

INSTITUTO FEDERAL DO ESPIRITO SANTO
CAMPUS SERRA
ENGENHARIA DE CONTROLE E AUTOMAÇÃO

GABRIEL SAADE PAGANI
RENAN GEORGIO CARVALHO CUZZUOL

RELATÓRIO DO TRABALHO FINAL DE COMUNICAÇÃO DE DADOS

SERRA

2018

Sumário

Introdução.....	3
1. Parte 1: Simulação e análise de camada física.....	3
1.1. Considerando o padrão RS485, faça um levantamento das principais tecnologias/protocolos de automação que utilizam desse padrão. Escolha um dessas tecnologias e apresente sua aplicação direta, e quais os benefícios e limitações que a adoção do RS485 proporciona a ela.	3
1.2. Utilizando o Matlab/Octave, reproduza um padrão de transmissão 01010101 sobre o sinal digital do padrão RS 485, utilizando-se dos exemplos apresentados em sala de aula. Gere o sinal digital por meio de conjunto de senoides.	3
1.2.1. Encontre o padrão de sinal analógico (com senos ou cossenos) capaz de gerar o sinal digital do padrão, conforme a especificação do RS-485. Apresente seu formato no domínio do tempo e a largura de banda necessária para gerá-lo no domínio da frequência. .	3
1.2.2. Considerando o problema de eco, proporcionado pela falta de terminação em redes de barramento, verifique qual é a SNR do sinal de eco capaz de afetar a interpretação correta do sinal pelas estações receptoras. Discuta a relação desse sinal de eco com a distância e número de estações do barramento, apresente o formato de onda para diferentes sinais de eco.	7
1.2.3. Considerando a interferência por ruído provocados por motores, verifique a relação de amplitude e SNR capaz de afetar a interpretação correta do sinal pelas estações receptoras considerando a proximidade de motores de indução (60Hz) e conjuntos de motores com inversores (considere sinais senoidais de 60Hz e seus harmônicos até o 7º). Apresente os formatos de ondas com essas interferências e testes a SNR capazes de provocar erro na leitura das estações receptoras. (gere uma função ruído senoidal com a frequência fundamental e suas armônicas, gradativamente da 1ª até a 7ª com amplitudes decrescentes a 1/3 da anterior). Apresente os gráficos e discuta o SNR para cada caso de ruído. 10	
2. Parte 2: Desenvolvimento de sistema de comunicação real baseado no padrão RS-485 ..	15
3. Parte 3: Integração do sistema com PC via rede Wifi.....	21
4. Referências.....	21

Introdução

O presente relatório explana as atividades desenvolvidas no projeto final da matéria de Comunicação de Dados. O trabalho consiste em 3 etapas: Simulação e análise de camada física, desenvolvimento de sistemas de comunicação real baseado no padrão RS-485, e integração do sistema desenvolvido com PC via rede Wifi.

1. Parte 1: Simulação e análise de camada física

- 1.1. Considerando o padrão RS485, faça um levantamento das principais tecnologias/protocolos de automação que utilizam desse padrão. Escolha um dessas tecnologias e apresente sua aplicação direta, e quais os benefícios e limitações que a adoção do RS485 proporciona a ela.
- 1.2. Utilizando o Matlab/Octave, reproduza um padrão de transmissão 01010101 sobre o sinal digital do padrão RS 485, utilizando-se dos exemplos apresentados em sala de aula. Gere o sinal digital por meio de conjunto de senoides.
 - 1.2.1. Encontre o padrão de sinal analógico (com senos ou cossenos) capaz de gerar o sinal digital do padrão, conforme a especificação do RS-485. Apresente seu formato no domínio do tempo e a largura de banda necessária para gerá-lo no domínio da frequência.

O RS-485 é baseado na transmissão diferencial de dados, onde, a mesma é alcançada por meio da utilização de um par de cabos trançado, tornando com isso a transmissão mais resistente a deterioração por alguma forma de ruído e/ou distorção em fase, tornando assim este um bom método para a transmissão de dados em ambientes onde existe uma presença relevante de interferência eletromagnética. Outras vantagens características do padrão RS-485 estão nas altas taxas de transmissão e as grandes distancias em que conseguem transmitir, ultrapassando 1200m. Diante das características anteriormente supracitadas no padrão RS-485 é altamente utilizado como especificações de meio físico de protocolos como Modbus, ProfBus, BACnet, Componet, EnDat e BiSS. Um dos grandes motivos pelo qual o protocolo Profbus adotar os padrões RS-485 está na não existência do sinal “zero” como referência e sim a adoção do diferencial flutuante, conferindo uma imunidade magnética muito grande, dentre os demais motivos estão: distancia de comunicação, taxa de comunicação, possibilidade de adicionar e retirar dispositivos da rede sem influenciar dispositivos da rede que já estão em operação, transmissão assíncrona NRZ e até 32 estações podem ser conectadas diretamente. Contudo dentro das limitações podemos citar a necessidades de terminações bem especificas para os barramentos e a não existência de um canal full-duplex.

O sinal digital do padrão RS-485 é uma onda quadrada com amplitude de $\pm 6V$. Sabe-se que qualquer sinal periódico pode ser formado através do somatório de senoídes de frequências diferentes. Sendo assim, para gerar uma onda quadrada com as características do sinal padrão RS-485, foram testados somatórios de diferentes números de harmônicos de senoides com frequencia fundamental igual a 2kHz, e amplitude igual a 3V para evitar que os picos nas transições de nível logico alto para baixo, e baixo para alto excedessem os limites de $\pm 6V$. Os sinais gerados (no dominio do tempo) são mostrados nas figuras 1, 3 e 5, e seus respectivos espectros de frequência são mostrados nas figuras 2, 4 e 6.

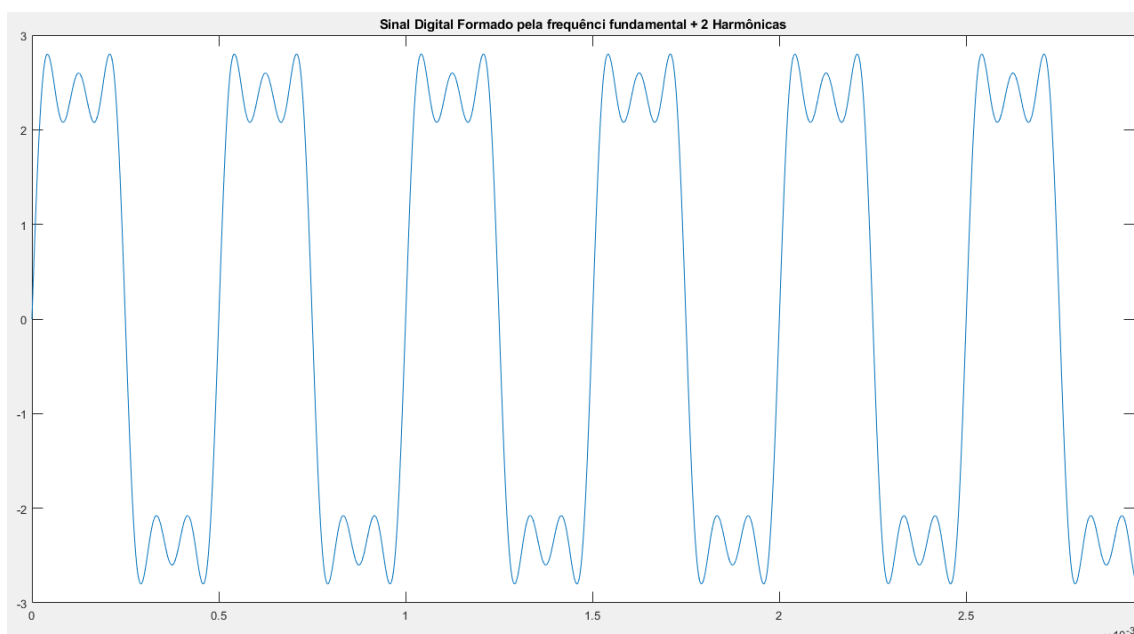


Figura 1: Sinal digital padrão RS-485 formado pelo somatório das 3 primeiras harmonicas de um sinal senoidal de 2kHz e 3V de amplitude

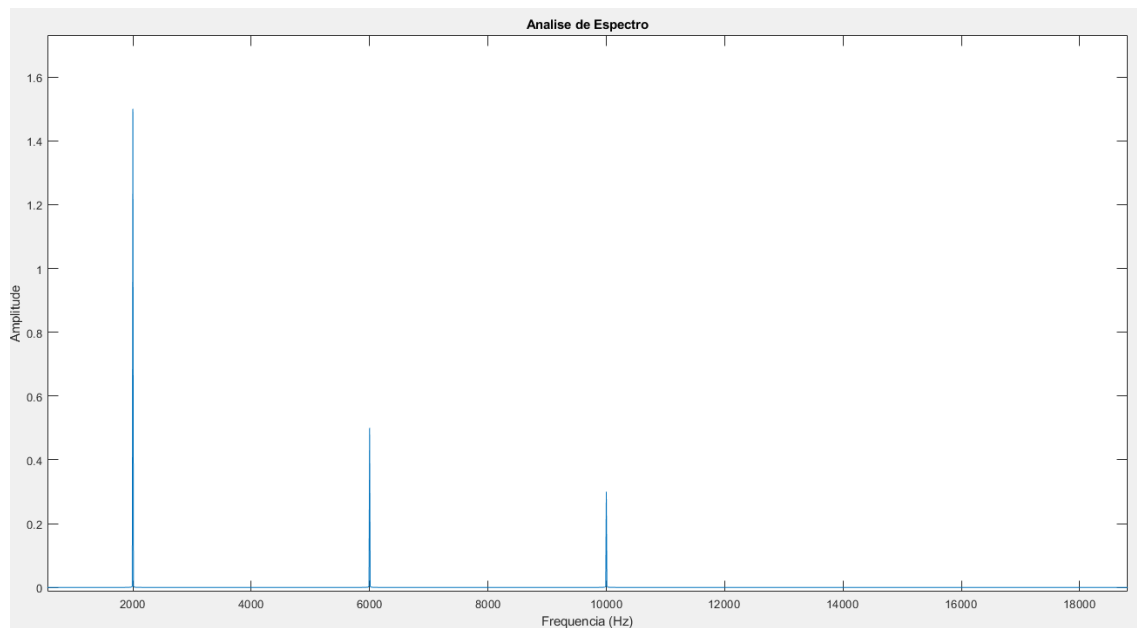


Figura 2: Análise de Espectro sinal digital padrão RS-485 formado pelo somatório das 3 primeiras harmonicas de um sinal senoidal de 2kHz e 3V de amplitude

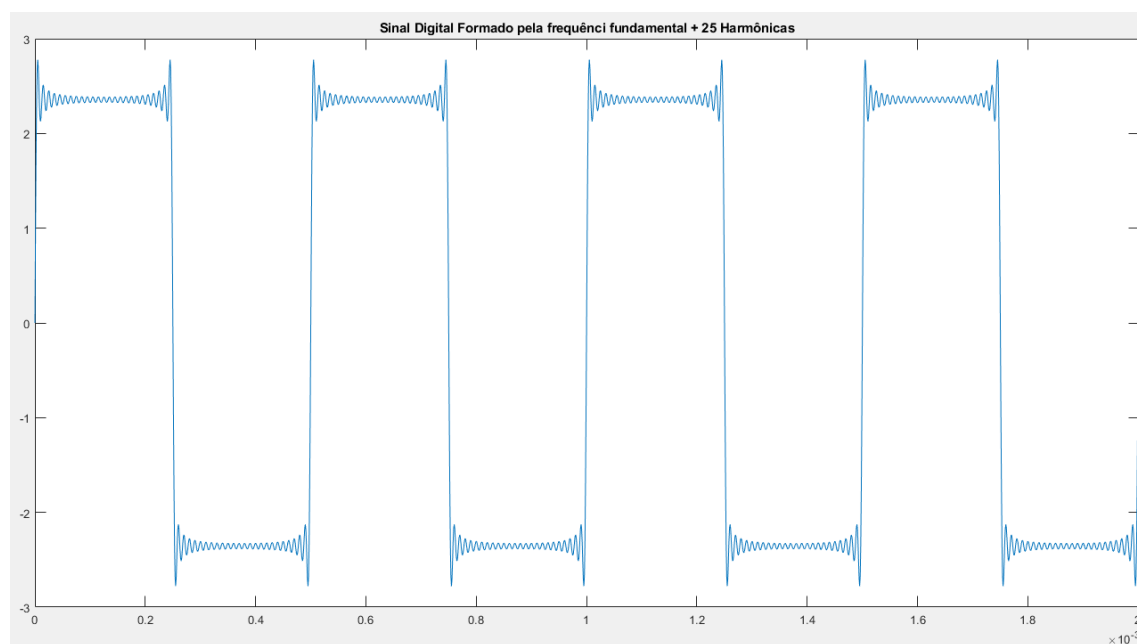


Figura 3: Sinal digital padrão RS-485 formado pelo somatório das 50 primeiras harmonicas de um sinal senoidal de 2kHz e 3V de amplitude

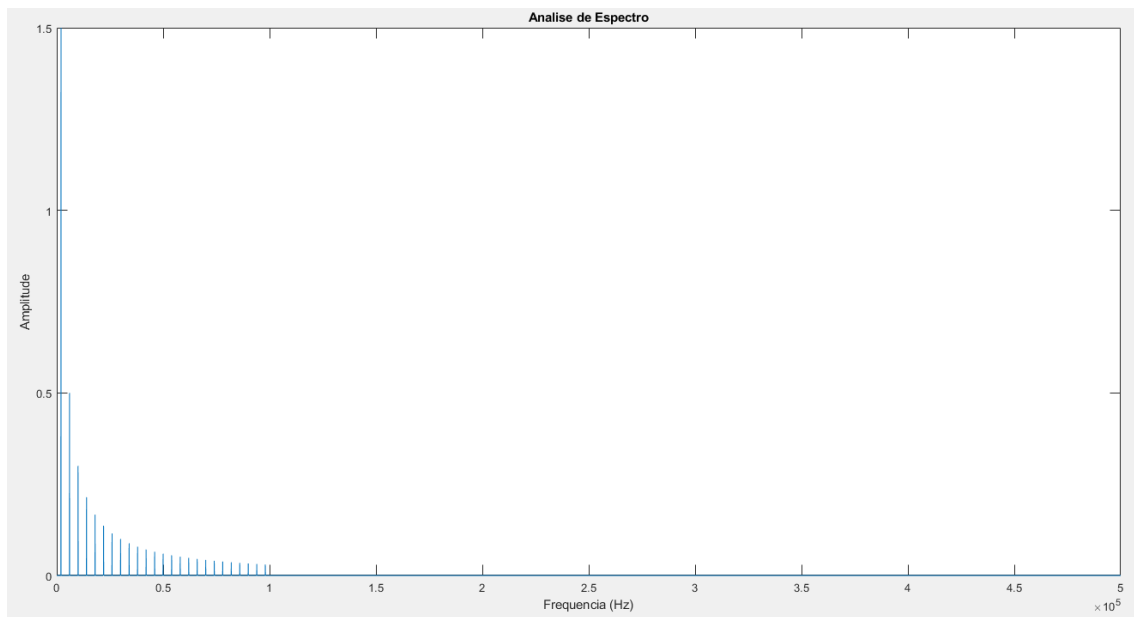


Figura 4: Análise de Espectro sinal digital padrão RS-485 formado pelo somatório das 50 primeiras harmonicas de um sinal senoidal de 2kHz e 3V de amplitude

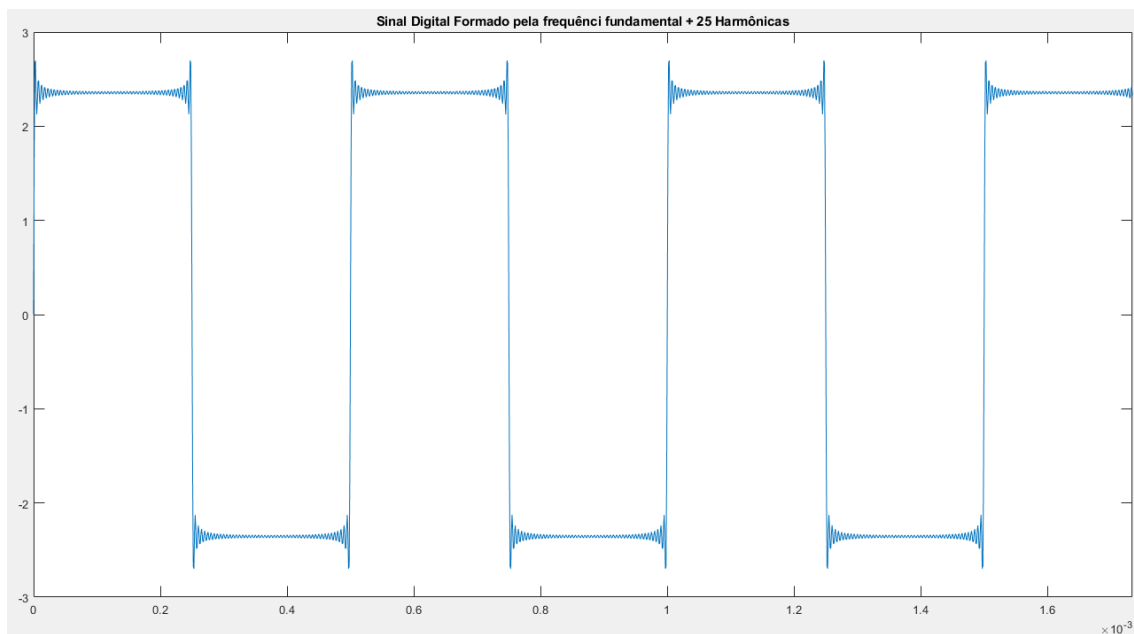


Figura 5: Sinal digital padrão RS-485 formado pelo somatório das 100 primeiras harmonicas de um sinal senoidal de 2kHz e 3V de amplitude

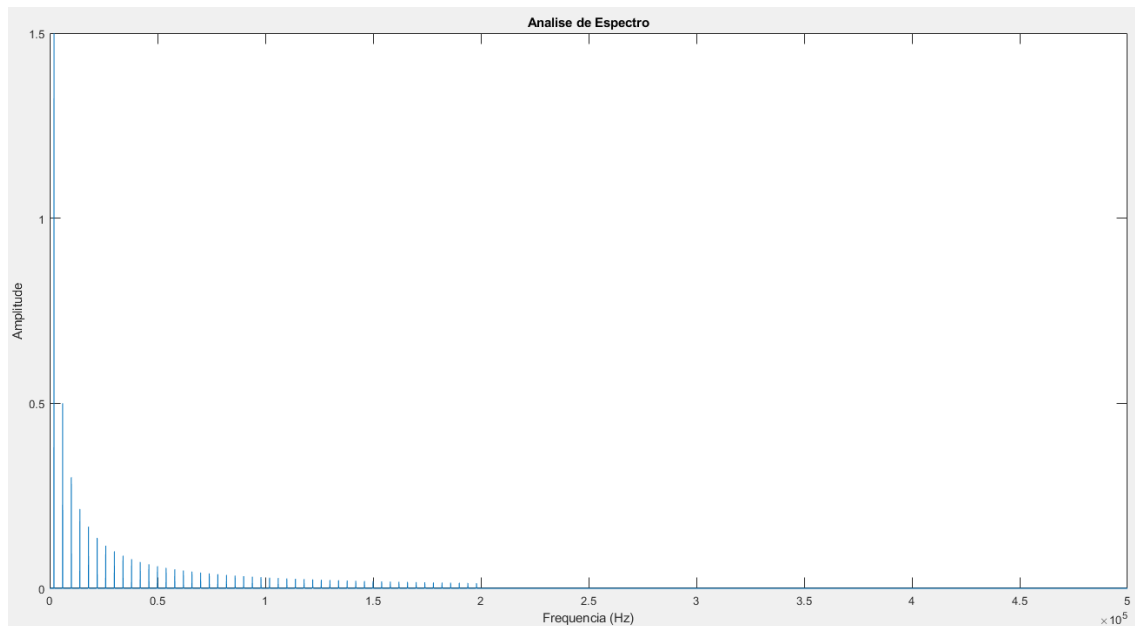


Figura 6: Análise de Espectro sinal digital padrão RS-485 formado pelo somatório das 100 primeiras harmônicas de um sinal senoidal de 2kHz e 3V de amplitude

Conforme a figura 1, observa-se que o sinal formado pelo somatório dos três primeiros harmônicos é suficientemente parecido com o sinal digital do padrão RS-485. Porém, a medida que adicionamos mais harmônicas, o sinal fica cada vez mais quadrado, ou seja, mais preciso e mais resistente a eventuais interferências e ruídos, conforme veremos no item 1.2.2.

1.2.2. Considerando o problema de eco, proporcionado pela falta de terminação em redes de barramento, verifique qual é a SNR do sinal de eco capaz de afetar a interpretação correta do sinal pelas estações receptoras. Discuta a relação desse sinal de eco com a distância e número de estações do barramento, apresente o formato de onda para diferentes sinais de eco.

Segundo a especificação do RS-485, para que um receptor seja capaz de “entender” um sinal digital nesse padrão de comunicação, o sinal recebido deve chegar ao receptor com uma amplitude de 0,2V à 6V (nível lógico ALTO), ou -0,2V à -6V, conforme a figura 7.

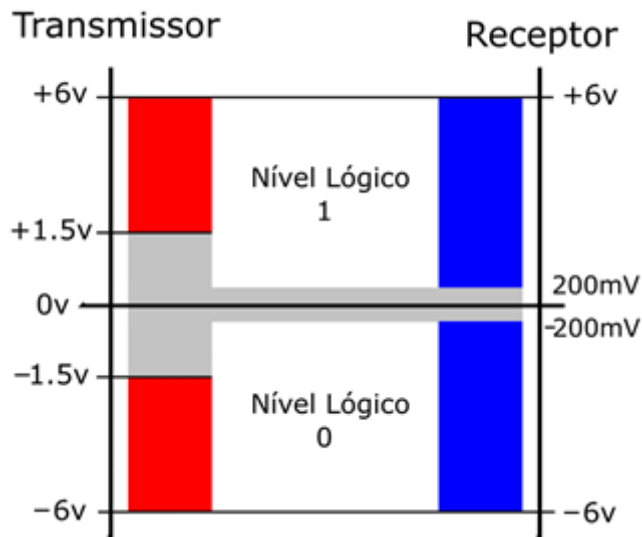


Figura 7: Padrão de comunicação do RS-485

Foi então gerado um sinal de eco, que nada mais é do que o próprio sinal com amplitude reduzida e com uma defasagem adicionada, conforme figura 8. Foi escolhida arbitrariamente uma amplitude do sinal de eco de $\frac{1}{3}$ da amplitude do sinal original e defasagem de $\pi/2$, e o resultado da interferência do sinal original com o sinal de eco é apresentada na figura 9. A figura 10 nos mostra a interação para o caso de um sinal de eco com amplitude de $\frac{1}{2}$ da amplitude do sinal original, ou seja, 1.5V.

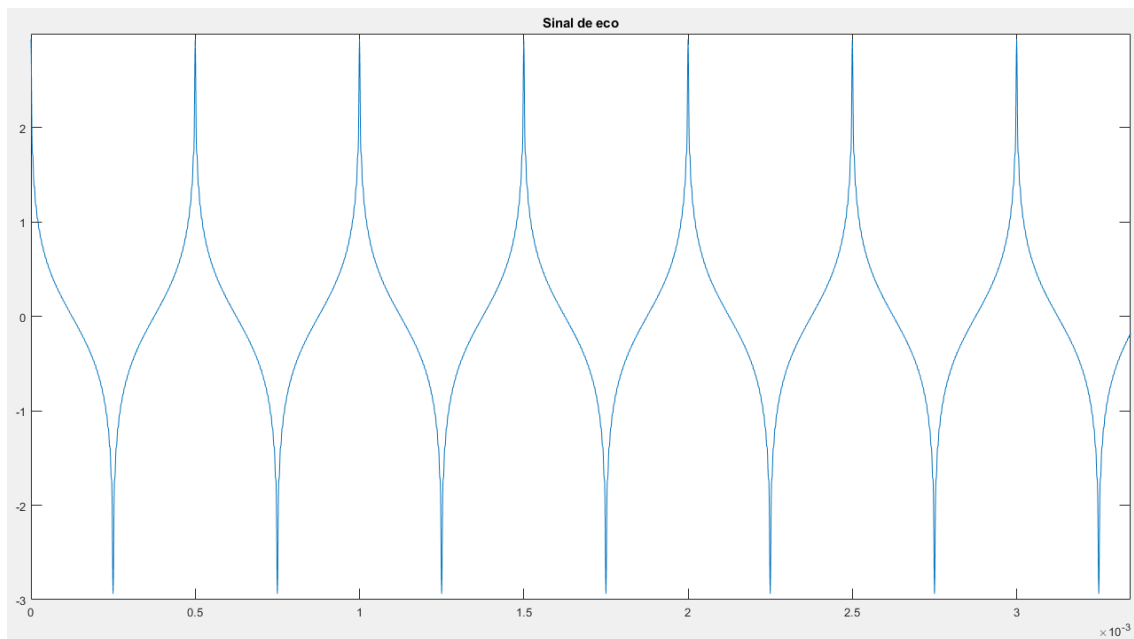


Figura 8: Ruído com defasagem de $\pi/2$ e amplitude de 1V

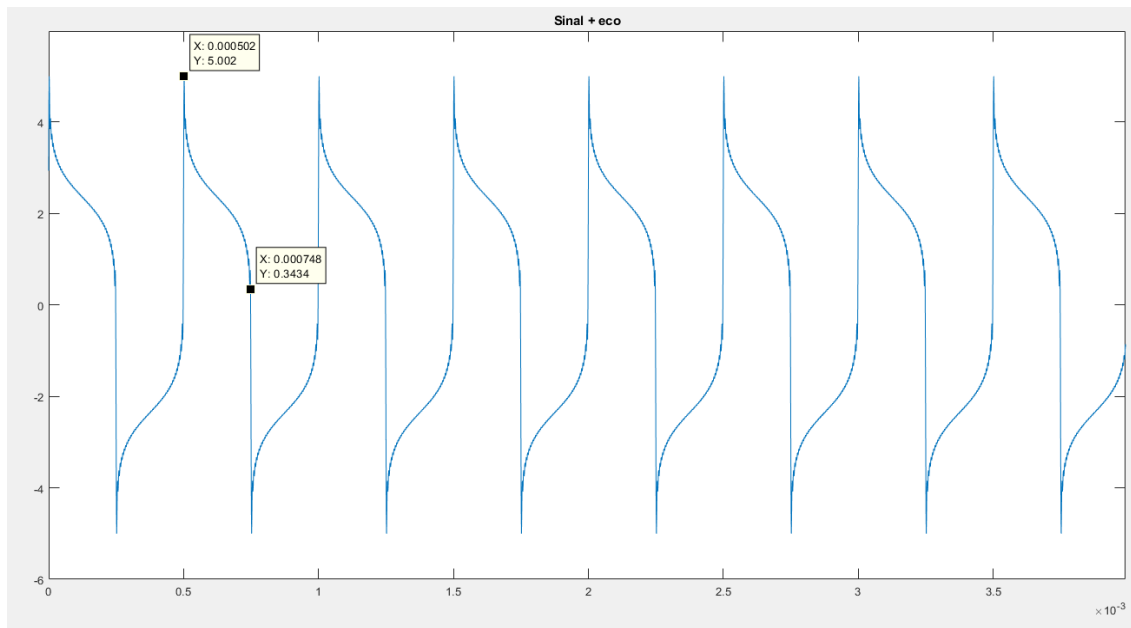


Figura 9: Sinal amplitude 3V + eco de amplitude 1V

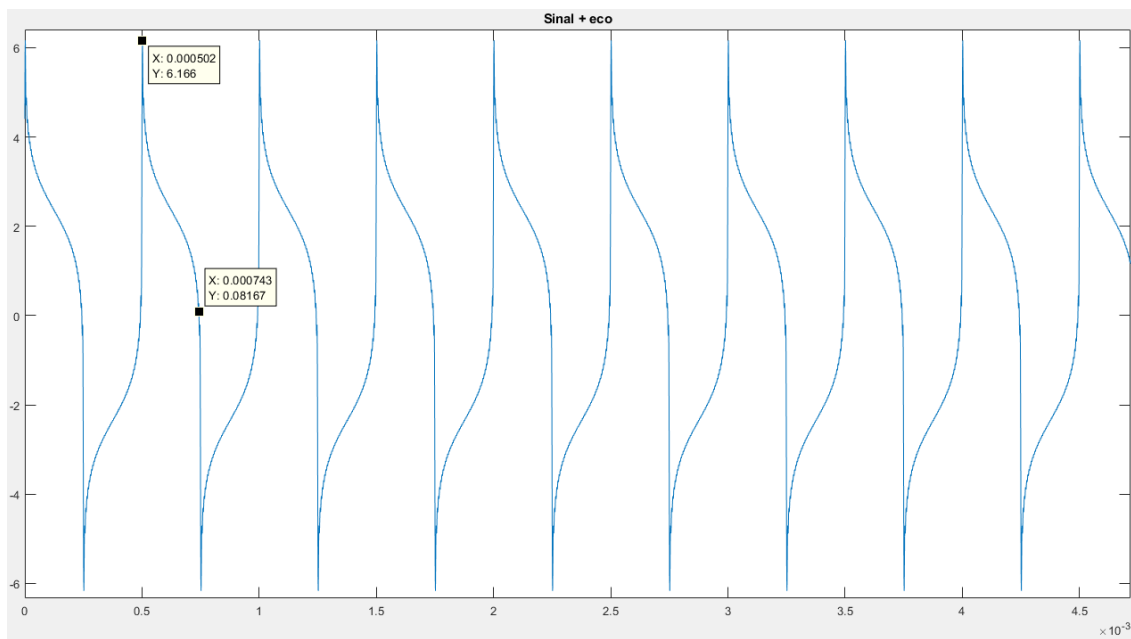


Figura 10: Sinal de amplitude 3V + eco de amplitude 1.5V

A figura 9 nos mostra que apesar de bastante distorcido, o sinal ainda poderia ser “compreendido” por um receptor que usa o padrão RS-485, pois os níveis logicos ainda estão na faixa de +/- 0.2V à +/- 6V. Porém, conforme explicitado na figura 10, um sinal de eco com amplitude igual à 1.5V já tem potencial para causar uma leitura incorreta no receptor, e portanto comprometer a comunicação.

1.2.3. Considerando a interferência por ruído provocados por motores, verifique a relação de amplitude e SNR capaz de afetar a interpretação correta do sinal pelas estações receptoras considerando a proximidade de motores de indução (60Hz) e conjuntos de motores com inversores (considere sinais senoidais de 60Hz e seus harmônicos até o 7º). Apresente os formatos de ondas com essas interferências e testes a SNR capazes de provocar erro na leitura das estações receptoras. (gere uma função ruído senoidal com a frequência fundamental e suas armônicas, gradativamente da 1ª até a 7ª com amplitudes decrescentes a 1/3 da anterior). Apresente os gráficos e discuta o SNR para cada caso de ruído.

Para simular o efeito da interferência causada por um motor de 60Hz, foi gerado um sinal com essa frequência e 1V de amplitude, conforme figura 11.

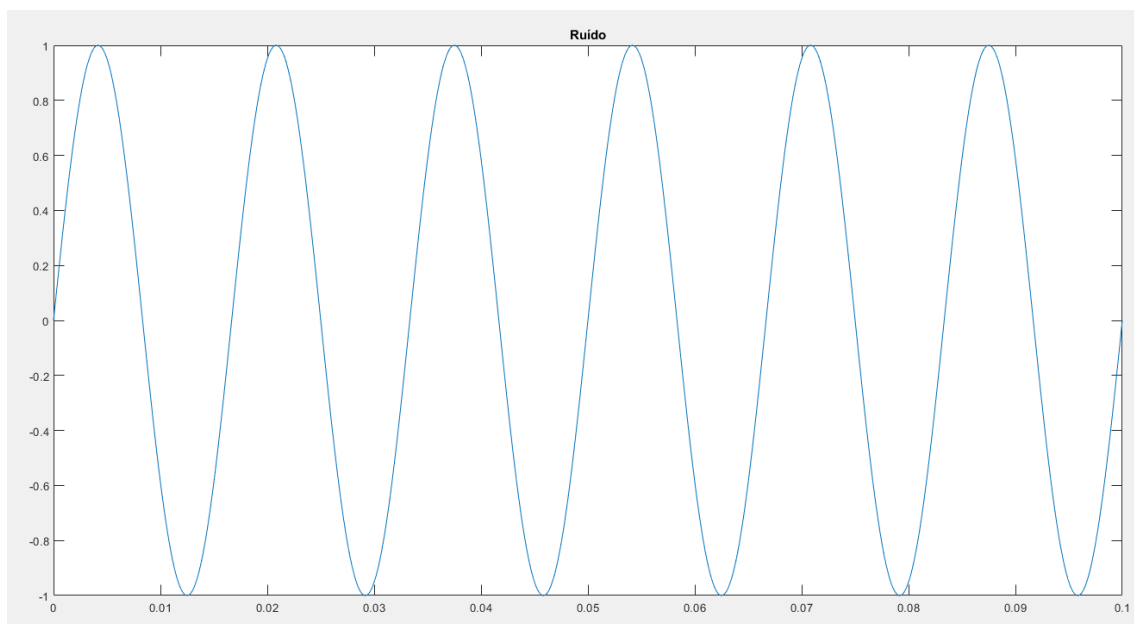


Figura 11: Sinal de ruído (motor), amplitude 1V e frequência 60Hz

O ruído apresentado na figura 11 foi então somado ao sinal principal gerado anteriormente no item 1.2.1, e o resultado é mostrado na figura 12.

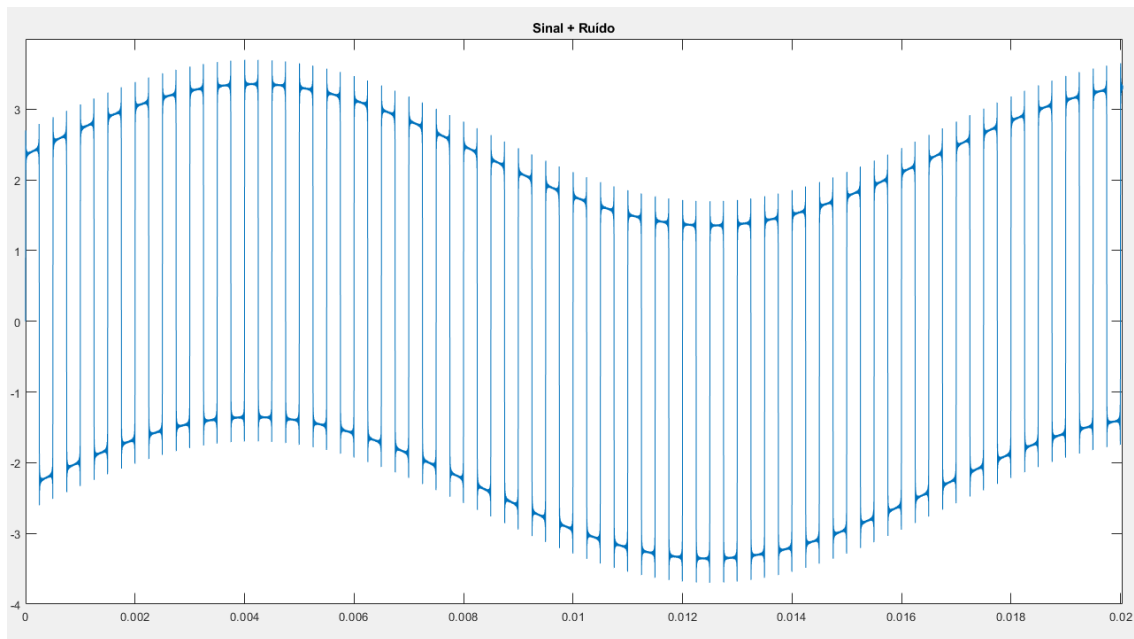


Figura 12: Sinal + ruído de 1V com frequência 60Hz

Observa-se que a comunicação não foi comprometida, pois os níveis lógicos ainda podem ser identificados. Adicionamos gradativamente mais harmônicas ao somatório usado para gerar o sinal do ruído, e somamos o ruído ao sinal gerado no item 1.2.1, e os resultados são mostrados nas figuras a seguir:

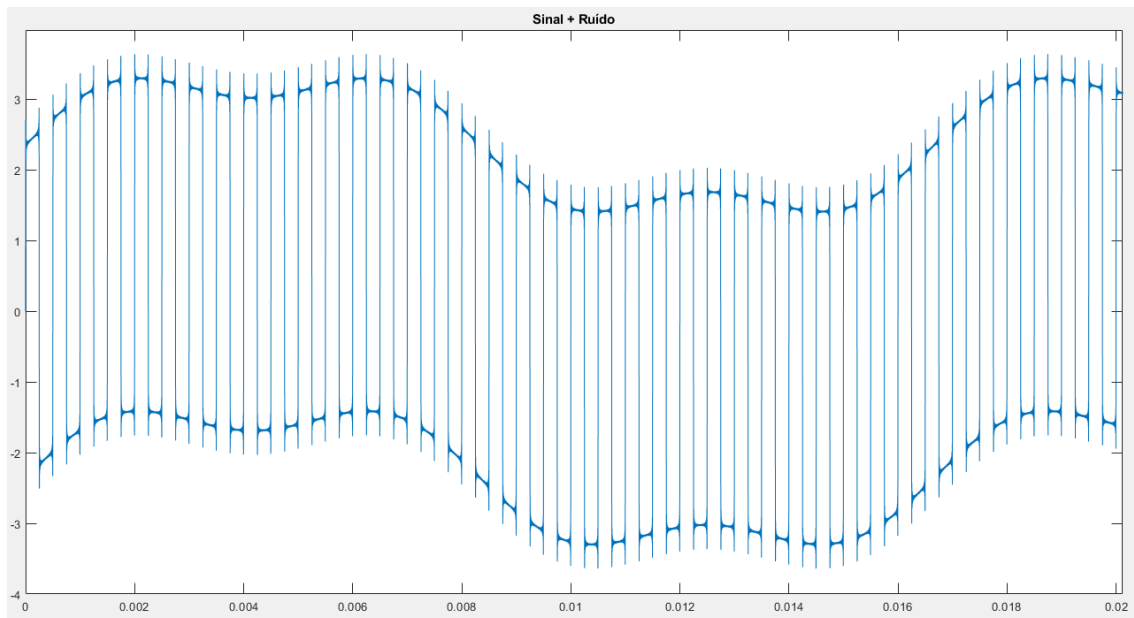


Figura 13: Sinal + ruído 1V com 3 harmônicas

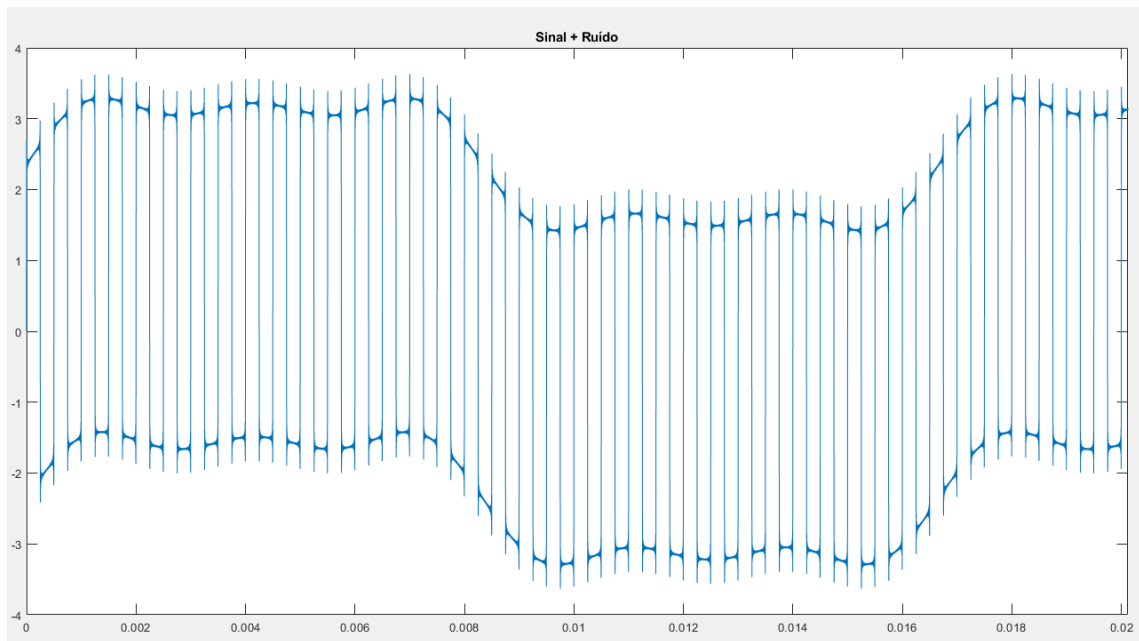


Figura 14: Sinal + ruído 1V com 5 harmônicas

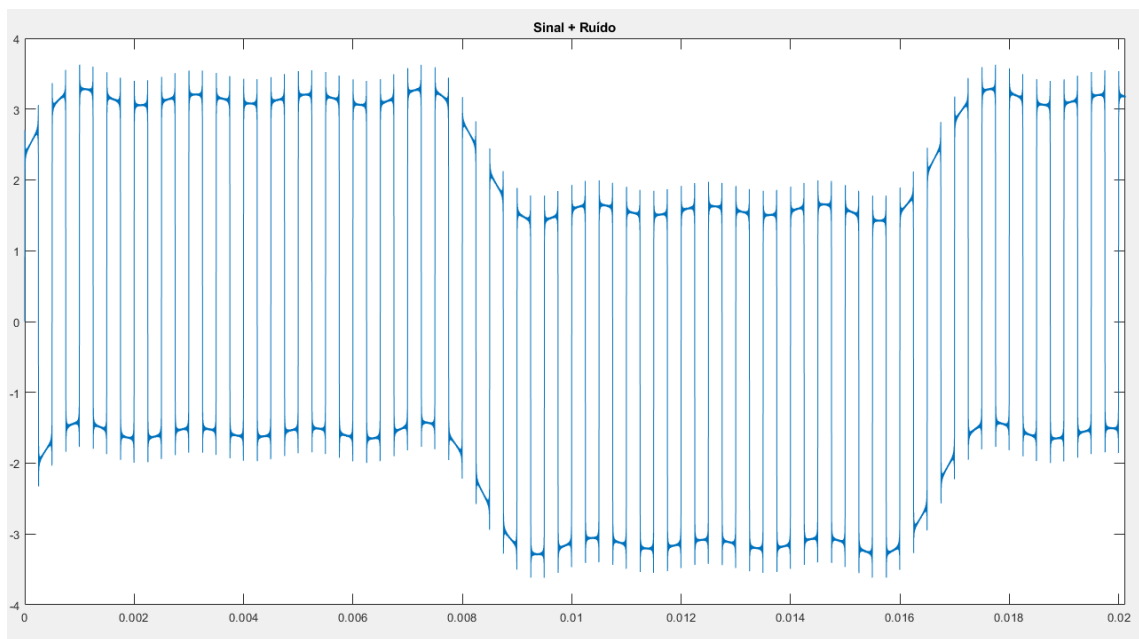


Figura 15: Sinal + ruído 1V com 7 harmônicas

Observa-se que o ruído de 1V (e, portanto, SNR de 3) não foi capaz de causar distorção suficiente para comprometer a identificação dos níveis lógicos, e portanto a comunicação não foi comprometida. Aumentamos então a amplitude do sinal de ruído para 2V, e refizemos os testes:

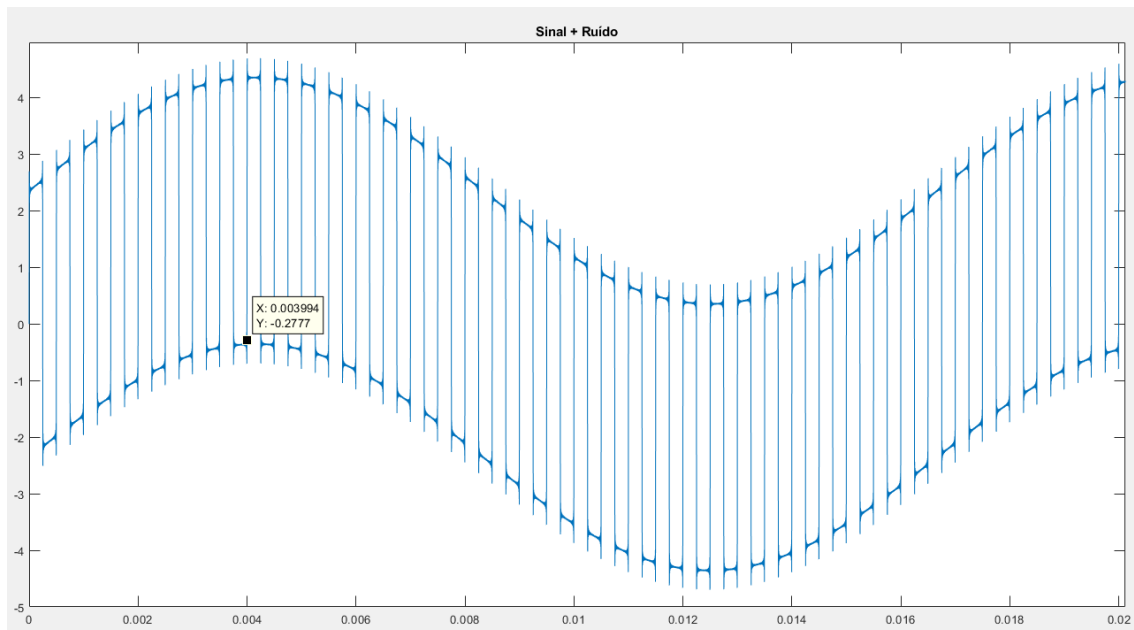


Figura 16: Sinal + ruído 2V com 1 harmonica

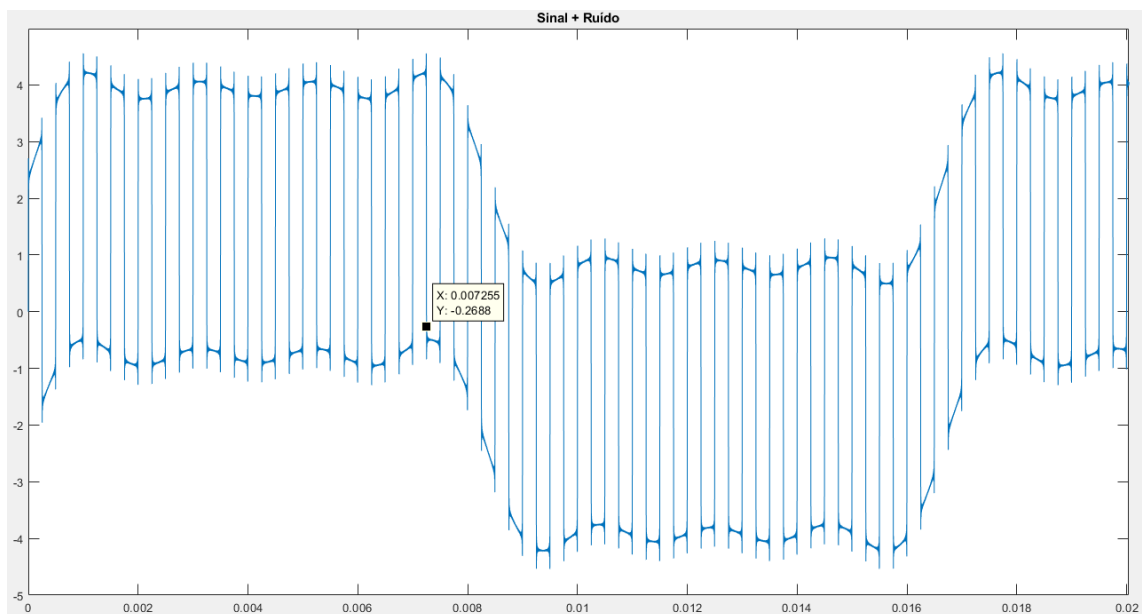


Figura 17: Sinal + ruído 2V com 7 harmonicas

Novamente, a distorção, dessa vez com SNR de 1.5, não foi suficiente para comprometer a comunicação, porém por muito pouco. Aumentamos então a amplitude do sinal de ruído para 2.1V, e refizemos os testes:

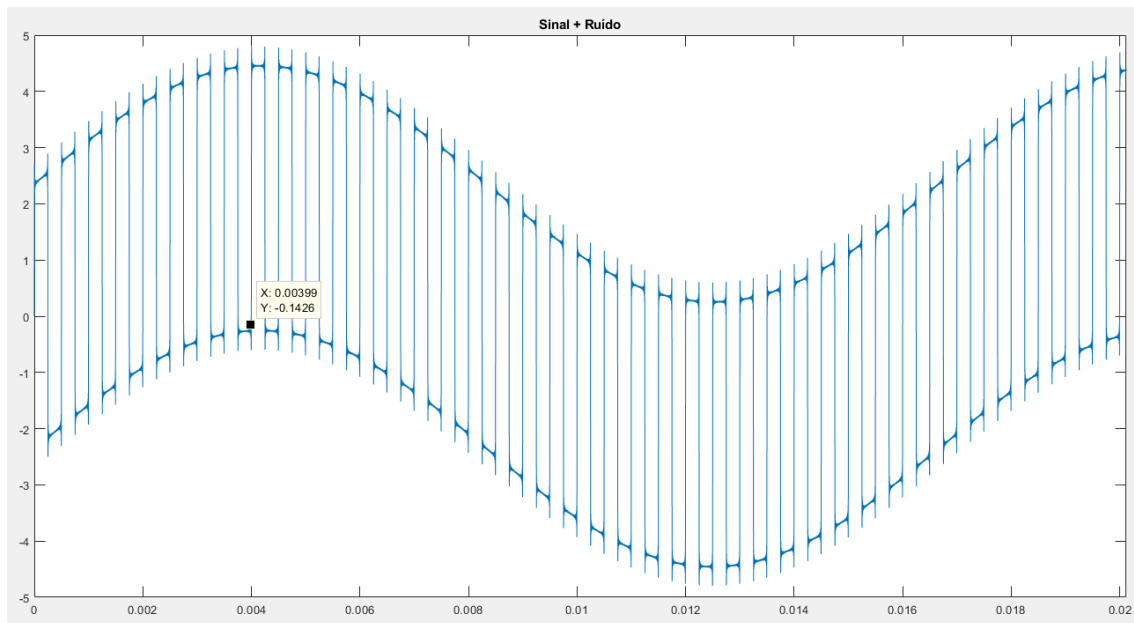


Figura 18: Sinal + ruído 2.1V com 1 harmônica

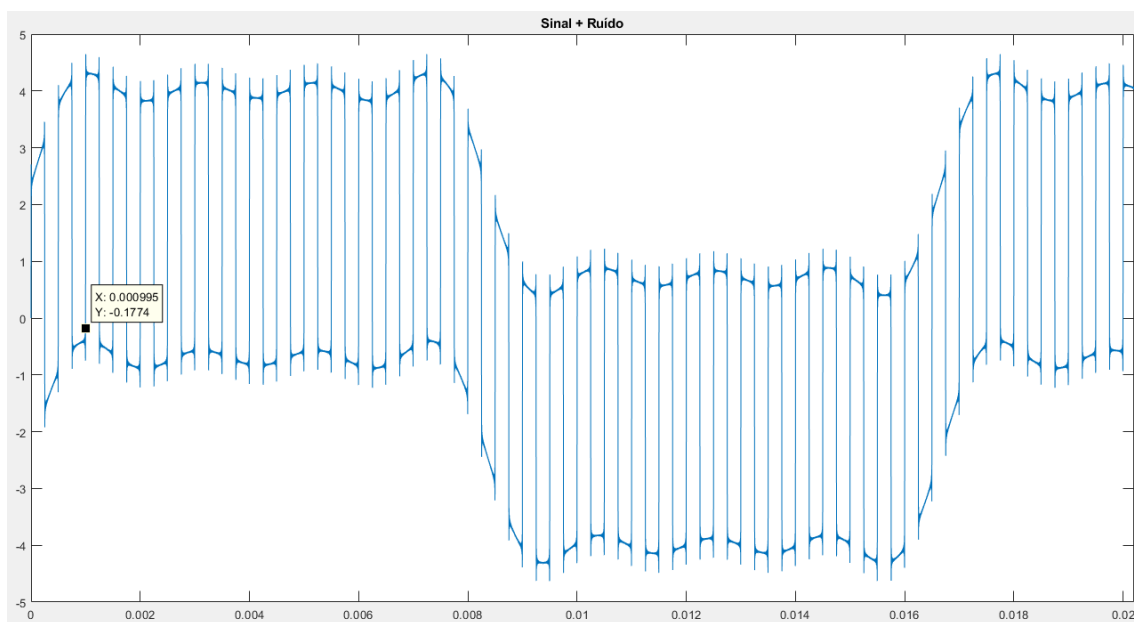


Figura 19: Sinal + ruído 2.1V com 7 harmônicas

A partir das figuras 18 e 19, conclui-se que um sinal de ruído de 60Hz é capaz de produzir interferência que comprometa a comunicação no nosso sinal de 2kHz para SNR menores que 1.5, apesar de que para valores de SNR próximos de 1.5 a comunicação ainda possa acontecer, pois os níveis lógicos só ultrapassam os valores para os quais os dispositivos podem reconhecê-los nas extremidades das transições. Para ilustrar um caso de interferência que mais provavelmente comprometeria a comunicação, aumentamos a amplitude do sinal de ruído para 3V, e o resultado é mostrado na figura 20.

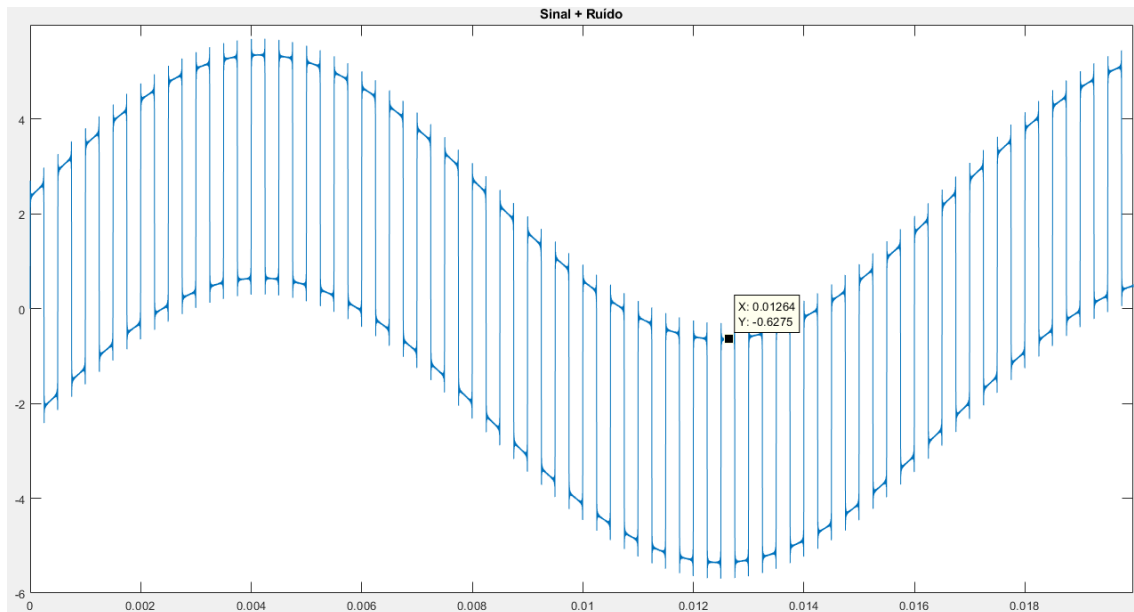


Figura 20: Sinal + ruído de 3V

Está explicitado na figura um valor de tensão que deveria corresponder ao nível lógico ALTO, porém sua tensão está bem abaixo do limite para o qual o dispositivo a identificaria como ALTO, e portanto o receptor não seria capaz de interpretar esse sinal corretamente. Concluímos então que um SNR menor que 1.5 começa a ser problemático para a comunicação, e deve ser evitado.

Parte 2: Desenvolvimento de sistema de comunicação real baseado no padrão RS-485

2.1 - Memorial

Conforme apresentado no dia 11/12, foi desenvolvido um sistema de comunicação usando um NodeMCU, que exercia o papel de mestre, e dois arduinos (escravos 1 e 2). Para transformar o sinal dos microcontroladores para padrão RS-485, foram usados conversores TTL – RS485 para cada dispositivo (mestre e escravos). Cada escravo possuía um atuador (LED embutido do Arduino) e um elemento sensor (potenciometro). Devido ao fato da tensão de operação do Mestre(NodeMCU) ser de 3.3V ao passo que todos os outros elementos operam com 5V, foi utilizado com conversor de tensão bidirecional. A comunicação se dava através de um barramento, que conectava todos os dispositivos. O Mestre controlava a comunicação: enquanto todos os escravos começavam no modo “escuta” após o start-up, o Mestre começava no modo “escrita”, e a comunicação se iniciava a partir de uma requisição de dados vinda do Mestre. Abaixo está um breve descritivo dos componentes utilizados:

- NodeMCU 0.9 V3: “é uma plataforma open source com a possibilidade de utilizar a linguagem de script Lua, uma linguagem de múltiplos paradigmas, oferecendo um pequeno conjunto de características gerais que podem ser estendidas para encaixar diferentes tipos de problemas, em vez de fornecer uma especificação

mais complexa e rígida para combinar com um único paradigma, essa linguagem permite que programadores quando implementam nomes, classes, e outras funções, empreguem poderosas técnicas de programação funcional e completos escopos lexicais. O NodeMCU contém o módulo ESP8266 para conexão via Wi-Fi, usa uma interface usb-serial e um regulador de tensão 3.3V, utilizando a comunicação via cabo micro-usb e conta com 11 pinos de I/O e um conversor analógico-digital. “



Figura 21 – NodeMCU 0.9 V3

- Arduino(Uno e Mega):” O Arduino é um hardware open source muito utilizado e sua escolha foi motivada pela familiaridade dos autores deste trabalho com o manuseio e manipulação deste equipamento. Utilizou-se a comunicação serial (UART TTL) na plataforma Arduino para realizar a comunicação entre a placa e o NODEMCU (mestre). Foi através da Entrada e Saída destacadas na imagem abaixo que foi realizada comunicação. Estes pinos estão conectados ao microcontrolador da placa - ATMEGA328 no Arduino Uno e ATMEGA 2560 no Arduino 2560”.



Figura 22 – Arduino Uno.



Figura 23 – Arduino Mega.

- Módulo MAX485 TTL para RS-485: “A função deste componente é converter o sinal em TTL para o padrão de comunicação utilizado neste trabalho: o RS 485. Este módulo, composto do chip MAX 485, é responsável por tal conversão”

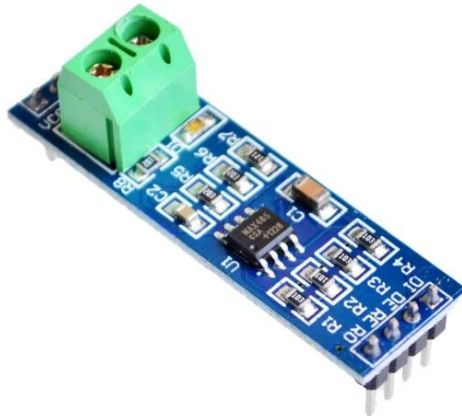


Figura 24 - Módulo MAX485 TTL para RS-485

- Conversor de nível lógico bidirecional 4 canais – 3.3/5v: “O Conversor de Nível Lógico é um pequeno módulo capaz de converter sinais de 5V para 3.3V e isso de forma totalmente segura, possuindo a capacidade de trabalhar com tensões menores, entre 2.8V e 1.8V. Este componente foi necessário para permitir a comunicação do NODEMCU com o módulo MAX 485. O NODEMCU opera na faixa de 3.3V, já o Arduino e o módulo MAX 485 atuam na faixa de 5V (TTL)”.

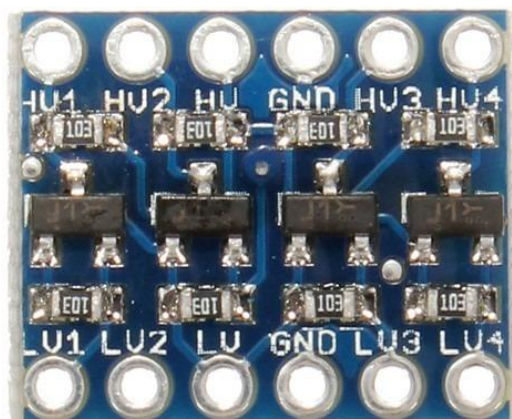


Figura 25 - Conversor de nível logico bidirecional 4 canais – 3.3/5v.

2.2 - Tabela de preços

Item	Quantidade	Valor unitário	Valor Total
Placa de fenolite 0,5L	3	10,00	30,00
NodeMCU wifi ESP8266	1	48,89	48,89
Arduino UNO	1	39,59	39,59
Arduino Mega	1	120,00	120,00
Furador de placa	1	54,77	54,77
Ferro de solda	1	19,90	19,90
Estanho 50g	1	19,99	19,99
Led	2	0,40	0,80
Potenciômetro linear AA 1K	2	1,50	3,00
Resistores	2	0,30	0,60
Conversor TTL	3	13,50	40,50
Conversor de tensão bidirecional	1	9,50	9,50
Sockets	10	1,50	15,00
Perclorato de ferro	1	10,00	10,00
Amônia	2	2,30	4,60
Papel Glossy 20 folhas	1	7,05	7,05
Caneta permanente	3	3,90	11,70

Gerando um valor total em materiais de R\$ 435,89 reais.

2.3 - Layout das placas:

Foram projetadas placas de circuito impresso para o Mestre e para os escravos, e o layout das placas é apresentado nas figuras 26 e 27.

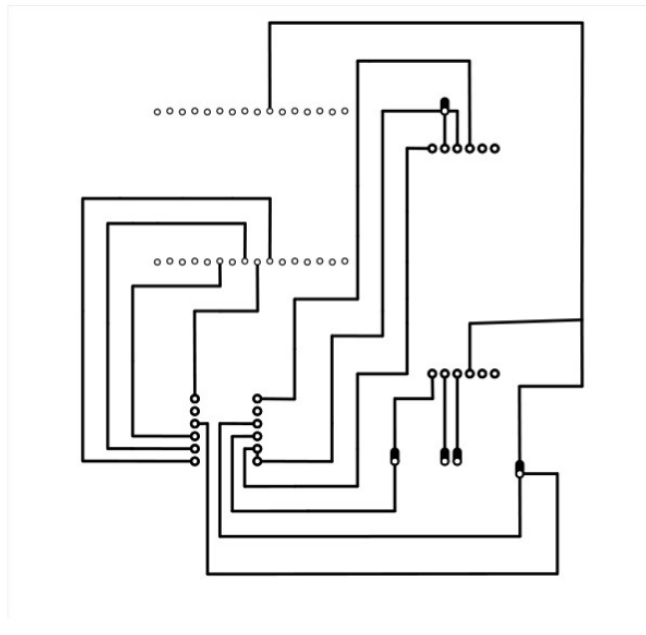


Figura 26 – Layout mestre

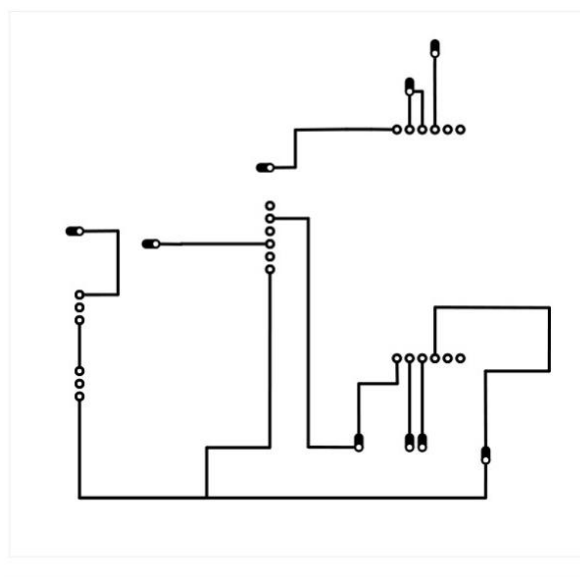


Figura 27 – Layout escravos 1 e 2.

E por fim após as placas projetadas, as mesmas foram confeccionadas seguindo os seguintes passos:

1. Folha Glossy foi resfriada.
2. O layout foi impresso na folha glossy
3. A placa de fenolite foi limpa

4. Com auxílio de um ferro de passar roupa foi executado o processo de “transfer” da folha glossy para a placa.
5. As linhas depositadas na placa de fenolite foram reforçadas com caneta permanente.
6. As placas desenhadas foram depositadas na solução de perclorato.
7. Após removido do perclorato, as placas foram limpas com amônia e em seguida sabão.
8. Os elementos foram soldados as placas

E o resultado pode ser observado na figura abaixo:

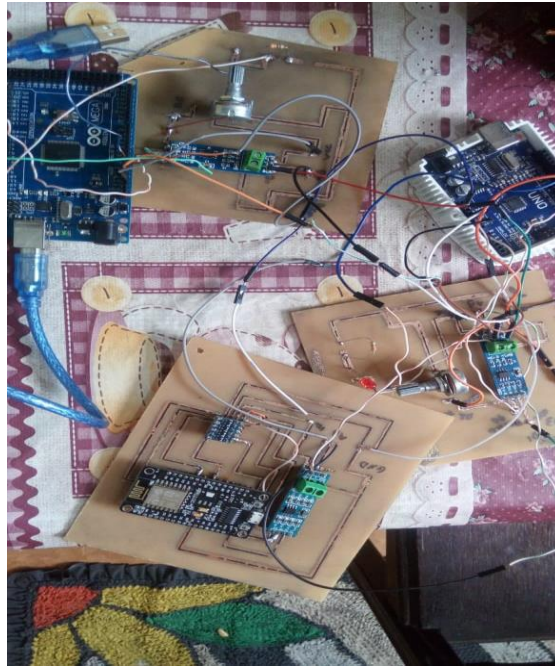


Figura 28 – Placas pcb em fenolite.

2.4 - Comunicação mestre-escravo:

Foram definidas mensagens válidas para nossa comunicação: o Mestre poderia mandar 6 comandos para o barramento:

1. Acionamento do atuador do escravo 1
2. Requerimento de leitura do sensor do escravo 1
3. Acionamento do atuador do escravo 2
4. Requerimento de leitura do sensor do escravo 2
5. Determinação do SP de alarme do escravo 1
6. Determinação do SP de alarme do escravo 2

Onde o SP de alarme determinava o valor limite para a variável sensoriada pelo escravo, e caso esse valor fosse ultrapassado, um alarme era gerado pelo mesmo.

Para que a comunicação funcionasse, era preciso garantir que quem devesse escrever estivesse em estado de escrita, e quem devesse escutar estivesse em modo de escuta. Para isso, foi padronizado que o Mestre sempre iniciaria a comunicação em modo escrita, e

comutaria para modo de leitura somente após enviar o comando de requerimento de leitura do sensor do escravo 1 ou 2. Enquanto isso, os escravos estariam sempre em modo leitura, e somente trocariam para modo escrita caso identificassem o comando de requerimento de leitura do sensor vindo do mestre, e se esse comando fosse de fato endereçado para ele (ou seja, o escravo 2 continuaria em modo leitura caso a mensagem 2 chegasse à ele através do barramento, e só mudaria se identificasse o comando 4).

Parte 3: Integração do sistema com PC via rede Wifi

A integração do sistema com o PC via rede Wifi para realização do controle da rede por meio de uma interface Web, foi realizado de forma a disponibilizar um controle por meio de supervisão do sistema de sensoriamento com comunicação mestre-escravo previamente implementado. Esta etapa foi realizada com o auxílio das bibliotecas: ESP8266WiFi.h e ESP8266HTTPClient.h. Onde foi definido um servidor local, onde foi possível fazer requisição no modo cliente ao sistema projetado. Sendo então possível enviar à um servidor Web os dados de temperatura e estado dos Leds nos dois escravos e ainda o valor de setpoint definido para acionamento do alarme do sistema para os dois sensores de maneira distinta..