captar essas vibrações, produzindo dados relevantes que auxiliem na compreensão do fenômeno e na otimização dos parâmetros do veículo por parte dos projetistas.

1.1 Objetivos

Este trabalho tem por objetivo a construção de uma rede de dispositivos móveis sem fio baseados em microcontroladores que seja capaz de captar e transmitir dados vibracionais veiculares.

2 Metodologia

2.1 Universal Assynchronous Receiver Transmpter (UART)

O Universal Asynchronous Receiver Transmitter (UART) é um componente eletrônico utilizado em telecomunicações para estabelecer a comunicação serial entre dispositivos. Ele permite a transferência de dados de forma assíncrona, o que significa que não há uma linha de relógio compartilhada entre os dispositivos. Um UART é composto por um transmissor (Transmitter) e um receptor (Receiver). O transmissor converte os dados paralelos em dados seriais e os envia em formato assíncrono. O receptor, por sua vez, recebe os dados seriais, converte-os de volta para o formato paralelo e os envia para o dispositivo receptor.

O UART utiliza um esquema de codificação chamado de "assíncrono", que inclui os seguintes elementos:

- **Bit de Start:** É um bit de valor lógico 0 que indica o início de um byte de dados. Ele sinaliza ao receptor que uma transmissão está prestes a ocorrer.
- **Dados:** Os bits de dados contêm a informação a ser transmitida. A quantidade de bits de dados por byte pode variar, sendo comum a utilização de 7 ou 8 bits de dados.
- **Bit de Paridade:** É um bit opcional que pode ser utilizado para verificar a integridade dos dados transmitidos. Ele pode ser utilizado para detectar erros de transmissão.

- **Bits de Stop:** São um ou mais bits de valor lógico 1 que indicam o final de um byte de dados. Eles sinalizam ao receptor que a transmissão foi concluída.

O UART é amplamente utilizado em telecomunicações e em uma variedade de aplicações, incluindo:

- **Comunicação entre dispositivos:** O UART é usado para a comunicação serial entre dispositivos, como microcontroladores, sensores, módulos de comunicação sem fio e periféricos. Ele permite a transferência de dados entre esses dispositivos de forma confiável e eficiente.
- **Conexão com dispositivos externos:** O UART é frequentemente usado para conectar dispositivos externos, como modems, GPS, displays de caracteres, leitores de código de barras e muitos outros dispositivos que utilizam comunicação serial.
- **Depuração e monitoramento:** O UART é amplamente utilizado para depuração e monitoramento de sistemas embarcados. Ele permite a comunicação serial com o sistema para visualização de informações, depuração de código e registro de eventos.
- **Interface com computadores:** O UART também é utilizado para fornecer uma interface serial entre dispositivos eletrônicos e computadores. Essa interface é comumente usada para conexões RS-232, RS-485 ou USB-Serial.

2.2 Inter Integrated Circuit (I2C)

O Inter Integrated Circuit (I2C) é um protocolo de comunicação serial síncrono amplamente utilizado na área de telecomunicações. Ele permite a comunicação entre dispositivos eletrônicos de forma eficiente e confiável, usando apenas dois fios: um para transmitir dados (SDA - *Serial Data Line*) e outro para controlar o fluxo de dados (SCL - *Serial Clock Line*).

O I2C é um protocolo mestre-escravo, onde um dispositivo mestre controla a comunicação com um ou mais dispositivos escravos. O mestre é responsável por iniciar e controlar as transmissões, enquanto os escravos respondem às solicitações do mestre. A comunicação ocorre em uma estrutura de barramento, onde todos os dispositivos estão conectados em paralelo. Durante a transmissão, o mestre envia pacotes de dados para os escravos. Cada pacote é composto por um endereço do escravo seguido pelos dados a serem transmitidos. O mestre inicia a comunicação gerando um sinal de START, seguido pelo endereço do escravo e um bit indicando se a operação é de leitura ou escrita. Os escravos respondem ao mestre no tempo alocado. Além disso, o I2C também possui um mecanismo de reconhecimento, onde tanto o mestre quanto os escravos podem reconhecer a recepção dos dados. O reconhecimento é feito por meio de um sinal ACK (acknowledge) transmitido pelo dispositivo receptor. Isso garante a integridade da transmissão e permite que o mestre saiba se o dispositivo escravo está disponível e operacional.

O protocolo I2C é amplamente utilizado em diversas aplicações de telecomunicações. Alguns exemplos incluem:

- **Configuração de dispositivos:** O I2C é comumente usado para configurar e controlar dispositivos como amplificadores, moduladores, demoduladores, conversores analógico-digitais e vice-versa. Ele permite que o dispositivo mestre configure os parâmetros e

opere os dispositivos escravos de maneira eficiente.

- **Gerenciamento de energia:** O I2C é usado em sistemas de gerenciamento de energia para controlar e monitorar o consumo de energia em diferentes componentes e dispositivos. Ele permite que o mestre acesse informações sobre o consumo de energia, ajuste as configurações de energia e controle o modo de operação dos dispositivos escravos.
- **Sensores e dispositivos periféricos:** Muitos sensores, como sensores de temperatura, pressão, umidade e acelerômetros, são projetados para se comunicar via I2C. Além disso, dispositivos periféricos, como displays LCD e memórias EEPROM, também podem ser controlados usando o protocolo I2C.
- **Redes de comunicação:** O I2C é usado em alguns sistemas de comunicação em rede, como redes locais de área reduzida (LANs) e sistemas de comunicação de dados entre dispositivos próximos. Nesses casos, o I2C é usado para trocar informações de controle e configuração entre os dispositivos da rede.

2.3 Protocolos UDP/IP

A transmissão de dados via UDP (*User Datagram Protocol*) é um protocolo de comunicação de transporte que opera no modelo de serviço não orientado a conexão. Diferentemente do TCP (*Transmission Control Protocol*), que oferece uma comunicação confiável e orientada a conexão, o UDP é um protocolo simples que fornece uma entrega de dados não garantida e sem controle de fluxo.

Ao usar o UDP, os dados são divididos em pacotes individuais chamados datagramas. Cada datagrama é então enviado independentemente ao destino, sem estabelecer uma conexão prévia entre o emissor e o receptor. Essa abordagem de comunicação torna o UDP mais leve e mais rápido em comparação com o TCP, pois não possui o *overhead* adicional de estabelecer e manter uma conexão. No entanto, a falta de mecanismos de controle de erro e garantia de entrega no UDP significa que ele pode ser menos confiável em comparação com o TCP. Os datagramas UDP podem ser perdidos, duplicados ou chegar fora de ordem, e não há mecanismos embutidos para lidar com essas situações. Cabe ao aplicativo ou à camada superior implementar a lógica para tratar desses problemas, se necessário.

A transmissão de dados via UDP é frequentemente usada em situações em que a velocidade e a latência são mais importantes do que a confiabilidade. Alguns exemplos comuns de aplicativos que usam UDP incluem:

1. Streaming de mídia: serviços de streaming de áudio e vídeo geralmente utilizam UDP para transmitir dados em tempo real. Embora possa haver perda de alguns pacotes, a entrega rápida é prioritária para evitar atrasos e interrupções na reprodução.
2. Jogos online: muitos jogos online utilizam UDP para enviar dados entre o servidor e os clientes. A velocidade de transmissão é fundamental para manter a jogabilidade fluida e responsiva, e a perda ocasional de pacotes pode ser tolerada em detrimento da latência reduzida.
3. Transmissão de dados em tempo real: aplicações que requerem transmissão em tempo real, como voz sobre IP (VoIP) e videoconferência, podem usar UDP. Embora a qua-

lidade possa ser afetada por pacotes perdidos, a baixa latência é crucial para manter uma comunicação eficiente.

4. DNS (*Domain Name System*): o protocolo DNS utiliza UDP para consultas simples de resolução de nomes de domínio. Como as consultas DNS geralmente são pequenas e não exigem uma comunicação confiável, o UDP é uma escolha adequada.

É importante ressaltar que, ao utilizar o UDP, o aplicativo precisa lidar com questões como controle de erros, retransmissão de pacotes perdidos e garantia de entrega, se essas funcionalidades forem necessárias. Essas tarefas são geralmente tratadas em níveis mais altos da pilha de protocolos ou no próprio aplicativo.

2.4 Vibração Forçada

Vibração forçada é definida como a vibração provocada pela aplicação de forças e momentos externos ao sistema (Inman, 2014). Ao contrário da vibração livre, que estuda o comportamento do sistema com um dado deslocamento ou velocidade inicial, mas sem forças externas, na vibração forçada busca-se entender qual o comportamento apresentado pela estrutura quando submetida a forças e torques externos.

Nos veículos, busca-se estudar a vibração induzida aos ocupantes. Essas vibrações podem vir de diversas fontes, das irregularidades do pavimento, da combustão do motor, do atrito aerodinâmico, entre outras. O estudo dessas vibrações e da influência de cada um dos parâmetros da suspensão do veículo são de extrema importância para a tomada de decisão na fase de projeto por parte dos engenheiros.

2.5 Fast Fourier Transform (FFT)

A Transformada Rápida de Fourier (FFT, do inglês Fast Fourier Transform) é um algoritmo eficiente para calcular a Transformada de Fourier Discreta (DFT, do inglês Discrete Fourier Transform) de um sinal discreto. A FFT é amplamente utilizada no âmbito da análise de sinais, pois permite decompor um sinal em suas componentes de frequência, revelando informações importantes sobre sua composição espectral.

A análise de sinais é uma área da engenharia e matemática que estuda as propriedades dos sinais em diferentes domínios, como o domínio do tempo e o domínio da frequência. A transformada de Fourier é uma ferramenta fundamental na análise de sinais, pois permite converter um sinal do domínio do tempo para o domínio da frequência. A Transformada de Fourier Discreta (DFT) é uma forma de transformada de Fourier que é aplicada a sinais discretos, ou seja, sequências de valores amostrados em intervalos regulares. No entanto, o cálculo direto da DFT para um sinal de tamanho N requer uma complexidade computacional de $O(N^2)$, o que pode ser computacionalmente custoso para sinais longos.

A FFT é um algoritmo que foi desenvolvido para acelerar o cálculo da DFT, reduzindo sua complexidade computacional para $O(N \log N)$. Essa redução significativa na complexidade torna a FFT muito mais eficiente para calcular a DFT de sinais longos. A ideia básica por trás da FFT é explorar a propriedade da simetria complexa conjunta dos coeficientes da DFT para reduzir o número de operações necessárias. O algoritmo divide recursivamente o sinal em subgrupos menores, calcula a DFT de cada subgrupo e, em seguida, combina as DFTs parciais para obter a DFT completa do sinal original. Existem várias variantes do al-

goritmo FFT, como a FFT de Cooley-Tukey, que é a mais conhecida e amplamente utilizada. Essa variante divide o sinal em subgrupos de tamanho par ou ímpar, aplicando a DFT nesses subgrupos e combinando as DFTs parciais para obter a transformada completa.

A FFT é um componente essencial em uma variedade de aplicações que envolvem análise de sinais, como processamento de áudio, processamento de imagens, comunicações digitais, análise espectral, entre outras. Ela permite que os engenheiros e cientistas extraiam informações valiosas sobre os sinais, identificando componentes de frequência, detectando padrões e realizando filtragem e análise espectral. Em resumo, a Transformada Rápida de Fourier é um algoritmo poderoso que torna o cálculo da Transformada de Fourier Discreta eficiente, permitindo a análise de sinais no domínio da frequência. Sua utilização é ampla e essencial em muitas áreas da ciência e engenharia, fornecendo uma ferramenta fundamental para entender e processar sinais em diferentes contextos.

3 Princípio de Funcionamento

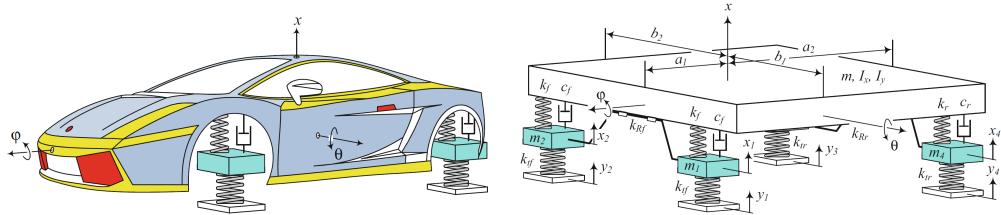


Figura 1: Modelo vibracional Full-Car.

O veículo como mostrado na Figura 1 pode ser modelado como um sistema de 7 graus de liberdade, 3 relativos aos movimentos de *bounce*, *roll* e *pitch* da carroceria e 4 relativos aos movimentos das rodas. O sistema proposto para aquisição de dados vibracionais tem como principal objetivo captar os 3 movimentos da carroceria.

Para estudar a transmissibilidade do movimento do pavimento para os ocupantes do veículo, a topologia adotada consiste em instalar um acelerômetro em cada uma das quatro rodas e um acelerômetro no interior do automóvel. Dessa forma, é possível avaliar os parâmetros da suspensão, como a massa, rigidez e o amortecimento, visando otimizar a relação conforto/*handling* proporcionado aos usuários.

A Figura 2 mostra como os dispositivos serão conectados em rede. Cada uma das placas se conecta ao notebook (*access point*) através dos módulos Wi-Fi presentes no ESP-01, utilizando o protocolo UDP/IP. Essa topologia de conexão é conhecida como *estrela*, justamente por contar com um elemento mestre que se conecta a todos os demais componentes escravos.

A Figura 3 mostra como é o funcionamento interno de cada uma das placas. Os dados de aceleração fornecidos pelo CI MPU6050 são transmitidos para o Attiny 85 através do protocolo I2C. O Attiny, por sua vez, transmite para o ESP-01 para que os dados sejam, de forma sem fio, entregues ao Notebook.

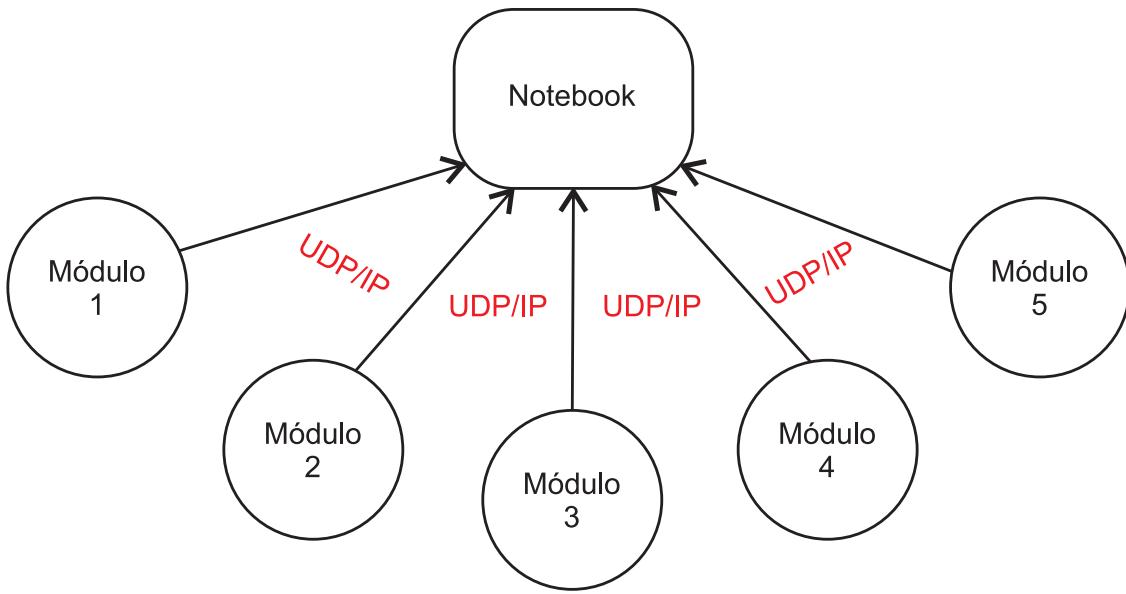


Figura 2: Diagrama de funcionamento em rede.



Figura 3: Diagrama de funcionamento interno.

4 Eletrônica

O circuito de cada módulo deveria fazer a comunicação entre o acelerômetro e um microcontrolador desejado. Para facilitar a instrumentação de qualquer veículo, seria essencial que o mesmo fosse embarcado, portátil e sem fio. Assim, a fonte de alimentação escolhida para o projeto foram baterias de lítio do tipo 18650, com 3.7V e 2500mAh de tensão e capacidade de carga nominais.

O microcontrolador escolhido, a princípio, seria o Attiny 85, pois a ideia inicial seria utilizar módulos HC12 para a transmissão dos dados através de serial transparente, pelo protocolo UART. Isso seria possível, pois o Attiny 85 possui portas I2C em *hardware* e bibliotecas para criar uma comunicação serial via *software*. Entretanto, o problema encontrado nessa ideia foi o tempo de amostragem muito alto, uma vez que, para esse sistema pensado, seria necessário o chaveamento de canais do HC12, para que um mesmo módulo mestre pudesse ouvir cada um dos escravos.

Posteriormente, o projeto foi modificado para funcionar via *wifi*, utilizando a plataforma ESP-01, baseado no ESP8266 e NodeMCU, possibilitando assim, a criação de um sistema de *clients* e *hosts*, com o *access point* sendo um notebook, o qual receberia os pacotes de dados de cada módulo e os armazenaria em memória interna, para posterior análise, processamento e *plotagem*.

Foram adicionados ainda, alguns circuitos auxiliares, para funções específicas e que, ao embarcá-los em cada módulo, facilitaria seu uso, como: módulo de recarga e controle de corrente para a bateria e também um circuito de comunicação entre o Attiny e o computador, com um conector KK para conexão USB. E ainda, foram reservados pinos (conectores KK) para *debug* (GND, RX e TX), sendo possível realizar acoplar monitores seriais para o ESP e para o attiny. As comunicações USB e para *debug* podem ser feitas através de um cabo montado para esse propósito e de outra plataforma externa, como o ESP-32, mostrados na Figura 4.

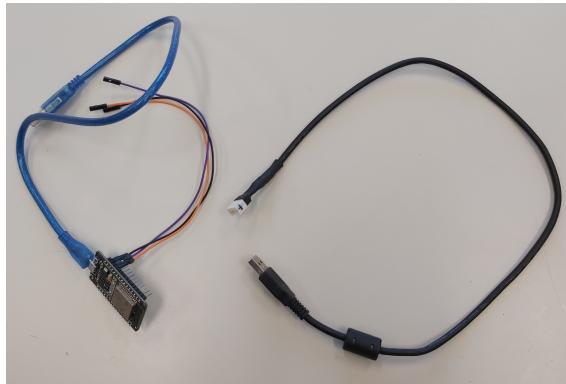


Figura 4: Acessórios para programação e debug.

Assim definido, o circuito final foi desenhado no Proteus, conforme Figura 5.

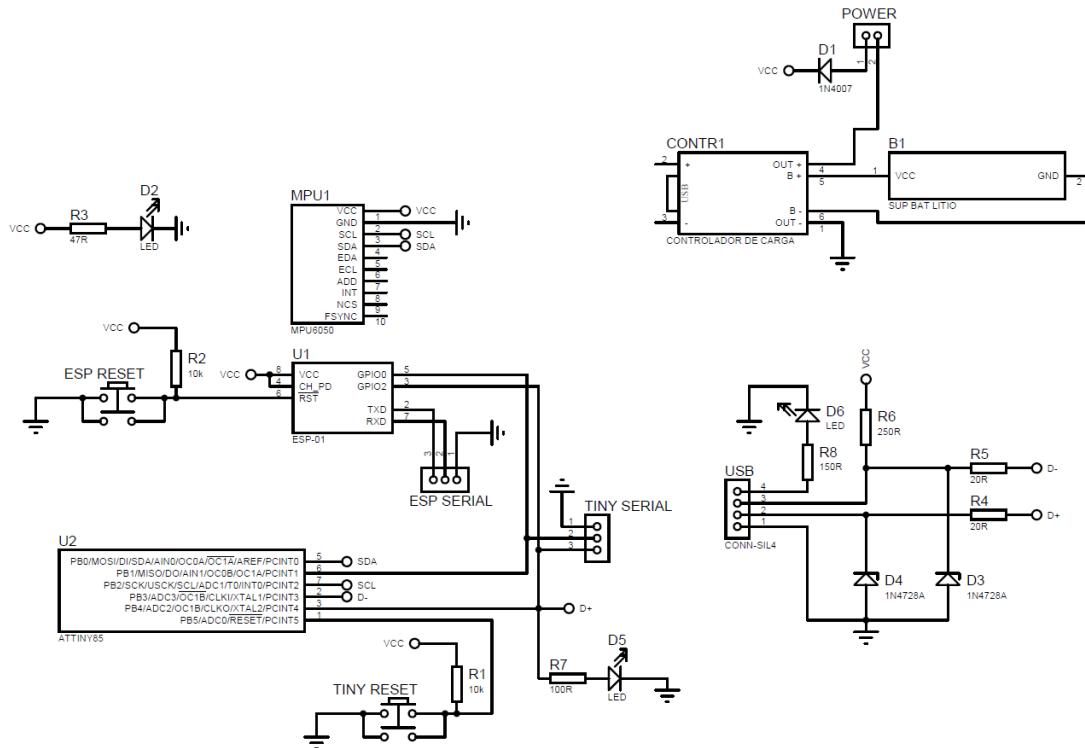


Figura 5: Circuito final de cada módulo.

Como os módulos seriam acoplados a um veículo, é ideal que os componentes fossem soldados para evitar mal contato e/ou desconexões indesejadas, principalmente porque o sistema estará submetido a intensas vibrações.

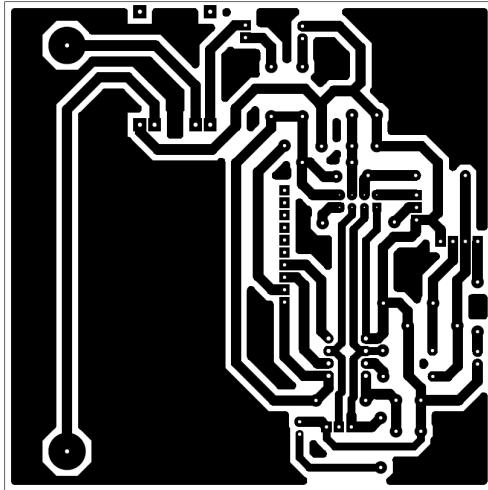


Figura 6: Layout para confecção das placas.

Por fim, as placas finais, já com os componentes soldados, são ilustradas nas Figuras 7 e 8.

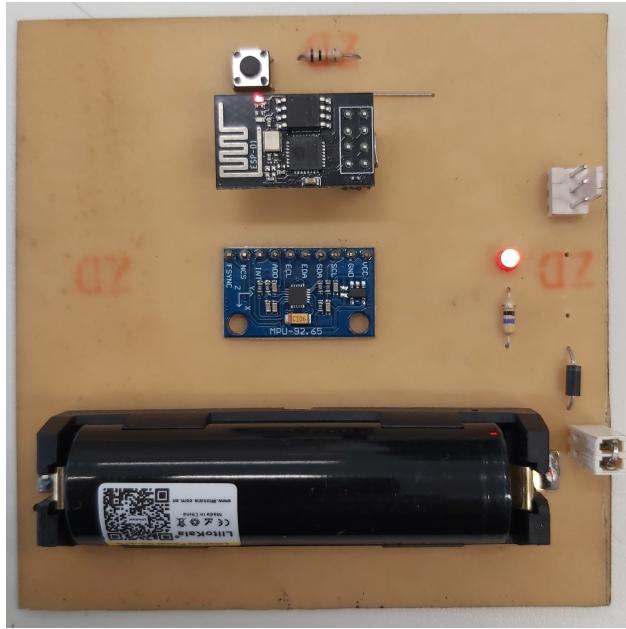


Figura 7: Placa montada com esp-01.

5 Programação

Para realizar a transmissão dos dados, foi utilizada a linguagem de programação *MicroPython*, que é uma versão reduzida do *Python* para microcontroladores.

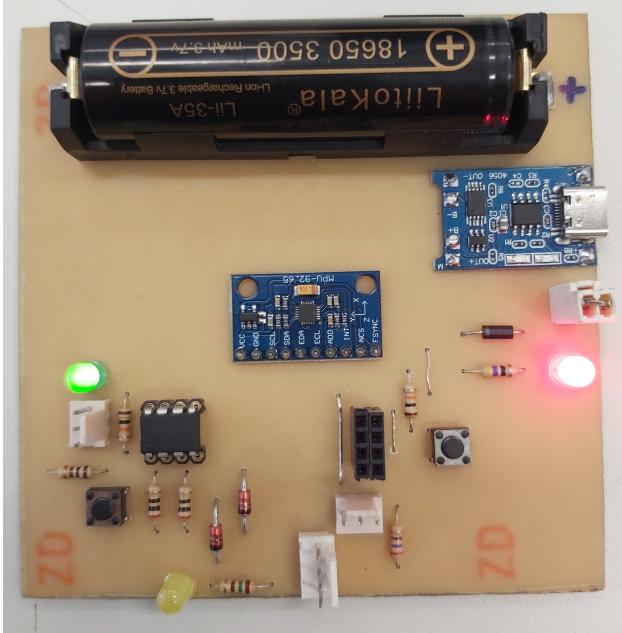


Figura 8: Placa montada com attiny85 e esp-01.

Resumidamente, os dados do acelerômetro são lidos através do protocolo I2C, e então entram em uma fila que possui capacidade máxima. Quando essa fila enche, os dados da fila são enviados juntamente de suas respectivas posições via UDP para um servidor que fica no notebook, e então os espaços de memória do microcontrolador são liberados para receber novos dados. É necessário transmitir a posição do dado na fila para garantir que os dados recebidos serão enfileirados na ordem correta.

6 Acabamento e Fixação

Em termos de fixação, surge a necessidade de projetar um modelo de suporte para acomodar a placa de circuito, protegendo seus componentes internos, visto que a mesma será submetido a ambientes impróprios para deixar exposto. Para o projeto utilizou-se do software Inventor da empresa Autodesk, modelando todas as necessidades do *case*.

Para o modo de fixação utilizou-se do recurso magnético, utilizando ímãs permanente, garantindo assim uma fixação rápida e eficaz. O suporte foi desenvolvido em duas partes, sendo a primeira, a parte que acomoda a placa, como pode ser visualizado na Figura 9. Os furos na parte inferior do *case*, foram projetados para a visualização dos LEDs de indicação da placa e também para a fixação de um botão para acionar ou desligar a placa.

Para a segunda parte do *case*, considera-se a parte que acomoda os parafusos de fixação, que foi projetado os furos para o parafuso ficar interno na peça. Em termos de fixação dos ímãs, utilizou-se do recurso de relevo com profundidades de acordo com a altura dos ímãs que foram retirados de sucatas eletrônicas. A posição de cada ímã foi pensado em versatilidade para a fixação do suporte no veículo, utilizando a torre do amortecedor como foco de fixação.

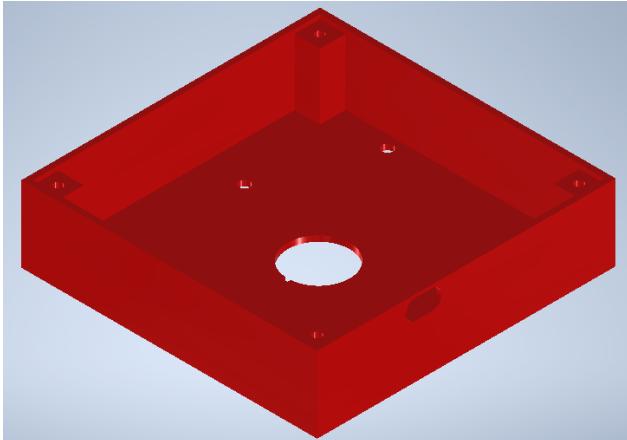


Figura 9: Parte superior do *case*

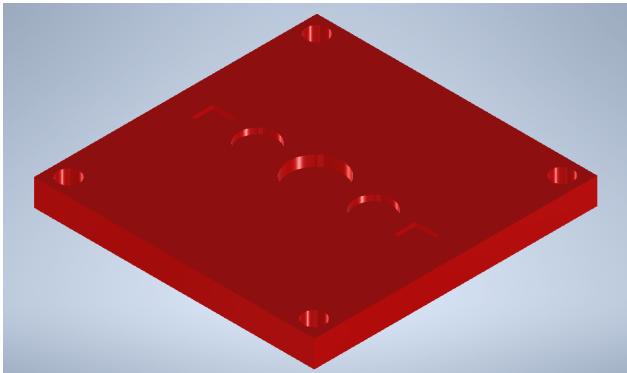


Figura 10: Parte superior do *case*

7 Resultados e Discussões

A Figura 11 mostra a resposta temporal de aceleração e a fft dessas respostas. Para esse experimento foi analisada a resposta de um veículo em marcha lenta. Percebe-se, como o esperado, que há uma oscilação na aceleração do eixo z proveniente da vibração do veículo nessa direção. Na resposta obtida da fft percebe-se picos em torno das frequências de 30 e 50Hz, também respostas vibracionais esperadas de um veículo em marcha lenta.

As Figuras 12 e 13 são as respostas obtidas de um Fiat Uno em dois pontos distintos, sobre a grade dianteira e sobre o bloco da transmissão, respectivamente. Na fft da resposta frontal, nota-se picos em torno da frequência de 20Hz, frequência esperada para esse modelo. Na fft da resposta da transmissão, por sua vez, percebe-se um ganho nas frequências próximas de 6Hz.

A Figura 14 mostra a resposta em marcha lenta de uma moto. Por ser um veículo de menor massa, naturalmente esperaria-se que frequências menores apresentariam um maior valor na fft. Esse comportamento é observado no gráfico, o que, de certa forma, valida o funcionamento correto da placa de aquisição de dados.

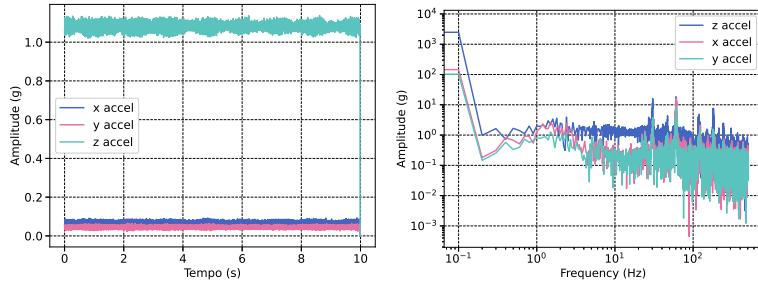


Figura 11: Resposta vibracional - Uno (interno)

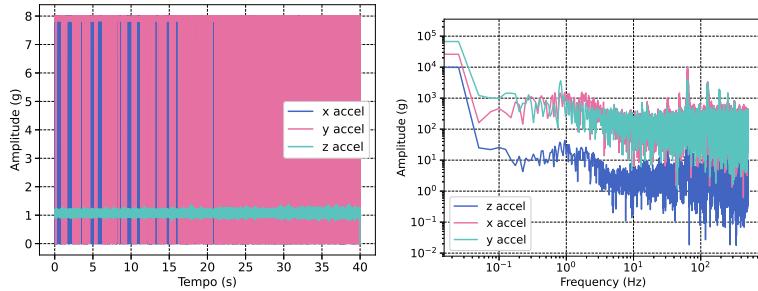


Figura 12: Resposta vibracional - Uno (dianteira)

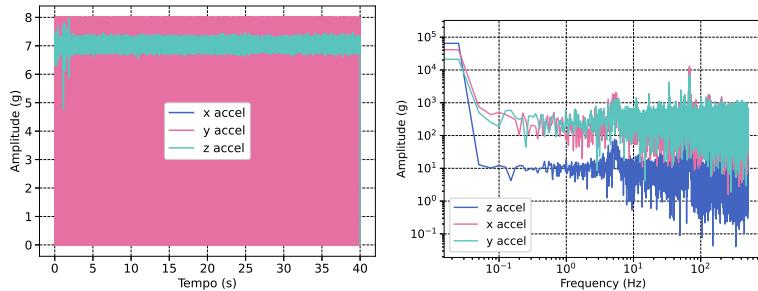


Figura 13: Resposta vibracional - Uno (bloco da transmissão)

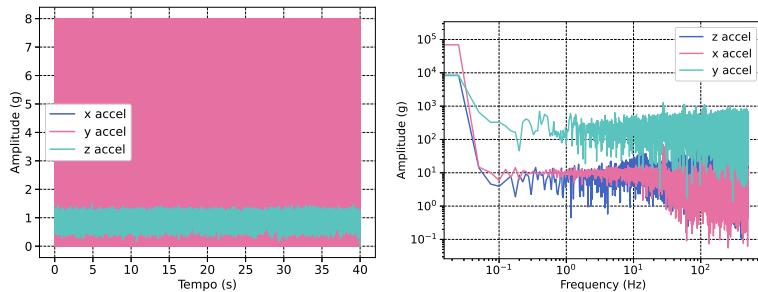


Figura 14: Resposta vibracional - Honda XRE (marcha lenta)

8 Conclusões

De modo geral, o trabalho foi proposto e executado de forma esperada, em conjunto com as disciplinas de Telecomunicações, Dinâmica Veicular e Vibrações. Apesar dos contratempos encontrados, como a memória reduzida do attiny85, o que não possibilitou a utilização

dele como um *buffer* a mais, que seria útil para o propósito. Por outro lado, a transmissão com o ESP foi bem sucedido e os resultados esperados puderam ser observados. Apesar dos ruídos (já esperados), foi possível extrair, através das *FFT*'s, as respostas em frequência previstas, para os modelos experimentados. Uma eventual melhoria seria a substituição dos microcontroladores atuais por um com maior memória, poder de processamento e possibilidade de *multithreading*, isto pois, apesar da excelente taxa de amostragem obtida (1kHz), para o sistema em análise, alguns dados são perdidos durante a transmissão dos dados para o computador; problema que seria solucionado em um processador que pudesse realizar duas tarefas simultaneamente: amostrar um pacote de dados, enquanto outro (previamente obtido) é transmitido via UDP.

Referências

Inman, D. J. (2014). *Engineering Vibrations*. Pearson Education, 4th edition.