



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de falhas no fornecimento de energia

Filipe Teixeira Cardoso

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Orientador
Prof. Dr. Daniel Chaves Café

Brasília
2019



Universidade de Brasília

Instituto de Ciências Exatas
Departamento de Ciência da Computação

Detecção de falhas no fornecimento de energia

Filipe Teixeira Cardoso

Monografia apresentada como requisito parcial
para conclusão do Curso de Engenharia da Computação

Prof. Dr. Daniel Chaves Café (Orientador)
ENE/FT/UnB

Prof. Dr. Ricardo Zelenovsky Prof. Dr. João José Costa Gondim
Universidade de Brasília Universidade de Brasília

Prof. Dr. José Edil Guimarães de Medeiros
Coordenador do Curso de Engenharia da Computação

Brasília, 09 de julho de 2019

Dedicatória

À minha mãe, cuja força e perseverança tanto me inspiram. Ao meu pai, cujo apoio incondicional sempre me amparou em momentos de dificuldade. Ao meu irmão mais velho, com o qual eu tanto aprendo. Ao meu irmão mais novo, quem eu mais admiro. À ciência e todos os seus gigantes, que um dia possa estar sobre seus ombros.

Agradecimentos

Agradeço primeiramente à minha família e principalmente aos meus pais, minha mãe Ivone e meu pai Léo, por me proporcionarem ao longo de minha vida todas as condições para estudar e alcançar esta conquista. Agradeço à todos os meus professores e professoras, desde o ensino básico até este momento, ao fim da graduação - trabalhadores da mais nobre das profissões. Dentre estes, agradeço sobretudo à meu orientador, o Prof. Dr. Daniel Chaves Café, pela instrução, incentivo, paciência e compreensão. Obrigado a meu colaboardor neste projeto e amigo, Marcos Vinicius Prescendo Tonin, com quem compartilhei as dificuldades e superações. Também aos meus colegas de curso, que me acompanharam nessa longa jornada por disciplinas, trabalhos, pesquisas, experimentos e provas. Aos meus amigos, pela camaradagem e apoio emocional. E por fim, à Universidade de Brasília e os respectivos departamentos, particularmente o Departamento de Engenharia Elétrica e o Departamento de Ciência da Computação, por proporcionarem a estrutura necessária para a minha formação e a de inúmeros outros cientistas e engenheiros.

Resumo

O trabalho aqui apresentado aborda a implementação de um sistema completo para monitoramento local e remoto - por meio do envio de dados, informações e alertas pela internet - da tensão na malha elétrica de uma sala ou edifício, de parâmetros derivados dessa tensão e, em menor foco, da corrente que alimenta um computador/servidor ligado à essa malha, intermediado por um dispositivo *no-break*. Inspirado em um problema recorrente nos laboratórios da Universidade de Brasília (principalmente pelo Laboratório de Dispositivos e Circuitos Integrados - LDCI), este problema sendo a falha no fornecimento de energia elétrica, o projeto proposto evoca uma questão maior na área de qualidade de energia, a qual não está limitada a esse contexto acadêmico e tem sido cada vez mais discutida frente ao surgimento de cada vez mais eletrônicos, eletrodomésticos e dispositivos computadorizados capazes de se conectar à Internet - como se verifica nas tecnologias recentemente em ascensão de Domótica, Automação para Edificações, Internet das Coisas (*IOT*) e *smart grids*. Decomposto em três sub-sistemas (analógico, digital e de rede) com objetivos gerais e específicos diferentes mas interconectados entre si, o produto final deste desenvolvimento foi simulado em computador e implementado em um protótipo que foi subsequentemente testado por meio de experimentos conduzidos em laboratório e emulações em ambiente controlado das possíveis situações relativas ao estado da rede elétrica. Assim, considerando as limitações financeiras e de tempo manifestadas no decorrer da construção do sistema, os resultados finais obtidos e a concretização bem sucedida de todos os objetivos inicialmente propostos, é possível compreender o projeto em questão como uma plataforma de monitoramento altamente expansível, com oportunidades para a integração de diversas outras funcionalidades e com grande potencial para evolução em termos de desempenho.

Palavras-chave: Rede elétrica, qualidade de energia, condicionamento de sinal, digitalização, Arduino, comunicação serial, protocolo MQTT, monitoramento de tensão.

Abstract

The work presented here addresses the implementation of a complete system for local and remote monitoring - through the exchange of data, information and warnings over the Internet - of the voltage signal on the electrical grid in a room or building, of parameters derived from this voltage and, with a lesser focus, of the electric current that feeds a computer/server connected to that mesh, intermediated by a UPS device. Inspired by a recurring problem in University of Brasília's laboratories (mainly by the Integrated Devices and Circuits Laboratory - LDCI), that problem being the failure in supplying electric power, the proposed project evokes a larger issue in the field of energy quality, which is not limited to said academic context and has been increasingly debated given the emergence of even more electronics, appliances and computer devices that are capable of connecting to the Internet - as is the case in technologies recently on the rise like Home and Building Automation, Internet of Things (IOT) and smart grids. Divided into three sub-systems (analog, digital and network) with different but interconnected general and specific objectives, the final product of this development cycle was simulated on a computer and implemented into a prototype that was subsequently tested through lab conducted experiments and by emulating the possible situations related to the electric grid state in controlled environment. Furthermore, considering the financial and time constraints that have arisen during the system's development, the final results obtained and the success accomplishing all of the initially proposed objectives, it is possible to look at the project in question as a highly expandable monitoring platform, with many opportunities for integrating other functionalities and with a great evolutionary potential in terms of performance.

Keywords: *Electrical Network, power quality, signal conditioning, digitalization, Arduino, serial communication, MQTT protocol, voltage monitoring.*

Sumário

1	Introdução	1
1.1	Contextualização	1
1.2	Objetivos	7
2	Projeto do Sistema	9
2.1	Sub-sistema Analógico	11
2.1.1	Fontes Modeladas	12
2.1.2	Transformador	13
2.1.3	Alimentação Positiva	14
2.1.4	Alimentação Negativa	16
2.1.5	Filtro Passa-Baixas Ativo	18
2.1.6	Deslocamento DC	22
2.1.7	Proteção à entrada analógica	23
2.1.8	Entrada Analógica	24
2.2	Sub-sistema digital	27
2.2.1	Bibliotecas, Diretivas e Variáveis	27
2.2.2	<i>Setup</i> inicial	28
2.2.3	Operação em <i>loop</i>	29
2.2.4	Monitor de Falhas	34
2.3	Sub-sistema de redes	36
2.3.1	Publisher	36
2.3.2	Broker	37
2.3.3	Subscribers	38
3	Simulações Computacionais e Resultados Experimentais	40
3.1	Sub-sistema analógico	42
3.1.1	Alimentação Positiva	44
3.1.2	Alimentação Negativa	45
3.1.3	Filtro Passa-Baixas Ativo com Deslocamento DC	48

3.1.4 Testes adicionais	50
3.2 Sub-sistema digital	53
3.2.1 Digitalização da entrada analógica	54
3.2.2 Métricas de qualidade de energia geradas	54
3.2.3 Respostas do monitor de falhas	55
3.3 Sub-sistema de rede	57
3.3.1 Publicação da página <i>web</i>	57
3.3.2 Troca de mensagens por meio do MQTT	58
3.3.3 Disposição dos dados e informações na interface e interação com o usuário	59
4 Considerações Finais	62
4.1 Resultados e Objetivos Cumpridos	62
4.2 Conclusão	64
4.3 Trabalhos Futuros	65
Referências	68

Lista de Figuras

1.1	Componentes inteligentes em um modelo de <i>Smart Home</i>	2
1.2	Modelos de <i>no-break</i> utilizados pelo LDCI.	5
1.3	Servidor utilizado pelo LDCI.	7
2.1	Diagrama do Sistema Desenvolvido.	10
2.2	Montagem do Condicionador de Corrente na placa Arduino Uno.	11
2.3	Modelos de fontes adotados.	12
2.4	Modelo do trafo adotado.	13
2.5	Modelo da fonte de tensão positiva adotado.	15
2.6	Modelo da fonte de tensão negativa adotado.	17
2.7	Modelo do filtro passa-baixas ativo adotado.	18
2.8	Modelo do divisor de tensão adotado.	23
2.9	Diodo Zener de proteção.	24
2.10	Modelo adotado para a entrada de leitura analógica do Arduino..	25
2.11	Circuito condicionador do subssistema analógico.	26
2.12	Ambiente de desenvolvimento Arduino com estrutura básica de programa..	27
2.13	Par de ferramentas disponíveis no ambiente de desenvolvimento do Arduino.	28
2.14	Representação gráfica dos pontos de inflexão numa curva senoidal [1].	33
2.15	Diagrama demonstrando a estrutura e funcionameto em alto nível do protocolo MQTT.	36
2.16	<i>Site</i> do broker de testes Mosquitto.	38
3.1	Sistema final emulado.	41
3.2	Curva simulada do sinal de entrada proveniente da tomada.	42
3.3	Avaliação simulada do circuito projetado.	43
3.4	Protótipo implementado do circuito projetado.	43
3.5	Trafo utilizado na implemetação do sub-sistema.	43
3.6	Gráficos obtidos para o diodo retificador de meia-onda.	44
3.7	Gráficos obtidos para o diodo retificador de meia-onda seguido de um capacitor de filtro.	44

3.8	Gráficos obtidos para a saída do regulador de tensão com o capacitor de estabilização.	45
3.9	Gráficos obtidos para o pino Vcc e RST do <i>timer</i>	45
3.10	Gráficos obtidos para o pino DIS do <i>timer</i>	46
3.11	Gráficos obtidos para o pino THRS e TRIG do <i>timer</i>	46
3.12	Gráficos obtidos para o pino CV do <i>timer</i>	47
3.13	Gráficos obtidos para o pino OUT do <i>timer</i>	47
3.14	Gráficos obtidos para o sinal entre os diodos da fonte de alimentação negativa projetada.	48
3.15	Gráficos obtidos para a tensão gerada pela fonte de alimentação negativa projetada.	48
3.16	Gráficos obtidos para o sinal de baixa tensão na saída do transformador utilizado.	49
3.17	Gráficos obtidos para o deslocamento DC gerado, presente na porta não-inversora do amp-op.	49
3.18	Gráficos obtidos para a tensão presente na porta inversora do amp-op.	50
3.19	Gráficos obtidos para a saída do amplificador operacional com a presença do capacitor para filtragem.	50
3.20	Gráficos obtidos para a saída do amplificador operacional sem a presença do capacitor para filtragem.	51
3.21	Curva obtida na saída do circuito implementado sem a devida alimentação negativa.	51
3.22	Sinal obtido na saída do circuito implementado com o terminal primário do transformador desconectado da tomada.	52
3.23	Arduino Uno utilizado com o <i>Shield</i> de Ethernet montado.	53
3.24	Código desenvolvido no ambiente Arduino.	53
3.25	Avaliação gráfica da quantização do sinal condicionado para diferentes resoluções.	54
3.26	<i>Display</i> do monitor de falhas para as diferentes situações possíveis com relação a energia e conexão com a internet.	56
3.27	Hardware da interface do sub-sistema digital com o de redes na rota de segurança do sistema.	57
3.28	Página criada iniciada indicando estado desconhecido (<i>Unkown</i>) do sistema.	58
3.29	Página publicada com sucesso pelo <i>host</i> GitHub Pages.	58
3.30	<i>Display</i> do monitor de falhas para os testes de envio de mensagens realizados.	59
3.31	Página <i>Web</i> com ferramentas de desenvolvedor a mostra para os testes de recebimento de mensagens realizados.	60

3.32	Demonstração das ferramentas para análise de amostras com o sistema em estado de funcionamento normal (<i>OK</i>), visualizada por meio de um navegador em <i>smartphone</i>	60
3.33	Página <i>Web</i> visualizada em navegador de <i>notebook</i> para duas das diferentes situações possíveis com relação a energia e conexão.	61

Lista de Tabelas

2.1 Tabela de possíveis estados do sistema.	31
3.1 Resultados de ensaios feitos em rede elétrica residencial.	55
4.1 Orçamento do projeto desenvolvido.	66

Capítulo 1

Introdução

Ao presente capítulo cabe as funções de contextualizar a atual conjuntura relativa ao tema e o problema a ser resolvido em si (incluindo sua especificação, importância, e solução cabível), definir os principais objetivos (gerais e específicos) do sistema que será desenvolvido e propor um plano de execução do projeto como um todo.

1.1 Contextualização

Hoje, está mais do que claro o importante papel que dispositivos eletrônicos e eletrodomésticos têm na vida cotidiana de qualquer indivíduo ou organização de indivíduos, seja em aspectos pessoais, sociais, profissionais ou acadêmicos. Desde o início do século XX [2], ferros de passar roupa, chuveiros elétricos, lâmpadas incandescentes, fluorescentes e de *LED*, computadores pessoais como *desktops*, *laptops* e *smartphones*, entre outros, ganharam e continuam a ganhar cada vez mais espaço no dia-a-dia das pessoas e consequentemente no mercado [3].

Nesse mesmo contexto, percebe-se que a computação vêm obtendo uma participação maior dentre as funcionalidades desses dispositivos, e não só em aparelhos mais especificamente relacionados à computação pessoal, como os já mencionados *desktops*, *notebooks* e *smartphones*, mas sim em todo um espectro de utensílios, sejam estes domésticos (como é o caso da **Domótica**) [4] ou não (tratando-se de forma mais ampla da **Automação para Edificações**) [5]. De acordo com uma pesquisa feita pela *Research and Markets* em 2015 por exemplo: “O mercado global de automação residencial e controle de segurança valia US\$ 5,77 bilhões em 2013 e deve atingir US\$ 12,81 bilhões até 2020; projeta-se para este mercado um crescimento em CAGR (Taxa Composta Anual de Crescimento) de 11,36% entre 2014 e 2020” [6]. Nesse sentido, é possível destacar dentre os eletrodomésticos citados e incluindo outros que hoje já estão disponíveis comercialmente como produtos *smart*: 1. alarmes de segurança, 2. geradores, 3. termostatos, 4. geladeiras, 5.

lampadas, 6. detectores de fumaça, 7. fornos, 8. fechaduras eletrônicas, 9. válvulas de água com alarme e 10. máquinas de lavar e secar (Figura 1.1). Dessa forma, ainda que ferramentas de *Home/Building Automation* não sejam aplicações muito recentes ou surgidas de conceitos particularmente novos - a primeira tecnologia de automação doméstica de uso geral, X10, foi desenvolvida em 1975 [7] e, em 1987, o protocolo de comunicações para Automação e Controle de Edificações (*BAC* ou *Building Automation and Control*) *BACnet* surge [8] - observa-se de fato uma tendência de crescimento e evolução dessas aplicações em função do desenvolvimento e barateamento de técnicas de comunicação *wireless*, controle/automação [5] e computação distribuída [9], dentre outras.

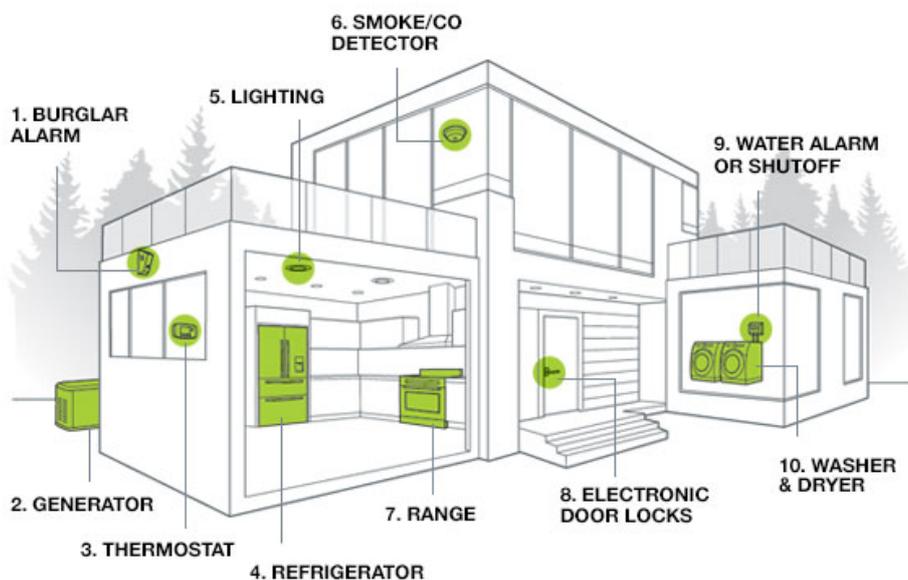


Figura 1.1: Componentes inteligentes em um modelo de *Smart Home* (Fonte: [10]).

Atualmente, a forma como tais técnicas têm impulsionado o desenvolvimento de aplicações em Automação Residencial e de Edificações vem, conseqüentemente, permitido também a eclosão de dispositivos *IOT* [11]. Termo mais provavelmente cunhado em 1999 [12], *Internet of Things* como conceito, assim como *Home* ou *Building Automation*, não representa nada muito novo - o primeiro utensílio conectada à uma versão mais primitiva da Internet, no caso uma máquina de venda de Coca-Cola modificada da Universidade Carnegie Mellon (EUA) [13], data do início dos anos 80 - mas ganha novo destaque hoje em função das inovações mencionadas e passa a ganhar destaque maior ainda em função da praticidade de endereçamento extensivo trazido pela popularização do *IPv6* [14]. Evidência disso pode ser vista em estatísticas de crescimento no número de dispositivos *IOT* disponíveis, que indicam aumento de 31% ao ano até 2017 quando atingiu a marca

de 8,4 bilhões de dispositivos [15], e em projeções de que esse número atinja a soma de 30 bilhões [16] com um valor de mercado global estimado em US\$ 7,1 trilhões [17] até 2020. Diante disso, a **Internet das Coisas** envolve, em sua crescente presença e progresso, aplicações nas mais diversas áreas: urbanismo, ambientalismo, distribuição de água, segurança e emergência, varejo, logística, controle industrial, agricultura, pecuária, domótica, saúde e, particularmente relevante no contexto deste trabalho, *smart metering*, mais especificamente com relação à *smart grids*, operantes no monitoramento e gestão do consumo de energia elétrica [18].

Tratando-se de uma aplicação baseada no uso de sensores e protocolos de comunicação integrados a objetos do dia-a-dia para monitoramento, análise e controle remotos de parâmetros físicos, a tecnologia de *smart metering* tem grande potencial na identificação e solução dos mais diversos problemas cotidianos [19]. Estes parâmetros, quando representam valores de tensão, corrente, fator de potência ou frequência relativos a sinais elétricos, possibilitam que os dados coletados pelos sensores sejam armazenados e processados em relatórios gráficos sobre estado atual do consumo de energia elétrica de um aparelho individual [20][21]. Estes relatórios, por sua vez, podem ser de grande utilidade na instituição de políticas de economia energética por parte do próprio administrador (determinação de um limite máximo permitido de energia consumida para um certo período de tempo, por exemplo) ou mesmo geradas e sugeridas de forma automática, baseadas no comportamento habitual do usuário (um cronograma de funcionamento automático otimizado, por exemplo) [20][21][22]. Além disso, a capacidade de comunicação incorporada também possibilita acesso remoto às informações produzidas e ao controle de certas funções do sistema (ligar e desligar, por exemplo), tanto por parte de um administrador quanto por parte de outros sistemas de outros aparelhos conectados a uma mesma rede *smart grid* [20][22].

Assim, tal tecnologia se apresenta como uma solução interessante à medida que poderia estar convenientemente acompanhando o rápido crescimento no número de eletrônicos e de eletrodomésticos em circulação (dado que os parâmetros físicos relacionados à energia elétrica consumida são mensuráveis para quaisquer desses utensílios) [23], ainda que limitada em certos aspectos. Dentre os grandes prós, destacam-se o acompanhamento e gerenciamento de energia para uso eficiente da eletricidade [19] - contrapondo-se ao constante aumento no consumo [22] e geração de eletricidade que, em 2012, chegou a representar 32% das emissões de carbono nos EUA (das quais mais de 70% foram atribuídos ao consumo residencial e comercial) [24]. Já por outro lado, ainda há desvantagens e limitações a serem superadas: desafios de segurança e privacidade, custos elevados, instalações complexas, entre outros [19].

Com essa discussão em mente, então, pode-se começar a explorar o problema a ser resolvido.

Problema

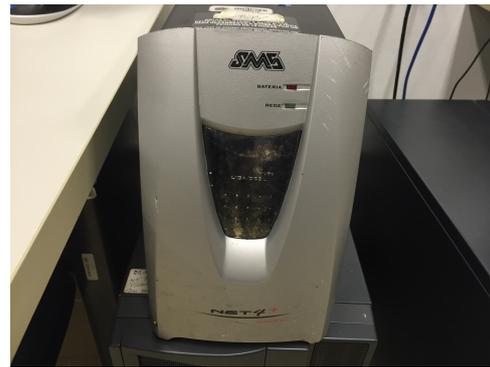
Inserido em um contexto acadêmico, o Laboratório de Dispositivos e Circuitos Integrados (LDCI) localizado no prédio SG-11 da Universidade de Brasília é apenas mais um dentre outros laboratórios no campus universitário Darcy Ribeiro que estão sujeitos a ocasionais oscilações em sua rede elétrica e quedas totais de energia. De fato, trata-se de um problema recorrente [25] e, em razão de anomalias como essas, alguns edifícios abrigam geradores a diesel como fonte de energia secundária em casos de falhas na alimentação da malha primária. Além disso, o próprio LDCI dispõe de alguns dispositivos *no-break* de diferentes potências - 600VA, 1400VA e 2000VA, além de um modelo de 3100VA para o servidor presente em uma sala separada - como forma de proteção adicional. Estes alteram seu modo de operação da configuração Rede (funcionamento normal) para Bateria (rede desenergizada) quando necessário, garantindo por um período de tempo limitado - ou até que a fonte de energia primária do prédio seja restaurada - que os computadores e servidores conectados às tomadas do aparelho continuem suas operações ininterruptamente e sem maiores complicações [26].

Entretanto, é inevitável que circunstâncias ainda mais adversas possam vir a ocorrer. Servidores que permanecem funcionais no decorrer de um fim de semana podem ter até o momento seu longo processamento subitamente interrompido em função de uma falha de energia não tratada antes do descarregamento total da bateria no *no-break*, ou seja, não houve quem percebesse a queda na energia ou responsável pela ativação de alguma alimentação secundária nessa ocasião. Expandindo esse olhar para laboratórios em outros campi, é possível observar situações similares: “Estudantes de pós-graduação em Ciências da Saúde da Universidade de Brasília (UnB) afirmam que perderam quatro meses de trabalho em uma pesquisa sobre câncer por causa da queda de energia no campus de Ceilândia. Segundo os alunos, o corte no fornecimento impediu que células da doença que eram mantidas em um freezer pudessem sobreviver” [27].

Ademas, pensando de forma um pouco mais aprofundada, as ocasionais oscilações da rede elétrica mencionadas também podem vir a originar certos problemas. Além da questão do uso ineficiente de energia elétrica por aparatos cotidianos apontado anteriormente [22], alguns equipamentos de alta precisão podem apresentar sensibilidade elevada a flutuações na fonte de fornecimento de energia à qual estão conectadas - o maquinário de alguns laboratórios do Instituto de Biologia da UnB campus Darcy Ribeiro se encaixa nessa descrição. Assim é possível que, em redes elétricas instáveis, o funcionamento irregular de instrumentos de medida utilizados em certos experimentos produzam resultados



(a) 600 VA



(b) 1400 VA



(c) 2000 VA



(d) 3100 VA (alimenta o servidor do LDCI)

Figura 1.2: Modelos de *no-break* utilizados pelo LDCI.

imprecisos ou até incertos, inclusive sem a ciência de quem conduziu os testes - o que torna particularmente difícil determinar a origem da falha na experimentação ou mesmo se por certo houve uma falha. Dessa forma, torna-se evidente que não se trata de um problema localizado e específico e, importante acrescentar, não exclusivamente um problema restrito à um contexto acadêmico.

Justificativa

De um ponto de vista mais amplo, a situação apresentada trata de um problema de qualidade de energia. **Qualidade de Energia**, ainda que não tenha uma definição exata ou unanimemente aceita, pode ser melhor descrita como o fornecimento de tensões e projeto de sistemas que garantam o sucesso na utilização, por parte de um usuário, da energia elétrica de um sistema de distribuição sem interferências ou interrupções. Parte disto, o entendimento de que a área envolve o estudo de parâmetros tais como forma de onda de um sinal *AC* (tensão ou corrente), a presença de sinais harmônicos em voltagens de barramentos e em correntes de cargas, existência de picos e momentos de baixa tensão, entre outros [28].

Neste quadro, é interessante notar que um sistema de fornecimento de energia com qualidade excepcional (referência na avaliação de qualidade) poderia ser qualificado pelo fornecimento de energia em forma de um sinal de tensão puramente senoidal, sem quaisquer variações em sua amplitude ou frequência. Em contrapartida, um sistema de fornecimento de energia com qualidade inferior implicaria a provisão de uma sinal de tensão inconsistente com essa forma, com distorções, às vezes, tão expressivas que poderiam ser capazes de prejudicar o desempenho, como descrito anteriormente, ou mesmo de danificar um equipamento [29].

Partindo dessa compreensão, hoje se tornam ainda mais evidente os problemas associados à (falta de) qualidade de energia. Ainda que a sensibilidade seja de fato variável entre tipos de equipamento diferentes - ou seja, o que um dispositivo considera baixa qualidade de energia pode ser considerada qualidade de energia razoável para outro - há, atualmente, uma tendência geral de crescimento na sensibilidade à variações de parâmetros (tais como níveis de distorção da onda ou a variação da amplitude) em função do uso cada vez mais amplo de eletrônica embarcada [29].

Proposta

Estabelecido o problema em questão e a justificativa para tal, abrimos caminho para uma discussão de fato mais prática e, no que diz respeito ao projeto desenvolvido, relevante. A solução para a complexidade explicitada não é única e poderia ser fundamentada partindo das mais diversas abordagens, mas para fins didáticos e de praticidade a questão será tratada, pelo menos em um primeiro momento, do ponto de vista das necessidades do já citado LDCI. Com isto acertado, a atual discussão pode ser sintetizada na seguinte vontade: ter acesso a informações do estado da rede elétrica no laboratório (incluindo se esta se encontra energizada, modo de operação do dispositivo *no-break* e dados relativos à qualidade de energia) ao longo de um certo período e em tempo real, com acesso local ou remoto. A partir dessa especificação surge então a ideia de um sistema analógico-digital simples, conectado à Internet e indiretamente alimentado por *no-break* capaz de monitorar a tensão em uma tomada da rede de energia e de enviar dados, informações e alertas a respeito dessa voltagem para clientes locais ou remotos (computadores no caso). Por conseguinte, a realização de um sistema tal como descrito possibilitaria identificar e responder a faltas de energia muito mais prontamente, ademais de fornecer noções sobre consumo e qualidade energética de uma carga individual do ambiente a um usuário onde quer que este esteja.

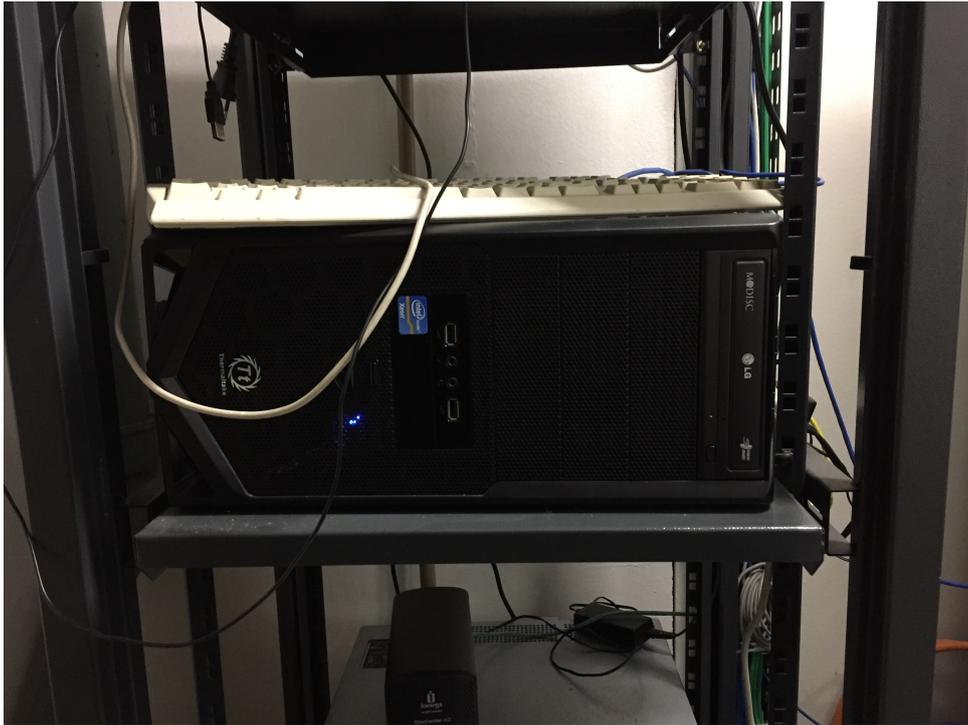


Figura 1.3: Servidor utilizado pelo LDCI.

1.2 Objetivos

Sumariamente, o projeto consiste na implementação de um sistema capaz de detectar falhas no fornecimento de energia para alertar computadores conectados à rede elétrica e clientes remotos. Para execução deste objetivo principal de maneira eficaz, decompõe-se este em três objetivos gerais menores, cada qual, por sua vez, é decomposto em cinco objetivos específicos, como hierarquizado a seguir:

- Garantir medida adequada, precisa e eficiente dos valores de tensão em uma tomada da rede de energia *AC* no laboratório:
 - Gerar um novo referencial terra para conformidade com instrumento de medida;
 - Desacoplar da malha o circuito condicionador e reduzir a tensão para leitura;
 - Fornecer alimentação adequada aos componentes eletrônicos utilizados;
 - Filtrar ruído indesejado (não relacionado à imperfeição da malha);
 - Isolar e proteger a carga do dispositivo responsável pela medida.
- Ler e estimar, a partir dos dados mensurados, informações que caracterizem a situação da energia fornecida pela malha e que sejam úteis na identificação de anomalias:
 - Converter o sinal de entrada analógico em digital;

- Coordenar leitura e processamento simultâneos das amostras feitas;
 - Representar forma de onda equivalente à tensão em vetor ou *stream*;
 - Calcular frequência do sinal lido;
 - Obter valores RMS, de pico e de vale relativos ao sinal lido.
- Estabelecer uma forma de comunicação rápida, confiável e segura para troca de dados, informações e avisos entre o sistema em laboratório e computadores (locais ou não) gerenciados por um administrador:
 - Eleger protocolos para envio e recepção em função do tipo de mensagem;
 - Padronizar formatos gerais das mensagens;
 - Desenvolver interface cliente-servidor;
 - Criar interface gráfica para cliente;
 - Definir restrições de acesso ao sistema em laboratório.

Logo, solucionar o problema principal mais complexo passa equivaler a solucionar quinze problemas mais simples - “dividir para conquistar” [30]. Categorizados em conjuntos de cinco de acordo com sua relevância para cada um dos três problemas gerais maiores, torna-se intuitivo o roteiro de abordagem destes.

Capítulo 2

Projeto do Sistema

O presente capítulo tem foco na implementação do sistema proposto. As seções que seguem tratam, respectivamente, dos sub-sistemas analógico, digital e de redes, propondo as soluções concretas dos problemas estabelecidos. Antes disso, para se ter uma visão geral das partes que compõem o sistema como um todo e que realizam a comunicação entre os sub-sistemas referidos, é apresentado o diagrama da Figura 2.1. Neste são apresentados os principais blocos que fazem a implementação do trabalho proposto incluindo as tecnologias e ferramentas mais importantes no seu desenvolvimento de uma forma mais superficial. Dentre os blocos e interfaces apresentados, o texto redigido neste capítulo entrará em maior detalhes a respeito de apenas alguns enquanto o funcionamento dos demais será suposto como já conhecido - pensados como caixas-pretas - a fim de manter o foco apenas naquilo que foi projetado no escopo deste projeto. Assim, alguns dos itens ilustrados no esquema da figura em questão estão presentes para manter a concisão com o sistema real e devem ser apenas brevemente citados no decorrer da redação. No caso específico do componente "Sensor de Efeito Hall (ACS712 -5A)" e do quarto sub-sistema apresentado como "Banco de Dados de Sinais" tem-se representado em alto nível parte de um projeto complementar a este, desenvolvido em paralelo como projeto final de graduação do aluno Marcos Vinicius Prescendo Tonin na Universidade de Brasília em 2/2018 - detalhes de implementação dessas partes podem ser conferidos no texto do autor com título "Desenvolvimento do frontend de uma plataforma de aferimento de qualidade de energia"[31].

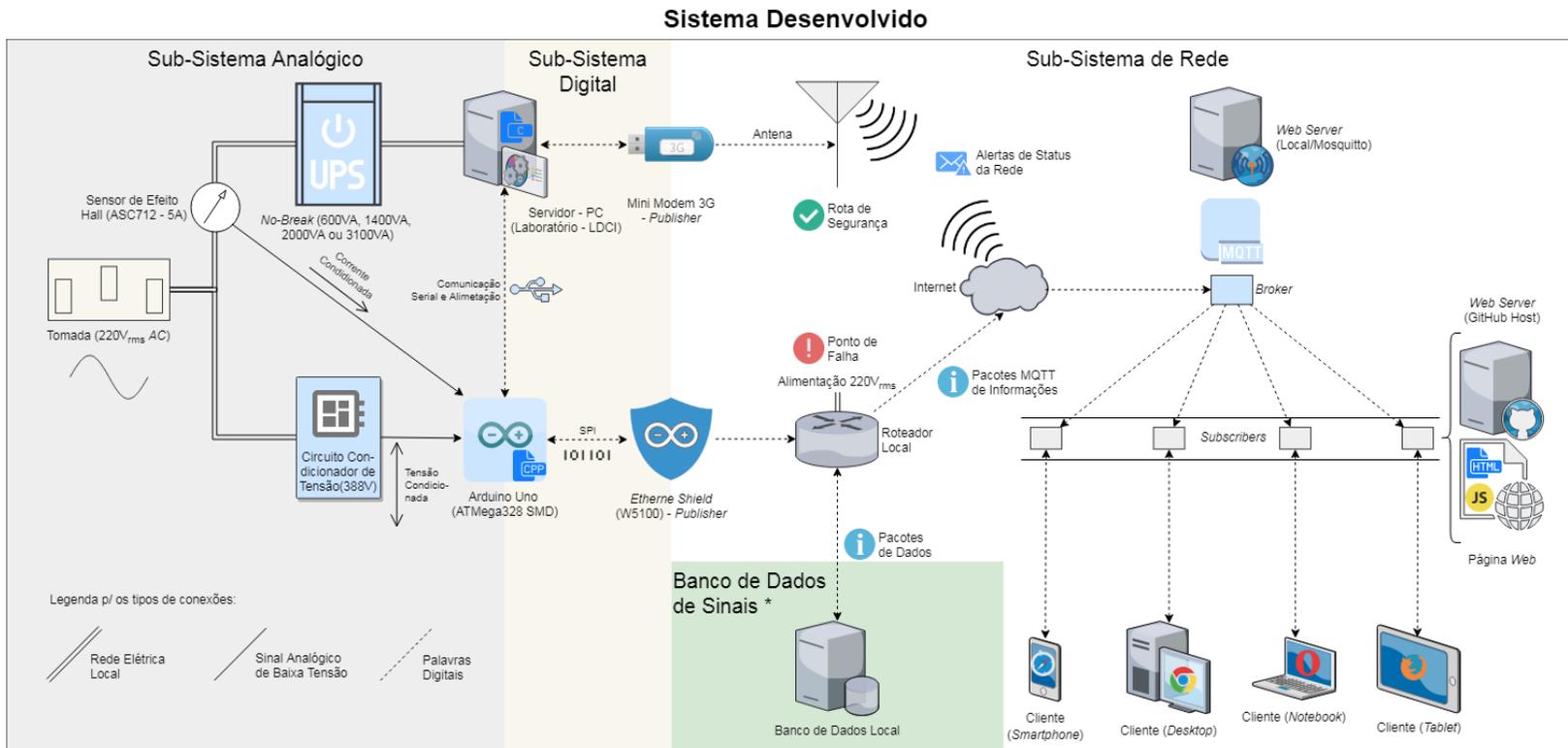


Figura 2.1: Diagrama do Sistema Desenvolvido.

2.1 Sub-sistema Analógico

No contexto deste trabalho o termo “condicionamento” refere-se à ação de adequar um sinal para que este - ou um semelhante a este em certos aspectos - tenha condições de servir como entrada para algum outro sistema específico de forma confiável (de forma adequada e com garantia de alguma equivalência/proporcionalidade com o sinal original). Estabelecida essa idéia, deve-se aplicá-la aos sinais de interesse do estudo sendo feito, no caso tensão e corrente elétrica. Começando o desenvolvimento pela parte analógica do sistema, esta secção concentrará esforços apenas em descrever o processo de concepção do circuito condicionador de tensão e pontos relativos às suas respectivas interfaces com outros blocos e sub-sistemas do diagrama ilustrado. O funcionamento do circuito para o condicionamento de corrente citado será abstraído, bastando supor que se comportará como esperado: gerando uma tensão na entrada analógica proporcional à corrente sentida (vide Figura 2.2).

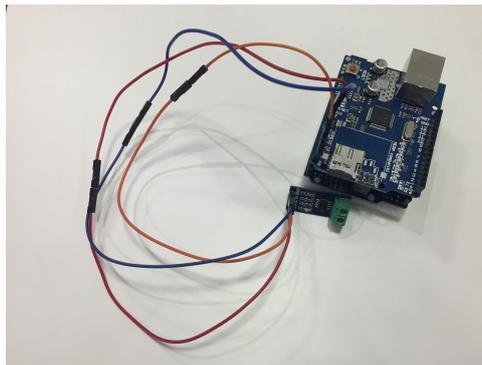


Figura 2.2: Montagem do Condicionador de Corrente na placa Arduino Uno.

De forma resumida, deseja-se que o sinal de entrada do sub-sistema proveniente de uma tomada elétrica ideal com magnitude $220V_{rms}$ (ou $311.127V_{pk}$) e frequência de $60Hz$ seja transformado em um sinal de saída independente e equivalente com magnitude $2V_{pk}$, frequência de $60Hz$ e um deslocamento vertical de $+2.5V$, garantindo uma variação de tensão entre $0.5V$ e $4.5V$ com $0.5V$ de margem de segurança acima e abaixo do sinal de saída ideal. Dessa maneira, considera-se o “caminho” do sinal de interesse desde a tomada até a entrada do microcontrolador e as ocasionais “ramificações” que auxiliam no funcionamento do sub-sistema:

2.1.1 Fontes Modeladas

Considera-se a disponibilidade de duas fontes de tensão: o próprio sinal alternado vindo da tomada de uma instalação elétrica e a alimentação contínua fornecida pelo microcontrolador. Detalhando ambos:

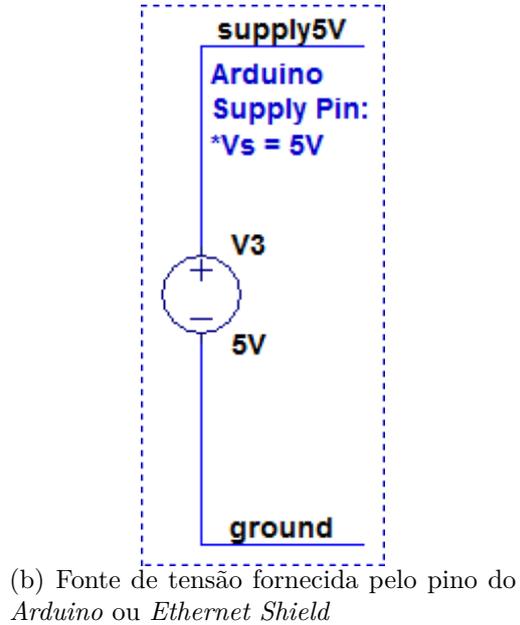
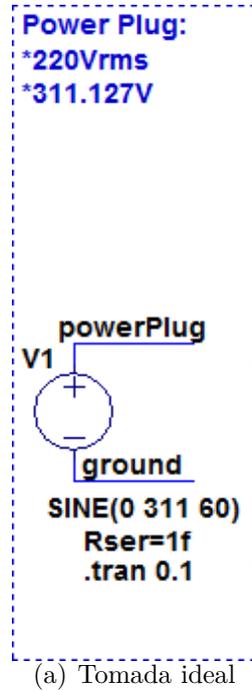


Figura 2.3: Modelos de fontes adotados.

Sinal da tomada

Como apresentado na Figura 2.3(a), modela-se a fonte do sinal a ser analisado como um gerador de senoides de amplitude igual a $220V_{rms}$ (aproximadamente $311.13V_{pk}$) e frequência igual a $60Hz$. Trata-se de uma situação ideal que provavelmente não reflete o que acontece na realidade - disparidade que de fato é a razão para estudo apresentado - mas para fins de projeto é adequada.

$$V_{tomada} = 311.13 \cdot \text{sen}(2\pi \cdot 60 \cdot t)V \quad (2.1)$$

Fonte DC Arduino

O modelo *Uno* do **Arduino**, utilizado no projeto, possui pinos de alimentação *DC* capazes de fornecer tensões próximas de $3.3V$ e $5V$ com uma corrente máxima para saída de aproximadamente $40mA$ [32]. Destas duas fontes, interessa ao circuito desenvolvido a segunda, a qual, como ilustrado na Figura 2.3(b), é modelada como uma fonte ideal

de $5V$ DC , que aproxima relativamente bem a realidade (como mostrado no próximo capítulo).

$$V_{arduino5} = 5V \quad (2.2)$$

2.1.2 Transformador

Para o transformador/trafo utilizado na implementação do trabalho foi escolhido um modelo do tipo abaixador com uma única espira primária e duas espiras secundárias. Os requisitos definidos para o abaixamento de tensão desejado - e conseqüentemente para a proporção entre as espiras primária e secundárias - nas duas espiras de saída partiram das necessidades simultâneas de obter-se um sinal de baixa tensão equivalente àquele presente na tomada e de obter uma fonte de baixa tensão alternada para geração de uma segunda alimentação contínua positiva para o circuito analógico.

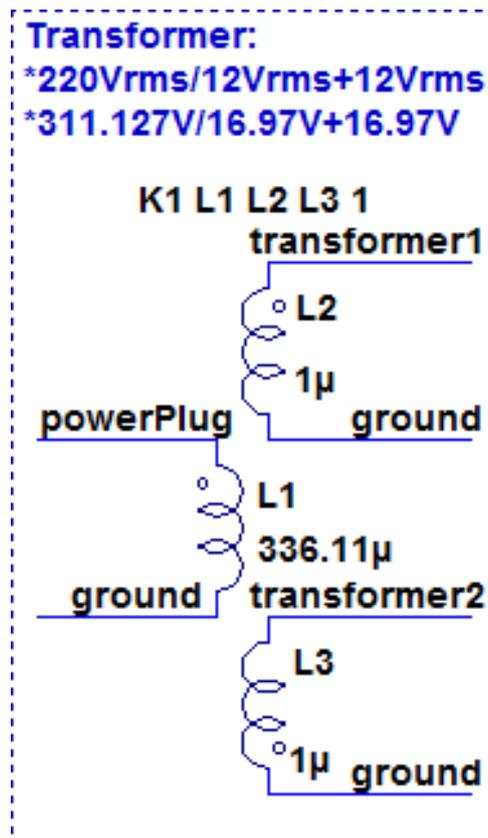


Figura 2.4: Modelo do trafo adotado.

Sinal de baixa tensão equivalente (primeira saída)

Em um primeiro momento, é importante abaixar a tensão proveniente da tomada para valores mais adequados os quais possam ser trabalhados por um variedade de componentes elétricos e eletrônicos sensíveis e com limitações relativas à voltagem aplicada. Além disso, é igualmente importante que esses valores abaixados preservem as demais características do sinal original proveniente da malha elétrica, como frequência e a forma geral da onda.

Fonte de baixa tensão alternada (segunda saída)

Como descrito para a primeira saída, abaixar a tensão de entrada para valores adequados a serem manipulados por componentes elétricos/eletrônicos é o principal propósito desta espira secundária. Mas diferentemente dessa primeira, não há necessidade maior de manter as características originais do sinal de entrada, dado que o objetivo dessa saída é alimentar um gerador de tensão contínua (desenvolvido a seguir).

Considerando esses aspectos, assim como a variedade de modelos disponíveis para esse tipo de equipamento no mercado, decidiu-se por um tipo “ $220V_{rms}/12V_{rms} + 12V_{rms} 800mA_{rms}$ ”, significando um **trafo com tomada central** capaz de transformar uma tensão primária de entrada com aproximadamente $311.127V_{pk}$ em duas tensões secundárias de saída com aproximadamente $16.97V_{pk}$ cada (para um mesmo referencial), com uma capacidade de fornecimento de corrente próxima de $1.131A_{pk}$ - mais do que suficiente para o funcionamento do sub-sistema. Assim, modelou-se tal componente como apresentado na Figura 2.4 de acordo com a Equação 2.3 [33] (vale ressaltar que a referência ‘ground’ da espira primária não é a mesma dos ‘grounds’ das espiras secundárias).

$$\frac{L_1}{L_{2/3}} = \left(\frac{N_1}{N_{2/3}}\right)^2 = \left(\frac{V_1}{V_{2/3}}\right)^2 = \left(\frac{220V_{rms}}{12V_{rms}}\right)^2 = 336.11 \quad (2.3)$$

2.1.3 Alimentação Positiva

Inicialmente, sendo o pino de alimentação do Arduino o único acesso a alimentação contínua disponível para o restante do circuito, é de se esperar que se alimente os demais componentes com os 5 volts de tensão facilmente acessíveis do microcontrolador. Porém, em função de não-linearidades e imperfeições desses componentes (discutidos posteriormente), torna-se evidente a necessidade de uma segunda **fonte de alimentação DC** com maior voltagem para garantir o bom funcionamento do condicionador de tensão. Dito isso, é proposta uma fonte geradora de tensão como a esquematizada na Figura 2.5 para disponibilizar uma voltagem constante de $12V$ para o resto dos componentes eletrônicos no sistema - incluindo, hipoteticamente, até para o próprio microcontrolador utilizado. Esta fonte pode ser dividida em quatro partes, cada qual com sua função específica.

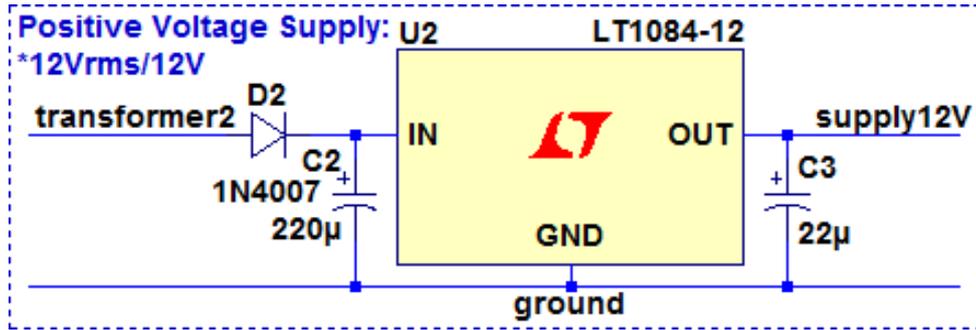


Figura 2.5: Modelo da fonte de tensão positiva adotado.

Diodo retificador

O papel dessa peça é cortar a ação de tensões negativas no sinal, ou retificar o sinal, de forma a torná-lo estritamente positivo. Tratando-se de um **retificador de meia-onda**, a frequência do sinal é mantida e, conseqüentemente, o período entre picos também (não há mais vales, ao menos no sentido de pontos únicos). Com isso estabelecido e considerando um sinal de entrada senoidal fornecido pela segunda espira secundária do transformador, decidiu-se por utilizar um único diodo modelo “1N4007” com queda de tensão característica de aproximadamente $1.1V$ [34], o que em teoria resultaria em um sinal retificado com valor de pico perto de $16.97V - 1.1V = 15.87V$ (considerando apenas a queda de tensão constante no diodo).

Capacitor de filtro

Representado pelo capacitor polarizado C2 na Figura 2.5, o desempenho do capacitor de filtro como redutor das variações de pulsos vindos do diodo retificador depende, de forma geral, das variáveis representadas pelo valor atribuído à sua capacitância e pela impedância da carga conectada em paralelo à este componente. Como não se tem controle total do valor de impedância que o sucede, o valor da capacitância em si será a variável que deve-se escolher para o funcionamento adequado dessa parte da fonte. Para boa medida, escolheu-se um **capacitor eletrolítico** de $220\mu F$, valor de capacitância relativamente alto mas com o tamanho físico do capacitor suficientemente pequeno, garantindo assim um *ripple* razoável. Fruto de uma escolha parcialmente arbitrária, a eficiência do capacitor escolhido, apesar de não calculada formalmente neste texto (na forma de parâmetros relativos ao *ripple* mencionado), é demonstrada nos resultados de simulações e experimentais abordados no próximo capítulo, mas neste momento pode-se levar em consideração que o capacitor de filtro fornecerá uma tensão de cerca de $15.87V$.

Regulador de tensão

A próxima parte da fonte de tensão em construção é o regulador de tensão, representado na Figura 2.5 pelo *label* “U2”. Operando sobre os 15.87V obtidos anteriormente, o eletrônico escolhido - no caso o **L7812** (equivalente ao LT1084-12, utilizado nas simulações e diagramas [35]) - gera os 12V de tensão constante desejados, sendo capaz de alimentar até 1.5A de corrente em sua saída. Sem entrar em muito mais detalhe a respeito de suas especificações, o aparelho com a entrada dada por si já gera um valor constante adequado para alimentação requerida, mas uma etapa adicional ainda pode ser adicionada por precaução.

Capacitor de estabilização

Por fim, como forma de garantir maior estabilidade para o valor de tensão na saída do regulador, recomenda-se a inclusão em paralelo de um segundo capacitor na fonte [35], representado na Figura 2.5 pelo **capacitor de tântalo polarizado C3** com valor nominal $22\mu F$. Este agirá de forma a evitar o descarregamento repentino da fonte em caso de surtos momentâneos de corrente na carga.

2.1.4 Alimentação Negativa

De forma semelhante ao caso da alimentação positiva, o circuito condicionador por si só não possui uma fonte de alimentação negativa adequada para a boa operação de aparelhos eletrônicos que compõe o sub-sistema abordado. No mundo ideal da teoria, o uso das alimentações positivas e de um referencial terra seriam suficientes para o funcionamento desses aparelhos, mas na realidade, devido às mesmas não-linearidades e imperfeições citadas, torna-se necessária a presença de uma **fonte de alimentação negativa constante** capaz de produzir uma voltagem que compense por essas falhas. Com isso em mente, propõe-se a fonte esquematizada na Figura 2.6 abaixo, capaz de fornecer uma voltagem negativa constante máxima de $-5V$ (a partir da fonte de 5V do Arduino) para o sub-sistema discutido. Para uma análise mais simples e direcionada, divide-se esta fonte em duas partes funcionais.

Temporizador

O que está sendo referido quando se escreve “temporizador” é o conjunto que engloba todos os componentes e conexões a esquerda do capacitor polarizado C6 na Figura 2.6. De maneira geral, a função desse conjunto é gerar um trem de pulsos (como um *clock*) de uma certa amplitude e com certa frequência a partir da tensão de entrada proveniente de alguma das fontes de alimentação positiva apresentadas. No caso, por questões

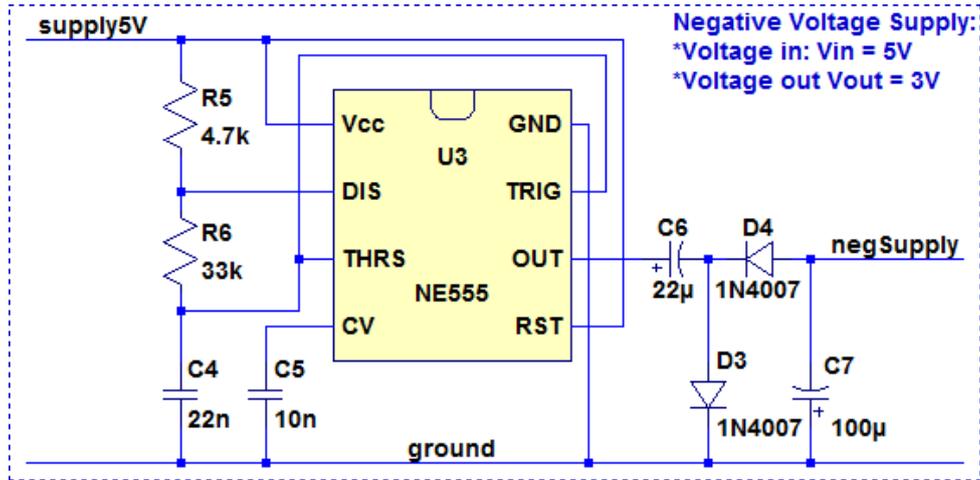


Figura 2.6: Modelo da fonte de tensão negativa adotado.

de velocidade da resposta no circuito integrado e de tensão negativa mínima requerida, ponderou-se que os 5V disponibilizados pelo Arduino seriam suficientes para o resultado desejado. A operação desta parte gira em torno do circuito integrado modelo **NE555**, um *timer* de precisão configurável feito para trabalhar com tensões de entrada/saída entre 5V e 15V e apto a receber e proporcionar magnitudes de até 200mA [36]. Nessa configuração, o conjunto como apresentado na Figura 2.6 deve gerar o *clock* como desejado com a frequência de operação na casa dos kHz.

Dobrador de tensão

Em continuação à primeira parte (temporizador) o dobrador de tensão é o conjunto de componentes restantes, à direita e incluindo o capacitor polarizado C6 na Figura 2.6. O funcionamento dessa segunda parte também dependerá da frequência do sinal gerado em sua entrada com relação ao tempo de atuação dos componentes elétricos (capacitores) e eletrônicos (diodos), com dois princípios básicos mantidos na situação apresentada: o par capacitor-diodo à esquerda grampeia os valores de pico da tensão de entrada em 0V - gerando valores de vales negativos com o dobro da amplitude de entrada (em outras palavras, iguais à amplitude pico-a-pico da tensão ou $-5V$ no caso) - enquanto o par capacitor-diodo à direita detecta e mantém o valor da tensão de vale obtida do par precedente. Assim, considerando o uso do modelo **1N4007** para ambos diodos ($V_f = 1.1V$ [34]) e de capacitores eletrolíticos de valores nominais de capacitância iguais a $22\mu F$ e $100\mu F$ respectivamente, espera-se obter uma tensão de saída aproximadamente igual a $-(5V - 2 \cdot 1.1V) = -2.8V$ (novamente, considerando exclusivamente uma queda de tensão constante nos diodos). Esse valor de alimentação obtido, apesar de ser consideravelmente

menor em magnitude quando comparado à alimentação de $-5V$ gerada por componentes ideais, mostrou-se satisfatório para garantir a não ocorrência das situações falhas citadas.

2.1.5 Filtro Passa-Baixas Ativo

Apresentadas as etapas iniciais do circuito desenvolvido - resumidas no abaixamento de tensão pelo trafo escolhido e na geração de voltagens de alimentação positivas e negativas a partir das fontes projetadas - dá-se início a discussão relativa ao refinamento do condicionamento feito até o momento. Com tensões variáveis de $-16.97V$ a $16.97V$, o sinal senoidal provido pela primeira espira secundária ainda possui uma magnitude muito elevada para ser medida pelo microcontrolador adotado. Logo é preciso de uma segunda etapa de abaixamento, tal qual resulte em outro sinal equivalente ao original, mas com uma variação de tensão ainda menor. O circuito apresentado na Figura 2.7 realiza esse refinamento por meio de um **filtro passa-baixas ativo**, que não só diminuí a tensão de saída com relação a de entrada mas também faz a filtragem de ruídos indesejados, realiza a soma de um deslocamento DC ao sinal redimensionado e promove o isolamento de uma carga conectada à sua saída. Segue abaixo, em detalhe, cada aspecto do modelo esquematizada na figura:

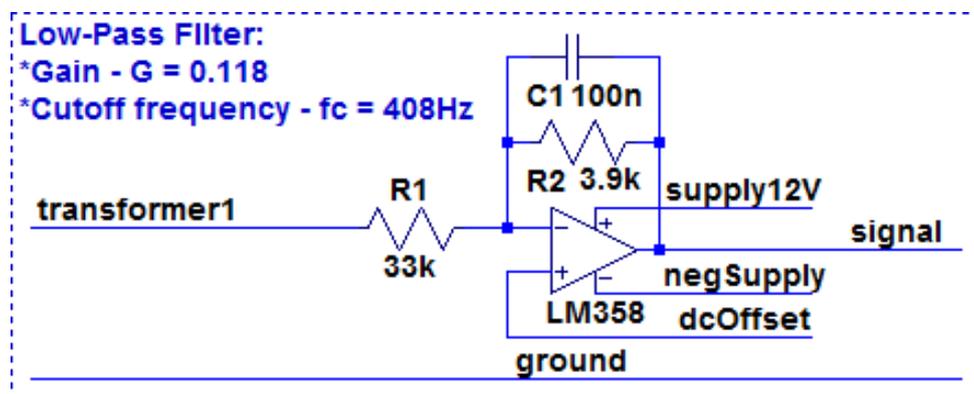


Figura 2.7: Modelo do filtro passa-baixas ativo adotado.

Alimentação

Os pontos levantados anteriormente relacionados à não-linearidades/imperfeições em componentes eletrônicos que precedem os projetos das fontes de alimentação dizem respeito basicamente ao circuito integrado utilizado nesta fase do “caminho” até a entrada analógica do Arduino. De acordo com o *datasheet* do CI **LM358**, utilizado na implementação do filtro da Figura 2.7, o par de amplificadores operacionais encapsulados deve ser alimentado por tensões (não necessariamente simétricas) de amplitudes variando de

$\pm 1.5V$ a $\pm 16V$ - ou em termos pico-a-pico intervalos de $3V$ e $32V$. Mas vale observar que o intervalo de tensão alcançado pela saída deste dispositivo, em função das não-linearidades/imperfeições mencionadas, não equivalerá ao intervalo de alimentação e, de acordo com a documentação referida, deve ser de $1.5V$ a $2V$ menor em amplitude do que a alimentação fornecida - ou em termos pico-a-pico de $3V$ a $4V$ menor [37]. Isso significa que para se obter uma saída que varie entre $0V$ e $5V$ sem maiores distorções deve-se fornecer ao dispositivo eletrônico intervalos de alimentação de $-1.5V$ a $6.5V$ ou até de $-2V$ a $7V$. Uma alternativa ao uso de amplificadores convencionais para solucionar esse problema seria substituí-los por amplificadores *rail-to-rail*, capazes de gerar saídas muito mais próximas da alimentação fornecida - ainda que não alcancem de fato estes limites [38]. Entretanto, em função tanto da disponibilidade como do preço dessa categoria de componentes no mercado brasileiro, não se tratou de uma opção recorrível no processo de design do circuito analógico. Explicitada essas questões, compreende-se a utilidade de se gerar, a partir das fontes projetadas anteriormente, as tensões de alimentação de $-2.8V$ e $12V$ no lugar de simplesmente utilizar as tensões fornecidas pelo microcontrolador em conjunto com o referencial ($0V$ e $5V$). Além disso, essa forma de alimentação terá uma implicação prática de interesse no sub-sistema subsequente: em situações de falta na energia proveniente do trafo e conseqüentemente de entrada nula no filtro passa-baixas, a fonte de tensão que continuaria em funcionamento (alimentação negativa) tenderia a forçar a saída do amp-op para valores negativos, os quais seriam lidos pelo microcontrolador como $0V$ contínuos de forma que seria possível identificar momentos de falha no fornecimento de energia pela tomada a qual o trafo está conectado - mas isso será melhor explorado mais a frente neste texto.

Ganho

Dado o principal objetivo do condicionamento de reduzir o sinal de tensão para leitura em um conversor A/D, é preciso que seja feita a escolha adequada dos valores de resistência $R1$ e $R2$ da Figura 2.7 de forma a se obter um novo sinal com a amplitude desejada. Esta, levando em consideração os limites de leitura do conversor presente no *chip* ATmega328P ($0V$ a $5V$ [39]), foi definida como $2V_{pk}$ ou $4V_{pp}$, deixando uma margem de segurança de $0.5V$ acima e abaixo do sinal diminuído ou $1V$ no total. Com essa especificação e levando em consideração o sinal de entrada com amplitude $16.97V_{pk}$ vindo da primeira espira secundária tem-se, a partir da Equação 2.4 característica do comportamento inversor do amp op, que a proporção $\frac{R2}{R1}$ dos resistores deve ser próxima a 0.12 em magnitude.

$$\frac{R2}{R1} = -\frac{V_{pk_s}}{V_{pk_e}} = -\frac{2V}{12\sqrt{2}V} \simeq -\frac{2}{16.97} \simeq -0.12 \quad (2.4)$$

Ao avaliar diversas possíveis combinações de resistências comerciais concluiu-se que o par cuja proporção mais se aproxima do valor calculado envolve resistências de valores nominais $3.9k\Omega$ e $33k\Omega$, tais que apresentam uma proporção de 0.12 entre si - no caso evitou-se o uso de três ou mais resistores configurados em série/paralelo por questões de custo e complexidade. Assim, teoricamente espera-se gerar um sinal senoidal defasado em $180^\circ(\pi)$ com amplitude perto de $2.01V_{pk}$ ou $4.02V_{pp}$, de acordo com a Equação 2.5.

$$V_{pk_s} = -\frac{R2}{R1}V_{pk_e} = -\frac{3.9k\Omega}{33k\Omega}12\sqrt{2}V \simeq -2.01V \quad (2.5)$$

Frequência de corte

Definidos os valores resistivos a serem usados nos componentes R1 e R2, o próximo componente físico a ter seu valor nominal determinado é o capacitor C1 encontrado no topo da Figura 2.7. Lembrando o funcionamento deste em conjunto com a resistência R2 para a caracterização do filtro em questão, tem-se a Equação 2.6 tal que fornece uma forma de se projetar o passa-baixas com a frequência de corte requerida a partir dos valores de capacitância e resistência referidos.

$$f_c = \frac{1}{2\pi R_2 C} \quad (2.6)$$

Uma vez que o valor de R2 já foi determinado anteriormente, é limitado ao valor de C1 a definição da frequência de corte da filtragem e assim, foi escolhida um **capacitor de cerâmica** com valor nominal $100nF$, resultando em uma frequência de corte por volta dos $408.09Hz$, como descrito na Equação 2.7.

$$f_c = \frac{1}{2\pi 3.9k\Omega 100nF} = 408.09Hz \quad (2.7)$$

Para chegar nessas especificações do componente capacitivo, algumas considerações foram feitas. Primeiramente, o material escolhido reflete a necessidade de processamento de um sinal com meio período de tensões negativas - o que excluí, por exemplo, o uso de capacitores polarizados como eletrolíticos e de tântalo - além da maior disponibilidade desse tipo de componente no mercado nacional e da preferência por um corpo que ocupe menor espaço, por um preço mais barato. Em um segundo momento, especificando a frequência do 6º harmônico do sinal de referência (proveniente da tomada) como sendo a maior frequência de interesse para o sinal condicionado - isto é, $6 \cdot 60Hz = 360Hz$ - optou-se por utilizar a maior capacitância nominal disponível comercialmente para essa classe de capacitores ($100nF$) de forma a obter uma frequência de corte um pouco acima dessa frequência limite, fixado o valor de R2. A partir dessas observações e aliadas à necessidade de se filtrar possíveis sinais de interferência vindos do *timer* na etapa de

alimentação negativa - o qual, como mencionado opera na faixa dos quilohertz - e outros ruídos indesejados de maior frequência sem maiores distorções nas frequências próximas de $60Hz$, avalia-se que frequência de corte calculada para tal limite capacitivo adequa-se bem às necessidades do projeto. Indo além, essa filtragem se mostrará mais necessária ainda, dado que atenuará a sobreposição de espectros de frequência no momento da amostragem do sinal pelo conversor A/D, cuja faixa de operação será configurada na faixa de $1kHz$, ou seja, o filtro projetado também agirá como um filtro anti-*aliasing*.

Offset

Paralelamente às questões de diminuição de tensão e filtragem, a outra exigência definida para o condicionamento bem-sucedido do sinal é a soma DC de um valor constante $V_{DC} = 2.5V$ ao sinal de saída analógica por meio do qual se garante a variação desta dentro do intervalo de leitura definido - $0V$ a $5V$. Também é importante observar que a escolha do valor V_{DC} desejado não só limita o sinal refinado de $2V_{pk}/4V_{pp}$ ao intervalo definido mas também coloca-o centralizado nessa faixa, de forma que o equivalente à uma voltagem $0V$ no sinal original da tomada é igual a $2.5V$ no sinal condicionado. Pela mesma lógica, também é possível inferir que valores de $\pm 311.13V$ no sinal original da tomada equivalerão aproximadamente aos valores $4.5V$ e $0.5V$ ($2V_{pk}$) no sinal condicionado e que os valores de $5V$ e $0V$ ($2.5V_{pk}$) no sinal condicionado equivalerão aos valores $\pm 388.91V$ no sinal original da tomada (de acordo com a relação de proporcionalidade na Equação 2.8). Isso significa que o sistema deverá ser capaz de observar valores de tensão até 25% maiores em amplitude com relação à situação em uma tomada ideal de $220V_{rms}$.

$$\frac{2V}{2.5V} \simeq \frac{311.13V}{388.91V} \quad (2.8)$$

Como mostrado na Figura 2.7, a adição desse deslocamento foi feito por meio da entrada não-inversora do amplificador operacional utilizado. Considerando a atuação diferencial desse dispositivo com relação às entradas inversora e não-inversora, descreve-se de forma simplificada (desconsiderando a ação capacitiva de $C1$) o valor de tensão desejado na entrada não-inversora V^+ para se obter um deslocamento V_{DC} por meio da Equação 2.9.

$$V^+ = \frac{V_{DC}}{\frac{R_2}{R_1} + 1} \quad (2.9)$$

Tendo fixado os valores de $R1$ e $R2$ previamente e considerando o deslocamento especificado, obtém-se então que a voltagem a ser aplicada na entrada não-inversora do amp-op tem de ser a mais próxima possível de $2.24V$ (de acordo com a Equação 2.10)

para que o *offset* também seja o mais próximo possível de $2.5V$. Deve-se enfatizar “ser o/a mais próximo(a) possível” justamente porque, de fato, tais voltagens não serão exatamente iguais aos valores mencionados, tanto por questões de projeto (abordadas logo em seguida) como pelas esperadas imperfeições dos componentes que constituirão o sistema analógico real e causarão algumas pequenas variações nos valores efetivamente medidos. Trata-se de uma situação válida também para a amplitude obtida após a etapa de ganho como descrito na Equação 2.5, a qual além de exceder o valor desejado de $2V$ em $5.61mV$ também sofrerá destas outras pequenas variações. São ressalvas importantes pois terão maiores implicações nos sub-sistemas subsequentes, que serão responsáveis por dar sentido físico aos valores lidos das entradas analógicas - isto é, serão responsáveis por atribuir um certo valor de tensão na tomada aos valores condicionados pelo circuito e discretizados pelo conversor analógico-digital.

$$V^+ = \frac{2.5V}{\frac{3.9k\Omega}{33k\Omega} + 1} = 2.24V \quad (2.10)$$

Isolamento

Por fim, é interessante ressaltar o papel de isolar a carga conectada à saída do filtro (no caso, a entrada do conversor A/D) de forma que, independentemente do valor de tal impedância, o circuito forneça as mesmas tensões e variações de tensão ao longo do tempo. Esse fenômeno se deve a certas características do *op-amp* como **alta impedância de entrada e baixa impedância de saída**, permitindo então que as características desejadas para o sinal de saída do circuito sejam mais ou menos constantes para diferentes conversores de um mesmo modelo ou até de modelos diferentes. Além disso, tal isolamento em conjunto com o diodo Zener apresentado em seguida fornecerá mais uma etapa de proteção às entradas analógicas do dispositivo de leitura, de forma a evitar eventuais perdas ou danos ao equipamento.

2.1.6 Deslocamento DC

Projetado o filtro passa-baixas com as devidas alimentações e sinal de interesse já diminuído pela etapa inicial de abaixamento (transformador), resta apenas uma última peça essencial para o funcionamento correto do circuito: um **divisor de tensão** capaz de gerar a voltagem a ser aplicada na entrada não-inversora do amplificador operacional a qual irá gerar o deslocamento $V_{DC} = 2.5V$ desejado para leitura (sendo esta aproximadamente $2.24V$, como já discutido). Tal modelo é apresentado na Figura 2.8 a seguir.

Tratando-se de um divisor de tensão simples com apenas dois resistores - novamente, evitou-se o uso de um número maior de componentes para manter o custo e a complexidade

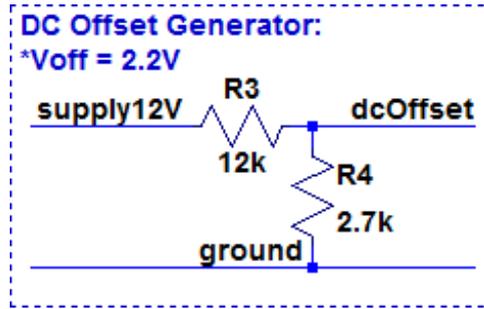


Figura 2.8: Modelo do divisor de tensão adotado.

do projeto relativamente baixos - e tendo como entrada os $12V$ constantes fornecidos pela fonte de alimentação positiva projetada (preferiu-se esta à fonte $5V$ CC do Arduino para evitar quaisquer problemas com a menor capacidade de fornecimento de corrente da placa) é possível obter facilmente a proporção dos valores resistivos partindo da Equação 2.11.

$$\frac{R4}{R3} = \frac{V^+}{V_{supply12} - V^+} = \frac{2.24V}{12V - 2.24V} = 0.23 \quad (2.11)$$

Os valores comerciais mais comuns e que melhor aproximam essa relação são de $12k\Omega$ para $R3$ e $2.7k\Omega$ para $R4$. Disso, obtém-se que a tensão de saída do divisor aplicada à entrada não-inversora deve ser próxima a $2.20V$, ou seja, na verdade $V^+ = 2.20V$. Combinando esse resultado com a Equação 2.9 rearranjada na Equação 2.12 é possível deduzir que o valor teórico esperado para o deslocamento DC final deve ser por volta de $2.47V$.

$$V_{DC} = V^+ \left(\frac{R2}{R1} + 1 \right) = 2.20V \left(\frac{3.9k\Omega}{33k\Omega} + 1 \right) = 2.47V \quad (2.12)$$

O fato de o deslocamento obtido teoricamente ($2.47V$) a partir destes valores nominais de resistência não ser exatamente o valor desejado ($2.5V$) retoma o ponto levantado na discussão da Equação 2.10 a respeito da capacidade do próximo sub-sistema de interpretar corretamente os valores de tensão lidos na saída do sub-sistema analógico - aspecto que será abordado mais a fundo nas secções seguintes.

2.1.7 Proteção à entrada analógica

Abordados os aspectos mais essenciais do funcionamento do circuito analógico projetado, parte-se agora para uma breve discussão relativa à proteção da interface analógica-digital, a qual faz parte do sub-sistema digital. Como já apresentado, esta interface é capaz de quantizar valores de tensão entre $0V$ e $5V$ de acordo com a especificação fornecida pelo fabricante [39], implicando que voltagens muito fora desta faixa permitida podem causar

danos ao equipamento dependendo da magnitude e do tempo de exposição a esses valores. Com isso em mente, e lembrando que a alimentação fornecida ao amplificador operacional ($-2.8V$ e $12V$) - o qual precede a entrada analógica - permite uma variação na sua saída de $-1.3V$ a $10.5V$ aproximadamente (considerando um pior caso, com a saturação em $\pm 1.5V$ com relação às alimentações [37]), verifica-se que é interessante a inclusão de um elemento diodo em paralelo com o pino de saída do amp-op de forma que ajude evitar que o sinal na entrada analógica fuja à faixa de tensão permitida. Isso porque, ainda que o circuito tenha sido projetado para operar dentro desta faixa, comportamentos transitórios e outras anormalidades podem levar a picos/vales de tensão que fogem ao intervalo permitido. Feitas essas considerações, o componente escolhido, um **diodo Zener** modelo 1N4733 posicionado como ilustrado na Figura 2.9, permite a passagem de tensões na faixa de $-1V$ a $5.1V$ de acordo com os valores nominais especificados no *datasheet* [40], aproximando consideravelmente o limite positivo de entrada ao valor desejado de $5V$ e ligeiramente reduzindo a faixa negativa de entrada. Trata-se de um uso prático das características não-lineares do componente utilizado.

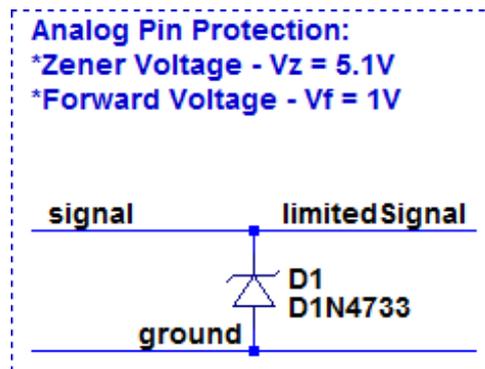


Figura 2.9: Diodo Zener de proteção.

2.1.8 Entrada Analógica

Por último, tem-se ao final do “caminho” a entrada analógica do conversor A/D sob a qual os valores de tensão serão amostrados e discretizados. Como já mencionado, em função das características próprias da saída do amp-op - no caso uma baixa impedância nesta - o sinal que chega ao elemento sensor do sub-sistema digital deve ser praticamente indiferente a variações da impedância de entrada, desde que esta seja consideravelmente maior do que a impedância de saída na etapa anterior (evitam-se efeitos de carregamento desta forma). Nesse contexto e de acordo com a documentação relativa ao ATmega328P, é indicado que o conversor em questão é otimizado para sinais analógicos em impedâncias de saída de até $10k\Omega$ aproximadamente - bem acima do valor baixo de saída do amplificador -

e é fornecido um valor típico para sua impedância de entrada de $100M\Omega$ [39]. Levantados esses pontos, segue na Figura 2.10 a modelagem da entrada analógica A0 que será ligada ao condicionador descrito.

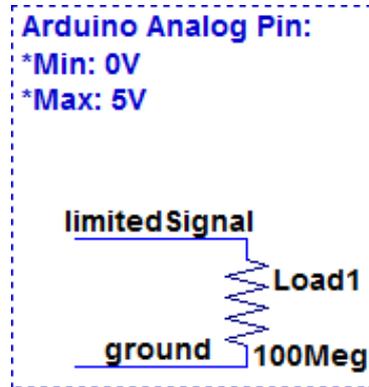


Figura 2.10: Modelo adotado para a entrada de leitura analógica do Arduino..

Com isso, segue na Figura 2.11 o esquemático completo do circuito projetado no ambiente de simulação LTSpice, de onde serão gerados resultados para o capítulo seguinte:

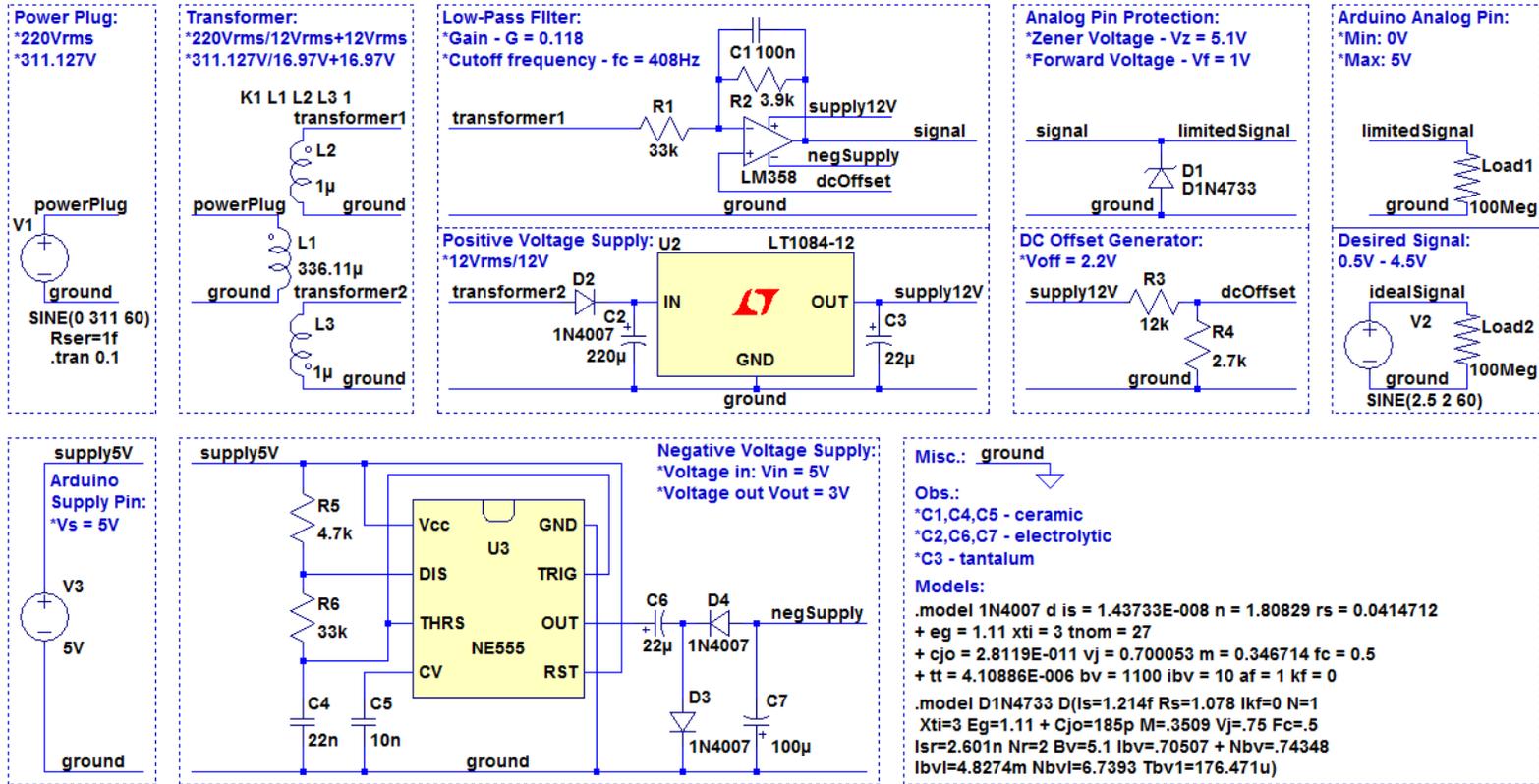


Figura 2.11: Circuito condicionador do subsistema analógico.

2.2 Sub-sistema digital

Seguindo o roteiro formulado com a apresentação do sub-sistema digital, a presente seção discutirá os códigos desenvolvidos (em linguagem C++ para o Arduino e C para o servidor em laboratório) para realização da integração com a parte analógica, dos cálculos referentes aos parâmetros de qualidade energética desejados, da comunicação serial com o monitor de falhas executado pelo servidor monitorado e das interfaces (tanto relativa ao ponto de falha como à rota de segurança) com o sub-sistema de redes. Acompanhando em um primeiro momento a estrutura básica de um programa para o Arduino (que neste trabalho opera pelo paradigma de programação em *loop* infinito, sem o uso de um sistema operacional próprio, por meio das declarações *setup* e *loop*, como na Figura 2.12) e finalizando com o algoritmo desenvolvido para o recebimento de informações pelo cabo *USB* por parte do computador pessoal, prosseguem as sub-seções:

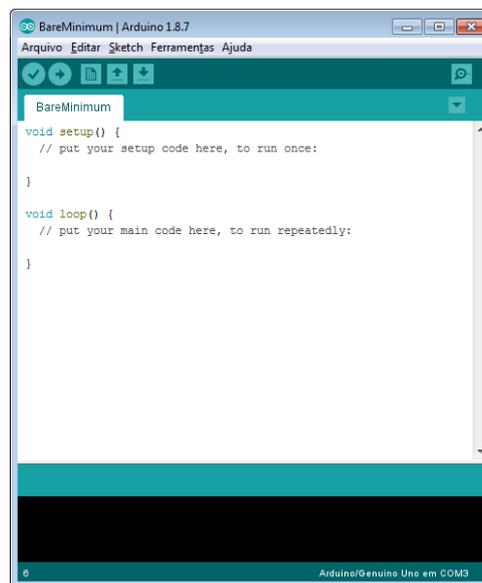


Figura 2.12: Ambiente de desenvolvimento Arduino com estrutura básica de programa.

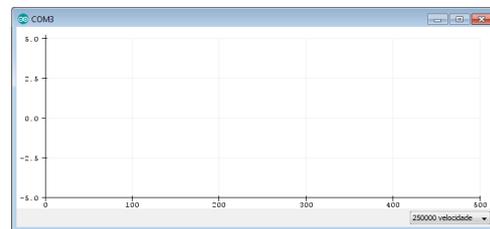
2.2.1 Bibliotecas, Diretivas e Variáveis

As primeiras linhas do programa desenvolvido para a placa Arduino, além de fazer a inclusão das bibliotecas que serão usadas - para comunicação serial (**SPI.h**), utilização do hardware de *ethernet* (**Ethernet.h**) e chamada a funções do protocolo MQTT utilizado (**PubSubClient.h**) - usam de diretivas de compilação condicional (*define*) para determinar a habilitação ou não de certas ferramentas de *debugging* e perfis de rede antes da execução em si do código:

- **DEBUG_MONITOR**: Se definido, o programa enviará informações textuais importantes pelo canal serial para serem exibidas - mais especificamente no Monitor serial do ambiente de desenvolvimento do Arduino (Figura 2.13(a));
- **DEBUG_PLOTTER**: Se definido, o programa enviará informações gráficas importantes pelo canal serial para serem exibidas - mais especificamente no *Plotter* serial do ambiente de desenvolvimento do Arduino (Figura 2.13(b));
- **LOCAL_SERVERS**: Se definido, o *broker* ao qual o sub-sistema digital se conectará não será o padrão remoto, presente no endereço **test.mosquitto.org**, mas sim o *broker* de um servidor local pré-estabelecido.



(a) Monitor Serial.



(b) *Plotter* Serial.

Figura 2.13: Par de ferramentas disponíveis no ambiente de desenvolvimento do Arduino.

Assim, serão indicados em pontos específicos desta secção a inclusão ou não de certos trechos de código por meio de um asterisco (*), dependendo das configurações habilitadas.

2.2.2 *Setup* inicial

Ao iniciar a execução do código C++ e declaradas as variáveis globais que serão usadas, o programa procede com a fase de *setup()*, função chamada uma única vez que irá tratar da ativação de artefatos iniciais em tempo de execução. Simplificadamente, esta fase consistirá na:

1. Inicialização de *LEDs* e botões;
2. Inicialização da comunicação serial com o computador pessoal ao qual o Arduino está conectado por meio da chamada `Serial.begin(256000)` - indicando que o parâmetro de *BaudRate* da comunicação em questão será de `256000bps` (mais detalhes serão apresentados quando discutido o monitor de falhas);
3. Listagem da configuração de amostragem no *console* do *PC* usando o protocolo serial;

4. Inicialização da conexão de rede com o Arduino *shield* por meio da chamada `Ethernet.begin(mac, ip)` - interrompendo indefinidamente a execução do programa no caso do *hardware* de *ethernet* não ter sido encontrado - e listagem da configuração de conexão no *console* do *PC*.

2.2.3 Operação em *loop*

Finalizado o *setup* inicial do programa, este entra no estado de *loop()*, função que prende a execução do código em um laço infinito enquanto o sub-sistema se encontrar energizado e não houverem interrupções provenientes do botão de *reset* na placa. Dentro dessa função serão realizadas as tarefas requeridas para a concretização dos objetivos específicos levantados para a parte digital do projeto, sendo tais tarefas divididas em três diferentes etapas de processamento: a etapa de digitalização (cujo propósito é a obtenção dos pacotes de dados para as tensões e correntes lidas), a etapa de caracterização (cujo propósito é deduzir o estado de funcionamento do sistema com relação à presença de energia na rede elétrica monitorada e de acesso à internet pelo dispositivo Arduino) e a etapa de comunicação (cujo propósito é gerar e encapsular as informações pertinentes em mensagens a serem transmitidas pelo sub-sistema de rede). Com isso em mente, disserta-se a seguir sobre o funcionamento dessas três etapas de maneira resumida:

Funcionamento da etapa de digitalização

Descreve-se a primeira etapa executada do *loop* no Algoritmo 1 a seguir, cujo comportamento é regido por certas variáveis relativas à digitalização.

A informação gerada da tensão máxima do pacote será particularmente útil nas próximas etapas, uma vez que será decisiva para a interpretação da presença ou não de energia elétrica na tomada monitorada. Considerando o projeto do sub-sistema analógico feito, o programa entenderá que um pacote contendo o byte de tensão máximo com valor 0 (dentre possíveis 256 valores de 0 a 255) indica uma falha no fornecimento de energia para o sistema - pois como já discutido, a saída do circuito de condicionamento projetado sem alimentação proveniente da tomada tenderá a ser lida como um sinal de 0V constantes pelo Arduino.

Funcionamento da etapa de caracterização

Com isso, a segunda etapa está descrita no Algoritmo 2 e define o estado de operação do sistema em função da capacidade do microcontrolador de se conectar à Internet por meio do periférico acoplado e também em função do estado da rede elétrica a qual o protótipo desenvolvido está conectado.

Algorithm 1 Algoritmo de digitalização

```
1: procedure DIGITALIZAÇÃO
2:   loop:
3:   if tempo decorrido desde a última leitura = período de amostragem then
4:     instante da última leitura  $\leftarrow$  tempo atual;
5:     amostra tensão  $\leftarrow$  A0 e corrente  $\leftarrow$  A1;
6:     pckV/I[contador]  $\leftarrow$  tensão/corrente (mapeia de 10 bits para um byte);
7:     if valor de tensão lida > maior valor lido até o momento then
8:       tensão máxima do pacote: maxPckV  $\leftarrow$  valor de tensão lida;
9:     contador  $\leftarrow$  contador + 1;
10:    faz o plot da tensão lida na entrada A0 do Arduino; *DEBUG_PLOTTER
11:    if pacotes preenchidos por completo then
12:      goto caracterizacao;
13:      reset:
14:      espera 5 segundos (mSecDelayPck/1000) para a próxima etapa de digitali-
15:      zação;
16:      tensão máxima do pacote  $\leftarrow$  0;
17:      contador  $\leftarrow$  0;
18:    goto loop;
```

Algorithm 2 Algoritmo de caracterização

```
1: procedure CARACTERIZAÇÃO
2:   caracterizacao:
3:   if cliente desconectado then
4:     p/ servidor local, tenta conexão MQTT com autenticação; *LOCAL_SERVER
5:     p/ servidor remoto, tenta estabelecer conexão MQTT;
6:   if conexão bem-sucedida then
7:     if tensão máxima do pacote > 0 then
8:       goto ok;
9:     else
10:      goto np;
11:   else
12:     if tensão máxima do pacote > 0 then
13:       goto ni;
14:     else
15:       goto ne;
```

Os quatro possíveis estados derivados desses critérios estão dispostos na Tabela 2.1, com **OK** indicando o funcionamento normal do sistema, **NP** (*Not Plugged*) indicando que o *plug* do sistema não está ligado à nenhuma tomada para monitoramento, **NI** (*No Internet*) indicando que o acesso a internet não foi possível por meio da placa Arduino e **NE** (*No Energy*) indicando que houve uma falha no fornecimento de energia pela rede elétrica monitorada. É interessante reforçar aqui que o estado NE, apesar de insinuar

Tabela 2.1: Tabela de possíveis estados do sistema.

		Acesso à Internet	
		Sim	Não
Circuito Energizado	Sim	OK	NI
	Não	NP	NE

que não haveria energia disponível para alimentar o servidor e por conseguinte o próprio microcontrolador e circuito analógico que monitoram a rede elétrica, não necessariamente implicará que o sistema de monitoramento está desenergizado, pois no caso de uma falha na rede elétrica o *no-break* que intermedeia a alimentação desse servidor altera automaticamente seu modo de operação para a configuração bateria, ainda que durante um curto período de tempo.

Funcionamento da etapa de comunicação

Por fim, a terceira e última etapa descrita pelo Algoritmo 3 irá gerar as informações necessárias para cada caso de funcionamento do sistema baseadas nos dados de tensão e corrente obtidos na primeira etapa.

Pode-se observar que, para o estado de funcionamento OK descrito, o parâmetro de qualidade de energia relativo à tensão de pico do sinal já havia sido obtido como parte das etapas de processamento anteriores, restando o cálculo dos parâmetros de tensão de vale (obtido de forma semelhante à tensão de pico), valor eficaz e frequência média do sinal. Nesse sentido, retoma-se a discussão relativa ao deslocamento DC iniciada na secção anterior - sendo esta sobre a forma como será relacionado o código do byte de tensão lida pelo conversor A/D ao respectivo valor de tensão na entrada analógica do conversor e consequentemente ao valor de tensão na tomada da rede monitorada. Isso porque, para o cálculo dos parâmetros de valor eficaz e frequência média, é necessário que seja definido um valor de byte que seja equivalente à tensão média do sinal condicionado (idealmente $2.5V$) e equivalente à tensão média do sinal proveniente da rede elétrica (teoricamente $0V$) simultaneamente. Assim, feitas as devidas considerações sobre o projeto do valor médio do sinal condicionado - na teoria realizado como $2.46V$ - o valor de byte escolhido para refletir essas tensões médias foi de 127 em representação decimal.

A partir desse valor médio de byte foi possível então o cálculo do valor eficaz do sinal digitalizado por meio da fórmula demonstrada na Equação 2.13 - na qual N representa o número de amostras disponíveis e V_i representa o valor não quantizado da i -ésima amostra. Nota-se que essa equação mostra na realidade um relação de proporcionalidade entre o valor RMS calculado e expresso em um byte ($vRMS$) e o valor RMS do sinal amostrado

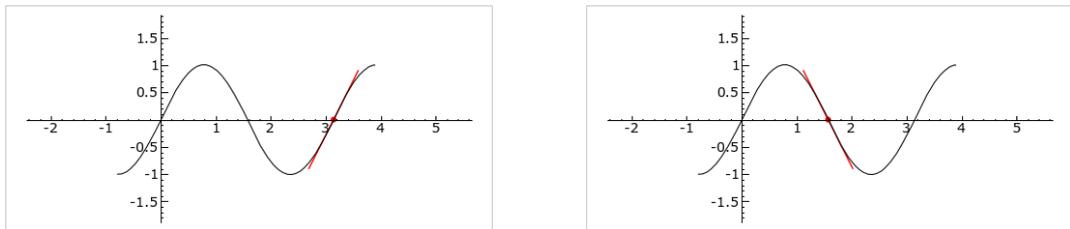
Algorithm 3 Algoritmo de comunicação

```
1: procedure COMUNICAÇÃO
2:   ok:
3:   prepara variáveis auxiliares para o cálculo das informações;
4:   for all amostras em pckV[] percorridas por contador do
5:     vMS ← vMS + (pckV[contador]-127)2/nMaxBytesPck;
6:     if valor de tensão lida < menor valor lido até o momento then
7:       tensão mínima do pacote: minPckV ← valor de tensão lida;
8:     if detectada inflexão positiva entre duas amostras consecutivas da onda then
9:       índice do último ponto de inflexão positiva da onda: LI ← contador;
10:    if índice do primeiro ponto de inflexão positiva ainda não encontrado then
11:      índice do primeiro ponto de inflexão positiva: FI ← contador;
12:    else
13:      número de inflexões após primeiro ponto: NI ← NI + 1;
14:    tensão RMS do pacote: vRMS ← √vMS;
15:    frequência média: freq ← 1/((LI - FI)/NI)*uSecSampPer);
16:    mensagem: msg ← mSecDelayPck/1000, freq, vRMS, maxPckV, maxPckI, pckV, pckI;
17:    envia a mensagem MQTT pela rota de funcionamento normal;
18:    imprime um aviso indicando o sucesso ou não do envio; *DEBUG_MONITOR
19:    acende LED para indicar o envio das informações;
20:    goto back;
21:
22:   np:
23:   envia alerta “NP” MQTT pela rota de funcionamento normal;
24:   imprime um aviso indicando o sucesso ou não do envio; *DEBUG_MONITOR
25:   goto not_ok;
26:
27:   ni:
28:   instrui o monitor de falhas a enviar um alerta “NI” pela rota de segurança;
29:   goto not_ok;
30:
31:   ne:
32:   instrui o monitor de falhas a enviar um alerta “NE” pela rota de segurança;
33:
34:   not_ok:
35:   apaga LED para indicar falha no envio das informações;
36:   back:
37:   goto reset;
```

da tomada/condicionado de fato (V_{RMS}), de forma que essa relação poderá ser usada no sub-sistema seguinte, na determinação do valor eficaz estimado para o sinal da tomada.

$$V_{RMS} = \sqrt{\frac{1}{N} \sum_{i=0}^N V_i^2} \propto \sqrt{\frac{nMaxBytesPck}{\sum_{i=0}^N} \frac{(pckV[i] - 127)^2}{nMaxBytesPck}} = vRMS \quad (2.13)$$

O cálculo da frequência média do sinal também partiu desse valor médio do sinal em byte, de forma que se observaram as transições de amostras com valores abaixo de 127 para amostras subsequentes com valor igual ou acima de 127. Essas transições foram interpretadas pelo programa como inflexões do sinal senoidal ocorridas entre duas amostras consecutivas - mais especificamente pontos de inflexão com derivada positiva (ou pontos de inflexão positiva, por simplicidade), como ilustradas na Figura 2.14(a). Para essas amostras consecutivas, o índice da segunda foi considerado o índice de inflexão positiva.



(a) Inflexão positiva e reta tangente em vermelho.

(b) Inflexão negativa e reta tangente em vermelho.

Figura 2.14: Representação gráfica dos pontos de inflexão numa curva senoidal [1].

Assim, a partir de um conjunto de índices de inflexão positiva obtidos de um pacote de tensão foi possível estimar a frequência do sinal a partir da Equação 2.14 apresentada - na qual i_n representa o n -ésimo ponto de inflexão, τ representa o período de amostragem do sinal e M representa o número total de pontos de inflexão reconhecidos. É interessante observar também que essa equação mostra uma aproximação entre a frequência média calculada no programa e expressa em um byte ($freq$) e a frequência média do sinal real da tomada/condicionado ($f_{média}$), uma vez que os índices de inflexão não coincidirão de fato com os mesmos instantes de tempo dos respectivos pontos de inflexão no sinal analógico.

$$\begin{aligned}
 f_{média} &= \frac{1}{\left(\frac{(i_1 - i_0)\tau + (i_2 - i_1)\tau + \dots + (i_{M-1} - i_{M-2})\tau}{M - 1} \right)} \\
 &= \frac{1}{\frac{(i_{M-1} - i_0)\tau}{M - 1}} \simeq \frac{1000000}{\frac{(LI - FI)}{NI} uSecSampPer} = freq
 \end{aligned} \tag{2.14}$$

Dessa forma, com todos os dados obtidos e todas as informações calculadas, espera-se que ao longo do funcionamento normal do sistema (OK) sejam geradas mensagens com um tamanho total de 70 bytes: $1byte$ ($mSecDelayPck/1000$) + $1byte$ ($freq$) + $1byte$ ($vRMS$) + $1byte$ ($maxPckV$) + $1byte$ ($minPckV$) + $32bytes$ ($pckV$) + $32bytes$ ($pckI$) + $1byte$ ($\backslash 0$) = $70bytes(msg)$. Vale mencionar que a inclusão do primeiro byte contendo a informação do tempo de espera entre o envio de um pacote MQTT e o início da etapa de amostra-

gem seguinte é uma forma de comunicar aos clientes *subscribers* o tempo que terão para atualizar suas interfaces entre a chegada de mensagens subsequentes.

Já para os demais modos de funcionamento optou-se por enviar uma mensagem MQTT contendo dois bytes apenas informando aos *subscribers* sobre o estado do sistema, como pode ser verificado no trecho de estado NP do Algoritmo 3 e como será verificado para os estados NI e NE a seguir, no monitor de falhas. Neste quesito, é interessante dizer que a maneira como o programa carregado no microcontrolador instrui esse monitor de falhas a enviar um certo alerta é por meio do envio de um ou dois caracteres pelo canal serial, o qual conecta o servidor monitorado e a placa do Arduino. No caso da necessidade de se enviar um alerta "NI" aos *subscribers*, o programa em execução nessa placa mandará por esse canal o código de retorno obtido na tentativa inicial de se conectar com o *broker* MQTT - tal código variando entre inteiros de -4 a 5 (de forma que o código 0, que significa uma tentativa de conexão bem sucedida, nunca será enviado nesse estado do sistema). Já no caso da necessidade de se enviar um alerta "NE", o programa em execução nessa placa mandará por esse canal o código 0.

2.2.4 Monitor de Falhas

Como último tópico relativo ao sub-sistema digital tem-se o algoritmo do monitor de falhas, um programa desenvolvido para intermediar a comunicação entre o sub-sistema em discussão e o sub-sistema de rede (abordado a seguir) em casos que essa comunicação não seja possível de forma direta - por meio do roteador local (o ponto de falha). Tais casos ocorrem nas situações em que o acesso à Internet pelo *shield* do microcontrolador não é bem sucedido (isto é, NI ou NE), forçando esse programa a comunicar tal evento para que o monitor de falhas tome alguma providência, isto é, avisar aos clientes *subscribers* na rede sobre o estado atual do sistema. Como será apresentado na discussão relacionada ao sub-sistema de rede, o acesso à Internet por parte do computador que executa o monitor de falhas sempre estará disponível para que seja possível avisar qualquer falha no funcionamento da rede elétrica ou de Internet de forma indireta - por meio do mini modem 3G (a rota de segurança).

Partindo dessas ideias, o código utilizado no monitor em questão foi baseado em um programa implementado na linguagem C, publicado em 30 de novembro de 2014 pelo autor Rahul S. para um dos lados da comunicação serial [41] - no caso de um computador pessoal com sistema operacional Windows. O código original utiliza as bibliotecas **Win32 API** e **windows.h** para comunicação com a porta serial, recebendo um vetor de caracteres. De forma resumida, seu funcionamento consiste em abrir uma conexão com uma porta COMxx por meio de uma função `CreateFile()`, definir os parâmetros da comunicação serial - como `BaudRate` (256000bps), formato dos dados trocados (8 bits

com 1 bit de parada e sem paridade) e controle de fluxo (não presente) - além dos *time-outs* leitura/escrita e configurar o Windows para monitorar a recepção de strings vindas do dispositivo serial, fazendo a espera por meio da função `WaitCommEvent()` e lendo os caracteres recebidos para um *buffer* (de tamanho 256 bytes) por meio de uma função de `ReadFile()`, imprimindo-os logo em seguida.

Este programa base foi então modificado de forma a se adequar às necessidades do projeto proposto - codificando-se um *loop* infinito que engloba desde a chamada `WaitCommEvent()` até a impressão das instruções recebidas de forma a habilitar a recepção de vários vetores de caracteres ao longo da execução e adicionando expressões condicionais para o reconhecimento da situação do sistema e ação adequada sobre cada uma das duas (NI ou NE). No caso da identificação de instruções indicando o envio de um alerta NI, o monitor de falhas faz uma chamada a outro programa (`mosquitto_pub`) para encaminhar o alerta: `system("C:\\ Windows\\System32\\cmd.exe /c \\\"mosquitto_pub -h test.mosquitto.org -p 1883 -i ac:monitor:modem:xxxxxxxxxxxxx -t voltageCurrentMonitor -m NI\\\")`. Já no caso da identificação de instruções indicando o envio de um alerta NE, o monitor de falhas faz uma chamada diferente a esse mesmo programa para encaminhar o alerta: `system("C:\\ Windows\\System32\\cmd.exe /c \\\"mosquitto_pub -h test.mosquitto.org -p 1883 -i ac:monitor:modem:xxxxxxxxxxxxx -t voltageCurrentMonitor -m NE\\\")`.

2.3 Sub-sistema de redes

Para finalizar a discussão relativa ao projeto do sistema, a presente secção irá tratar da estrutura de rede adotada, como apresentada na Figura 2.1, bem como das ferramentas (protocolo MQTT, Mosquitto Broker, mini modem 3G, GitHub *Host*) e bibliotecas (PubSubClient.h, mqttws31.js, highcharts.js, boost.js, exporting.js, jquery.min.js) utilizadas para a implementação dessa estrutura. Assim, seguindo mais uma vez a ideia de um “caminho”, desta vez percorrido pela mensagem MQTT proveniente do sub-sistema digital, desde um dos dois possíveis clientes *publishers*, passando pelo *broker* hospedado local ou remotamente, até um cliente *subscriber* qualquer (vide Figura 2.15):

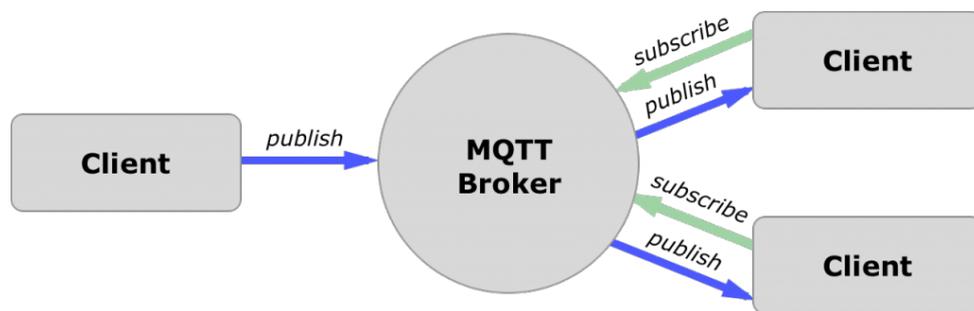


Figura 2.15: Diagrama demonstrando a estrutura e funcionamento em alto nível do protocolo MQTT (Fonte: [42]).

2.3.1 Publisher

No contexto do protocolo MQTT o *publisher*, como o cliente que estará efetivamente enviando mensagens para serem repassadas pelo *broker*, está presente em dois dispositivos: no conjunto Arduino mais *Ethernet Shield* - o qual, em situações de rede elétrica energizada, realiza o acesso à internet por meio do roteador local e de forma cabeada (o ponto de falha) - e no servidor que executa o programa do monitor de falhas, o qual em situações de falha no fornecimento de energia pela rede elétrica realiza o acesso à internet por meio de um **mini modem 3G** com cartão *SIM* pré-pago (a rota de segurança).

Envio pelo Ponto de Falha

A biblioteca utilizada na programação do Arduino para a realização da comunicação MQTT do lado cliente, a **PubSubClient.h** (desenvolvida por Nick O’Leary [43]), opera em conjunto com a biblioteca nativa utilizada na configuração do *shield* (**Ethernet.h**) e conta com procedimentos para o funcionamento do dispositivo tanto como *publisher* e/ou como *subscriber*, mas para o projeto realizado, que trata a troca de mensagens de

forma unidirecional, a comunicação feita pelo sub-sistema de redes dependerá apenas das funções para publicação das mensagens de estado OK e NP - sendo o tamanho destas limitado a 128 bytes na implementação original da biblioteca (passível de modificação). Vale mencionar a lista de códigos para possíveis estados do cliente, gerados na tentativa de conexão com o servidor e enviados (em sua maioria) via comunicação serial para o monitor de falhas em situações de NI:

- 4. MQTT_CONNECTION_TIMEOUT;
- 3. MQTT_CONNECTION_LOST;
- 2. MQTT_CONNECT_FAILED;
- 1. MQTT_DISCONNECTED;
- 0. MQTT_CONNECTED (nunca enviado ao monitor de falhas);
- 1. MQTT_CONNECT_BAD_PROTOCOL;
- 2. MQTT_CONNECT_BAD_CLIENT_ID;
- 3. MQTT_CONNECT_UNAVAILABLE;
- 4. MQTT_CONNECT_BAD_CREDENTIALS;
- 5. MQTT_CONNECT_UNAUTHORIZED;

Envio pela Rota de Segurança

De maneira semelhante, as ferramentas utilizadas pelo monitor de falhas para a realização da comunicação MQTT do lado cliente, o programa **mosquitto_pub.exe** em conjunto com a biblioteca dinâmica **mosquitto.dll**, também foram os únicos recursos necessários (dentro um conjunto de ferramentas maior, apresentado logo a frente) para concretizar os alertas previstos (NI e NE) levando em consideração a troca unidirecional de mensagens.

2.3.2 Broker

Dentre os diversos *message brokers* disponíveis *online* o **Eclipse Mosquitto**, escolhido para este projeto, trouxe a possibilidade de testar o seu funcionamento em um servidor remoto mantido pela própria Eclipse *Foundation* bem como a possibilidade de hospedar um *broker* localmente. A ferramenta Mosquitto em si contém um conjunto de programas para a publicação em um certo tópico (`mosquitto_pub -h [hostname] -p [port-number] -i [client-id] -t [message-topic] -m [message] -u [username] -P [password] ...`), inscrição em um certo tópico (`mosquitto_sub -h [hostname] -p [port-number] -i [client-id] -t [message-topic]`

-u [username] -P [password]] ...), geração de senhas criptografadas (mosquitto_passwd -U [passwordfile]) e execução do *broker server* em si (mosquitto - mosquitto.conf), configurado por uma variedade de opções em um arquivo separado.

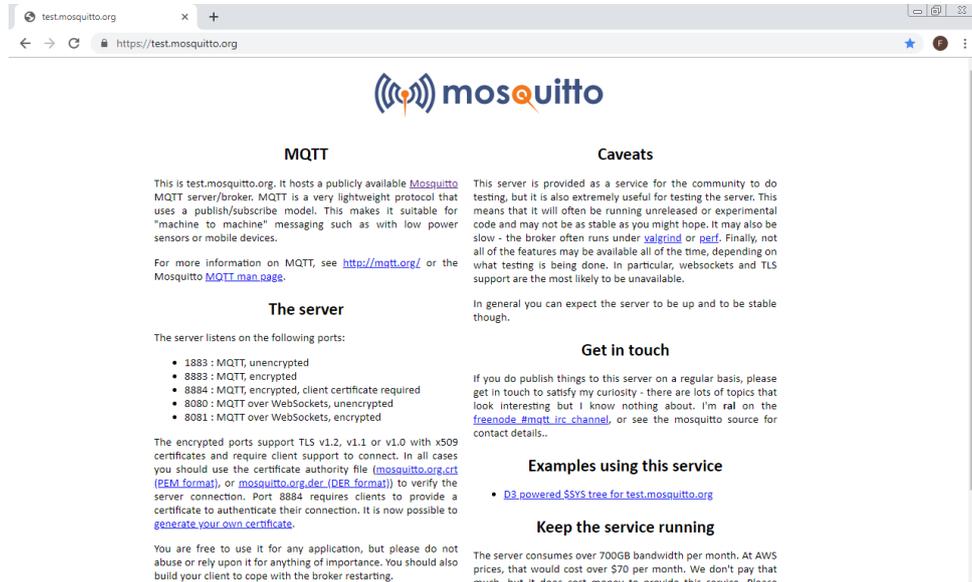


Figura 2.16: Site do broker de testes Mosquitto.

Dessa maneira, é possível ajustar as restrições de acesso ao sistema pretendidas (limitadas no escopo deste trabalho à verificação por usuário e senha) tanto com relação a quais *publishers* podem publicar conteúdo em um certo tópico como com relação a quais *subscribers* podem receber esse conteúdo encaminhado em um *broker* executado localmente, por um servidor para o qual se tenha acesso direto a máquina - mas não no servidor de teste disponibilizado publicamente no domínio “test.mosquitto.org”.

Ainda, como mostrado na Figura 2.16, o *message broker* escolhido para teste também provê diversas portas para escuta, cada qual implementando um diferente (ou nenhum) protocolo de segurança e dentre as quais utilizaram-se: *port* 1883 para o recebimento de dados não criptografados provenientes do *publisher* e *port* 8081 para o envio de dados criptografados aos *subscribers* por meio de *WebSockets* (API suportada pela maior parte dos *browsers* que acessam páginas utilizando protocolos de segurança tais como *HTTPS* e *SSL/TSL*).

2.3.3 Subscribers

Ao fim do “caminho” na estrutura de rede do protocolo utilizado encontram-se os *subscribers*, clientes que, no contexto do projeto apresentado, são representados por uma página *web* hospedada no domínio oferecido pelo serviço **GitHub Host**. Esta página possui um

frontend codificado em **HTML**, complementado por um *backend* implementado na linguagem **JavaScript** com auxílio de diversos módulos pertinentes a diferentes atividades:

Componentes da Página Web

- **mqttws31.js**: biblioteca relativa a comunicação pelo protocolo MQTT do lado de um cliente JavaScript executado em um *browser* (v3.1.1, Andrew Banks, 2013 IBM Corp.) [44];*
- **jquery.min.js**: biblioteca relativa a manipulação eficiente em código de elementos do *frontend* - HTML/CSS (v1.8.2, John Resig, 2012 The jQuery Team) [45];
- **highcharts.js**: biblioteca relativa a disposição dos dados recebidos em gráficos (v6.2.0, Torstein Honsi, 2018 Highcharts) [46];
- **exporting.js**: biblioteca relativa a realização da exportação dos gráficos ilustrados (v6.2.0, Torstein Honsi, 2018 Highcharts) [46];
- **boost.js**: biblioteca relativa a otimização na rederização das curvas no gráfico (v6.2.0, Torstein Honsi, 2018 Highcharts) [46].

* com modificações à função **parseUTF8**, por conflitos na codificação gerada no *publisher* (ASCII) e recebida no *subscriber* (UTF8).

Capítulo 3

Simulações Computacionais e Resultados Experimentais

Discutidas as questões relativas ao funcionamento do projeto e detalhes de implementação deste, apresenta-se na discussão aqui presente os resultados obtidos no desenvolver do sistema e no sistema final obtido - testado em um ambiente residencial controlado (vide Figura 3.1) por impossibilidades técnicas, institucionais e burocráticas relativas à rede a qual está conectado o laboratório que inspirou este trabalho (bloqueio de *firewall* por parte da instituição UnB). Assim, seguindo a fórmula adotada de apresentação dos sub-sistemas analógico, digital e de rede em ordem:

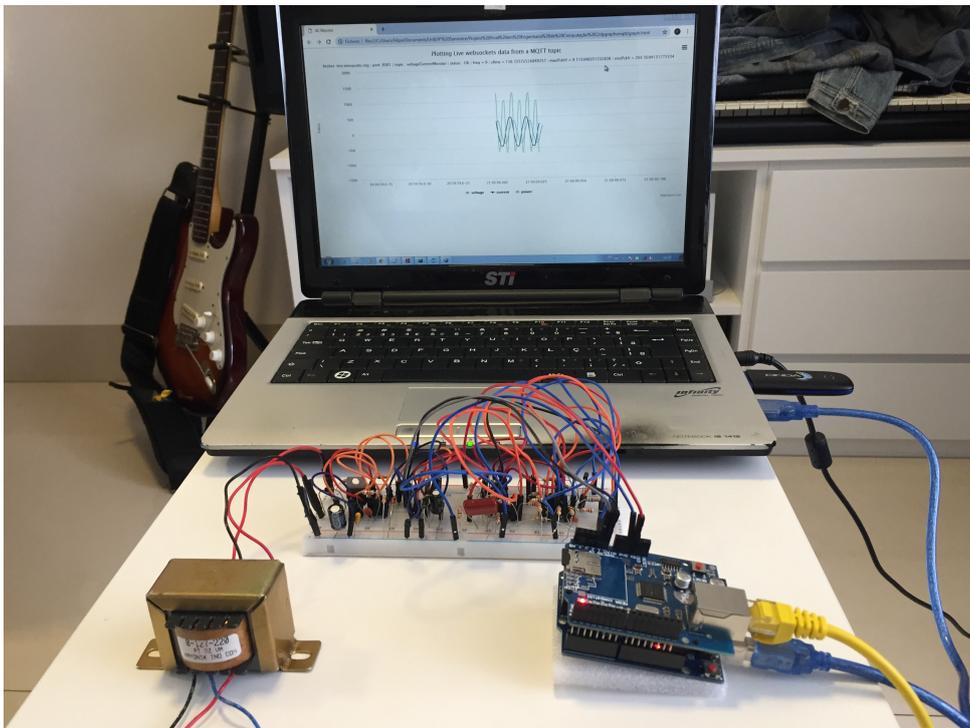


Figura 3.1: Sistema final emulado.

3.1 Sub-sistema analógico

Exibe-se nesta secção os resultado obtidos tanto de simulações como de experimentos conduzidos no Laboratório de Circuitos Elétricos e Eletrônicos da Universidade de Brasília, localizado no prédio SG-11 e, quando possível, uma comparação lado a lado entre ambos - de forma a salientar as principais semelhanças e diferenças entre as formas de onda geradas em cada caso para um mesmo sinal do sub-sistema.

Partindo dessa premissa, formaliza-se por meio de um resultado simulado aquela que a princípio deve ser a curva do sinal de entrada do cicuito analógico vindo da tomada: uma onda senoidal ideal com amplitude de $220V_{rms}/311.127V_{pk}$ e frequência em $60Hz$, como apresentada na Figura 3.2.

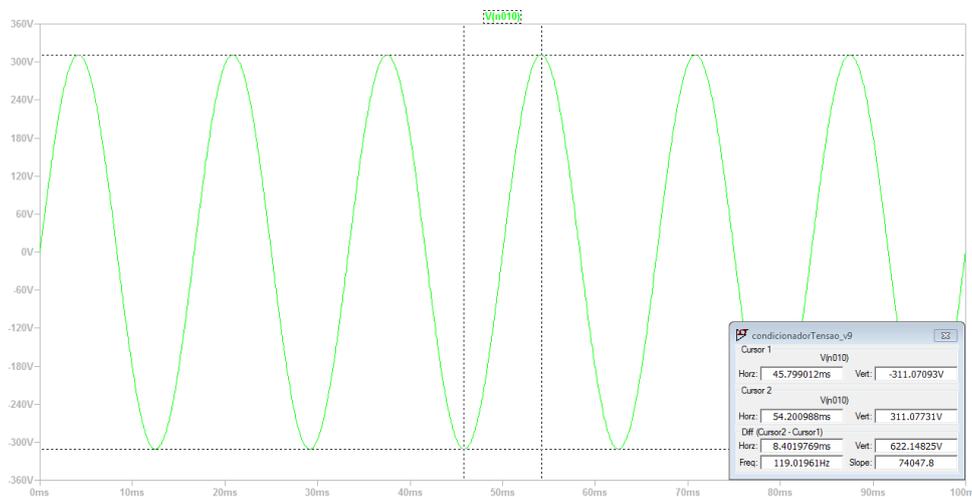
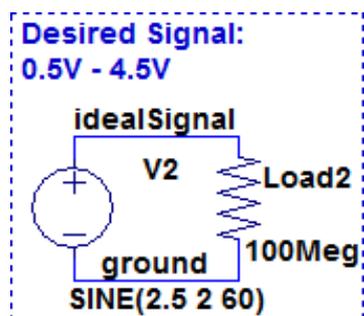


Figura 3.2: Curva simulada do sinal de entrada proveniente da tomada.

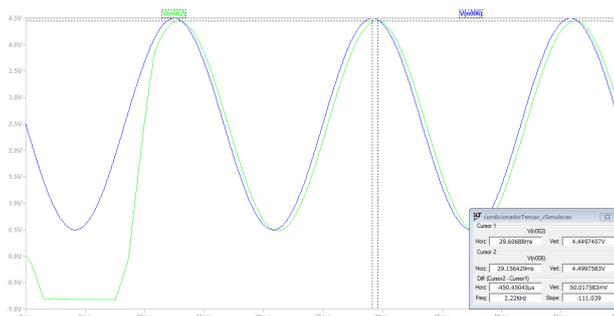
Segue então a formalização daquilo que a princípio deve ser a curva do sinal de saída desejada para o circuito projetado (modelada pelo esquemático da Figura 3.3(a)) sobreposta da curva do sinal de saída obtido no circuito simulado, ambas na Figura 3.3(b) - a primeira em azul e a segunda em verde.

Avaliada então de forma satisfatória a acurácia do circuito em ambiente de simulação, partiu-se para a implementação de um protótipo físico utilizando os componentes apresentados no segmento Analógico da Tabela 4.1. A montagem deste pode ser conferida na Figura 3.4, com o trafo transformador utilizado exibido em mais detalhes na Figura 3.5.

Como discutido, o trafo com tomada central escolhido possui três terminais de entrada (Figura 3.5(a)) - dois configurados de forma a gerar a tensão desejada de $12V_{rms}$ para uma entrada de $220V_{rms}$, deixando o *tap* central flutuando - e três terminais de saída (Figura 3.5(b)).



(a) Esquemático em simulação do sinal desejado



(b) Curva simulada do sinal desejado comparado à curva de saída do circuito projetado

Figura 3.3: Avaliação simulada do circuito projetado.

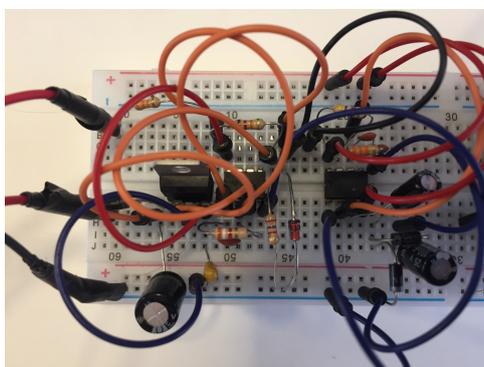
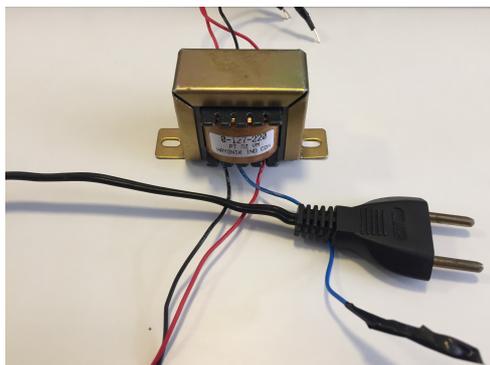
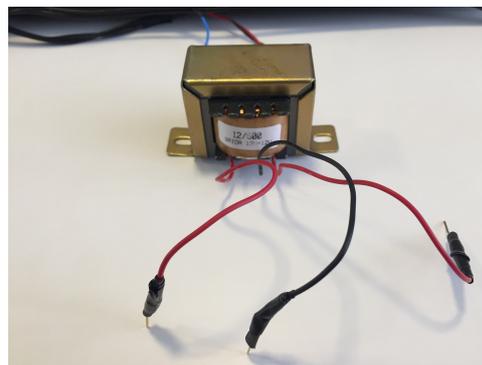


Figura 3.4: Protótipo implementado do circuito projetado.



(a) Terminais de entrada



(b) Terminais de saída

Figura 3.5: Trafo utilizado na implementação do sub-sistema.

A partir deste ponto expõe-se então os resultados simulados e experimentais propriamente ditos, estes representados pelos sinais de tensão em cada nó do circuito elaborado (quando aplicável) após um certo período do comportamento transitório - ou seja, considerando apenas o regime permanente das simulações computacionais.

3.1.1 Alimentação Positiva

Começando pela fonte de tensão positiva elaborada:

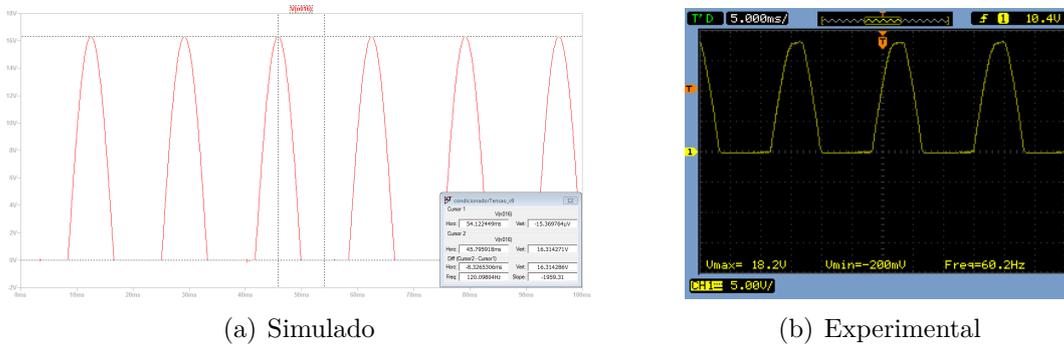


Figura 3.6: Gráficos obtidos para o diodo retificador de meia-onda.

Já é possível observar nos gráficos da Figura 3.6(b) distorções na amplitude da forma de onda pós-trafo, a qual apresenta um valor de pico máximo de $18.20V$ (e um valor mínimo de $-200mV$, podendo ser atribuído a algum problema de resolução do osciloscópio). Supõe-se que a operação desse trafo é aproximadamente linear para frequências por volta de $60Hz$ na rede elétrica e que as espiras primárias e secundárias são tais que a proporção entrada/saída esteja muito perto de $220/12$, verifica-se que tal sinal de tensão está cerca de 15% ($2.33V_{pk}$) acima do valor esperado em teoria de $15.87V_{pk}$ para uma rede elétrica $220V_{rms}$ (considerando a queda de tensão no diodo de $1.1V$).

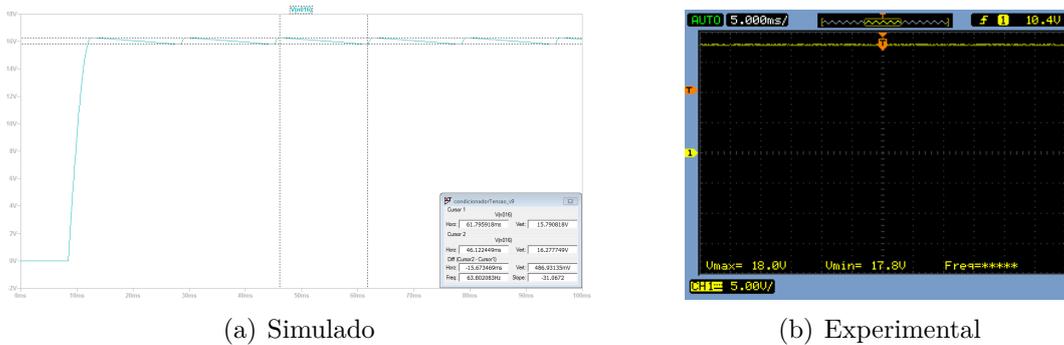


Figura 3.7: Gráficos obtidos para o diodo retificador de meia-onda seguido de um capacitor de filtro.

Tal observação também pode ser feita de forma semelhante na comparação da Figura 3.7, com valores de *ripple* simulados no intervalo de $15.79V$ a $16.28V$ (3.7(a)) e experimentais (3.7(b)) no intervalo de $17.80V$ a $18.00V$. Pode-se concluir então, somando as considerações feitas anteriormente à suposição de que a interferência do equipamento de medição (osciloscópios, cabos *BNC-Protoboard*, etc) é pequena comparada à influência

da instalação na sala, que a rede elétrica no laboratório de testes experimentais no momento da medição teria a amplitude de sua senoide cerca de $36.67V$ maior (isto é, seria de $347,79V_{pk}$ ou $245.93V_{rms}$) do que o ideal para uma rede de $220V_{rms}$ - feita a proporção com tensões acima em $2V_{pk}$ para as saídas do transformador considerado quase ideal.

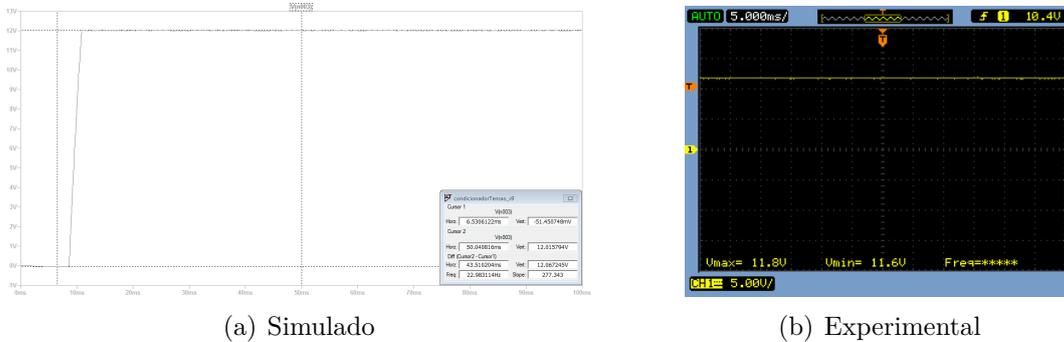


Figura 3.8: Gráficos obtidos para a saída do regulador de tensão com o capacitor de estabilização.

A ação do regulador de tensão com o capacitor de estabilização logo em seguida reduz drasticamente essa diferença entre os valores de tensão teóricos/simulados (muito próximos de $12V_{DC}$) com relação aos valores obtidos experimentalmente (variando entre $11.6V_{DC}$ e $11.8V_{DC}$) no próximo nó, como se constata da Figura 3.8 - gerando então a voltagem necessária de alimentação positiva.

3.1.2 Alimentação Negativa

Partindo para a fonte de alimentação negativa feita:

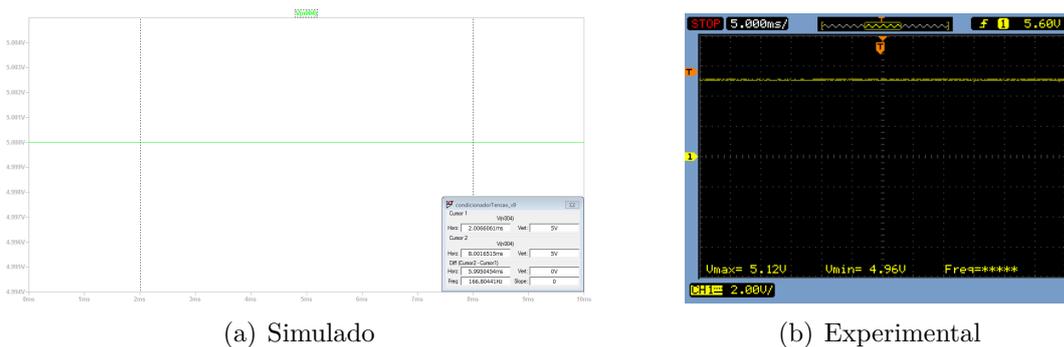
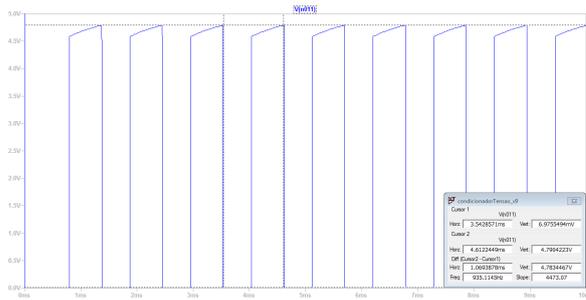
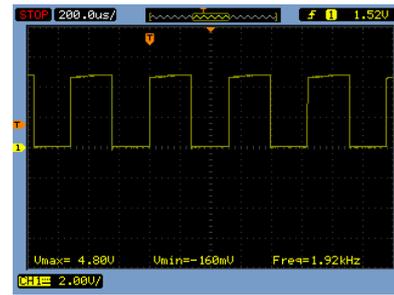


Figura 3.9: Gráficos obtidos para o pino Vcc e RST do *timer*.

Conectado aos pinos Vcc e RST do CI NE555 está o pino com alimentação contínua de $5V$ (teóricamente e em simulação - 3.9(a)) fornecida pelo Arduino ou pelo *Ethernet Shield* conectado ao Arduino. Experimentalmente, notam-se voltagens bem próximas a essa ($4.96V$ - $5.12V$, em 3.9(b)), ambos vistos na Figura 3.9.



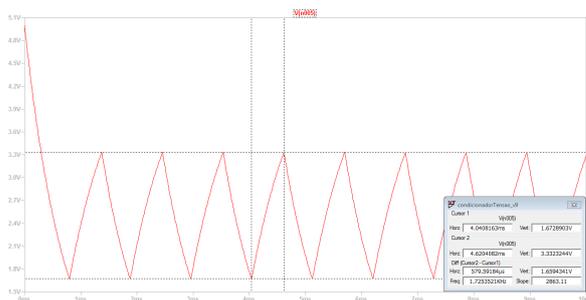
(a) Simulado



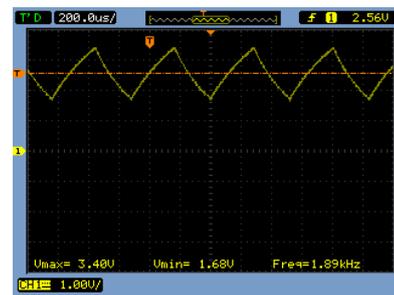
(b) Experimental

Figura 3.10: Gráficos obtidos para o pino DIS do *timer*.

Analisando a Figura 3.10 percebe-se a próxima grande diferença entre os sub-sistema simulado e o sub-sistema físico: no *software* de simulação utilizado a frequência de operação do modelo integrado NE555 foi estimada em $935.12Hz$ ou, por simplicidade, na faixa de $1kHz$, enquanto a frequência de operação do componente físico adquirido foi estimada em $1.92kHz$ ou na faixa de $2kHz$ - trata-se de uma tendência do sinal em DIS que se repetirá para outros pinos do CI em função de diferenças de implementação internas entre ambos modelos do componente. Apesar dessa diferença, poderá ser verificado que o circuito de alimentação negativa funcionará de forma aceitável, dado que a frequência do *timer* terá maior influência apenas no momento de transitório ou de inicialização do sub-sistema, não prejudicando a geração da tensão negativa necessária em regime permanente. Ademais disso, a forma de onda trapezoidal aparece em ambos casos e as faixas de tensão geradas estão relativamente próximas: de $6.98mV$ a $4,79V$ em simulação (3.10(a)) e de $-160mV$ (novamente atribuído a algum erro do osciloscópio) a $4.80V$ experimentalmente (3.10(b)).



(a) Simulado



(b) Experimental

Figura 3.11: Gráficos obtidos para o pino THRS e TRIG do *timer*.

A mesma proximidade na forma de onda e nos valores para faixas de tensão gerada se mantém no caso dos pinos THRS e TRIG conectados entre si (Figura 3.11) - tanto em simulação como experimentalmente surge nesse nó uma onda quase triangular, com

variações entre $1.67V$ e $3.33V$ a $862.67Hz$ (estimado) para o primeiro caso (3.11(a)) e entre $1.68V$ e $3.40V$ a $1.89kHz$ (medido) no segundo (3.11(b)).

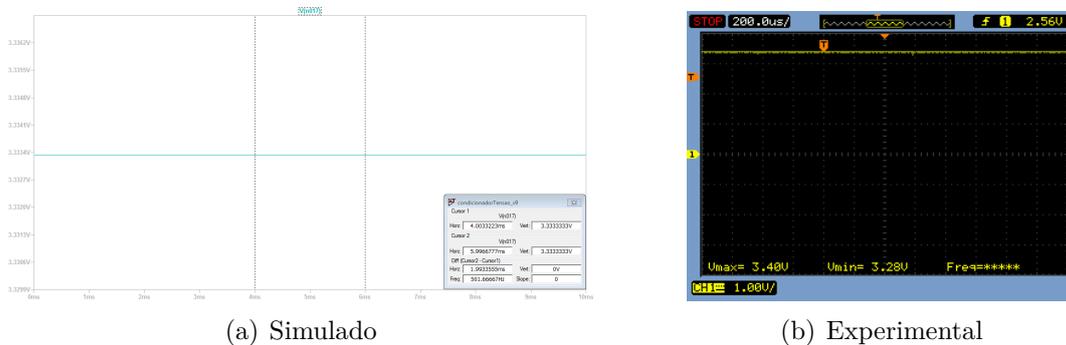


Figura 3.12: Gráficos obtidos para o pino CV do *timer*.

Para leituras do pino CV na Figura 3.12 tem-se tensões DC de $3.33V$ em 3.12(a) e de $3.28V$ a $3.40V$ em 3.12(b).

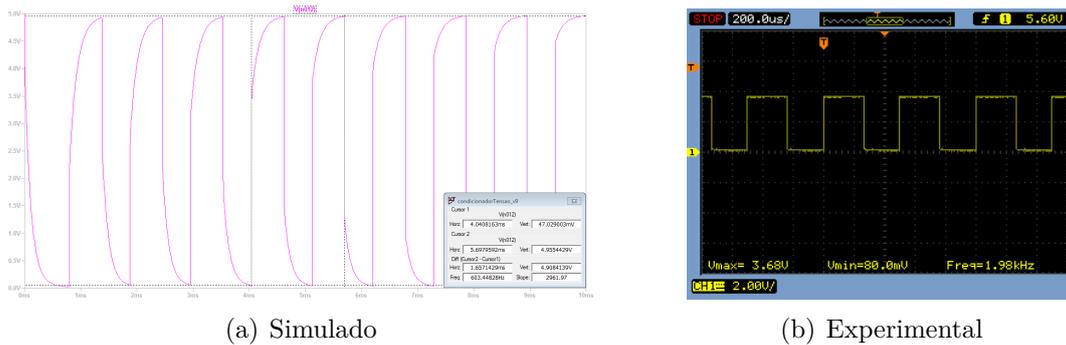
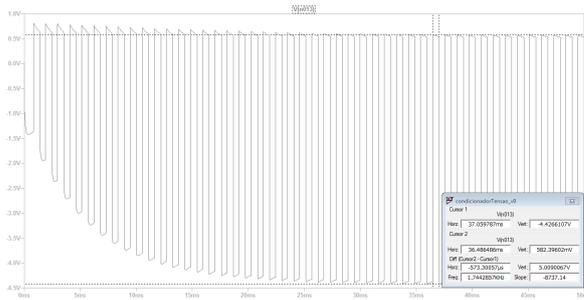


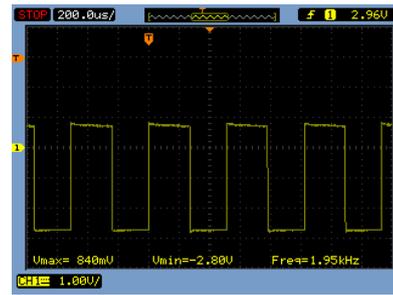
Figura 3.13: Gráficos obtidos para o pino OUT do *timer*.

Já para as leituras no pino OUT do *timer* na Figura 3.13 outras diferenças que não na frequência surgem. Enquanto em simulação o nó em questão fornece uma forma de onda da tensão mais próxima de várias respostas sub-amortecidas subsequentes com transições suaves entre $47.03mV$ e $4.96V$ a estimados $905.17Hz$ (3.13(a)), empiricamente tal nó forneceu uma forma de onda quadrada da tensão com transições abruptas entre $80mV$ e $3.68V$ a $1.98kHz$ medidos (3.13(b)). Estipula-se que essa discordância entre as formas e a voltagem máxima atingida para os dois casos deve-se à dinâmica de operação interna e resposta do circuito integrado, as quais devem ser diferentes entre o modelo simulado e o modelo adquirido - estando também relacionada à diferença na frequência de operação nos dois casos.

Como consequência da situação descrita, as medidas tomadas para a tensão no nó entre os diodos do grampeador e do detector de vale no dobrador de tensão (Figura 3.14)



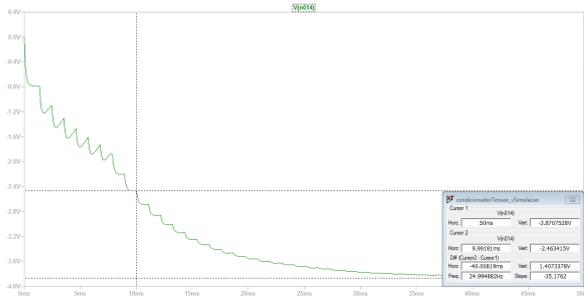
(a) Simulado



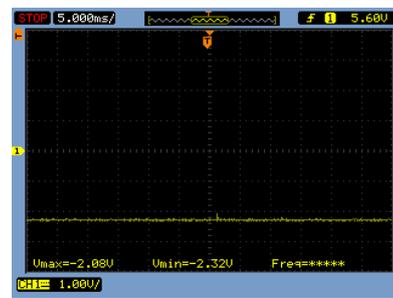
(b) Experimental

Figura 3.14: Gráficos obtidos para o sinal entre os diodos da fonte de alimentação negativa projetada.

também divergiram: tem-se alternância de tensões entre $-4.43V$ e $582.40mV$ a $872.14Hz$ em 3.14(a) e entre $-2.80V$ e $840mV$ a $1.95kHz$ em 3.14(b).



(a) Simulado



(b) Experimental

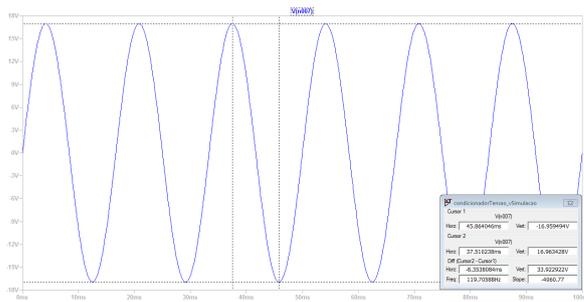
Figura 3.15: Gráficos obtidos para a tensão gerada pela fonte de alimentação negativa projetada.

Mesmo com os imprevistos descritos previamente, ao final do circuito da fonte de alimentação negativa a tensão gerada na prática - entre $-2.32V_{DC}$ e $-2.08V_{DC}$ (3.15(b)) - provou-se suficiente para o propósito estabelecido de evitar saturações em valores saída do condicionador próximos de $0V$ (ainda que não tenha atingido os valores esperados na teoria de $-2.8V$ ou simulados de -3.87).

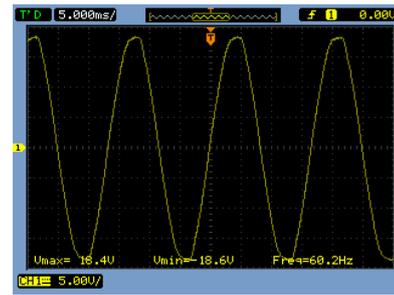
3.1.3 Filtro Passa-Baixas Ativo com Deslocamento DC

Por fim, analisando os sinais em nós do filtro condicionador propriamente dito:

Assim como discutido nos resultados referentes à fonte de alimentação positiva, percebe-se da Figura 3.16 que na prática a amplitude da senoide pós-trafo (variando de $-18.6V$ a $18.4V$ em $60.2Hz$, 3.16(b)) foi maior do que a amplitude simulada em computador (variando de $-16.96V$ a $16.96V$ em $60Hz$, 3.16(a)) para uma rede elétrica de $220V_{rms}$, mas agora por cerca de $1.5V_{pk}$ no momento da medição. Por meio das mesmas considerações e



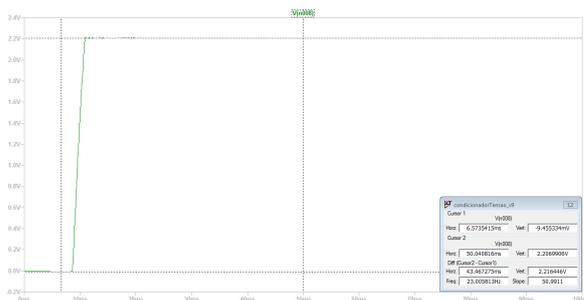
(a) Simulado



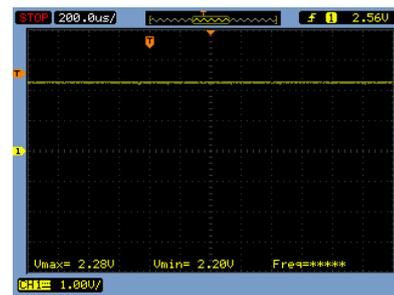
(b) Experimental

Figura 3.16: Gráficos obtidos para o sinal de baixa tensão na saída do transformador utilizado.

suposições feitas na subsecção citada, é possível chegar numa tensão de pico para a senoide na tomada da bancada em laboratório cerca de $27.50V$ maior (totalizando $338,63V_{pk}$ ou $239.45V_{rms}$) do que o ideal para uma rede de $220V_{rms}$, reforçando então o resultado derivado anteriormente - ao menos qualitativamente. Com esse reforço em mente, o resultado obtido para a saída do circuito implementado poderá ser melhor explicado.



(a) Simulado



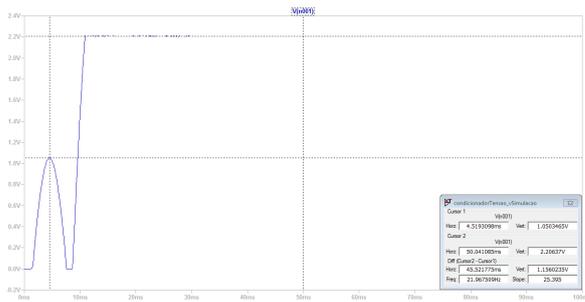
(b) Experimental

Figura 3.17: Gráficos obtidos para o deslocamento DC gerado, presente na porta não-inversora do amp-op.

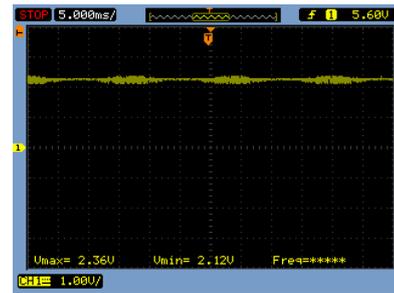
Antes dessa investigação final entretanto, é interessante verificar a acurácia do sinal na entrada não-inversora para deslocamento DC gerado na Figura 3.17, sendo os valores medidos entre $2.20V$ e $2.28V$ (3.17(b)) relativamente parecidos com os $2.20V$ esperados teoricamente e também com os $2.21V$ simulados para tal sinal (3.17(a)).

Além disso, pelo funcionamento do *op-amp* com realimentação negativa (garantindo o curto virtual), nota-se quase a mesma acurácia na Figura 3.18 para o sinal na entrada inversora, com valores medidos entre $2.12V$ e $2.36V$ (3.18(b)) relativamente parecidos com os $2.21V$ simulados para o sinal (3.18(a)) - ademais de uma pequena mas perceptível influência da senoide de entrada do nó anterior à resistência $R1$ ($33k\Omega$).

Finalmente, podem-se observar as tensões de saída simuladas e empíricas do circuito condicionador na Figura 3.19. Como esperado, de acordo com os pontos levantados no

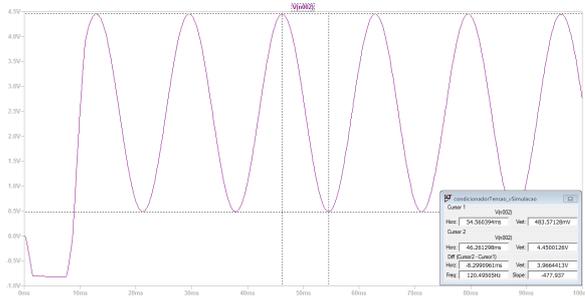


(a) Simulado

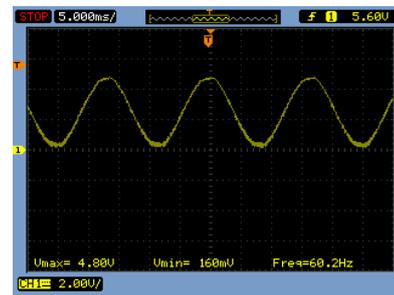


(b) Experimental

Figura 3.18: Gráficos obtidos para a tensão presente na porta inversora do amp-op.



(a) Simulado



(b) Experimental

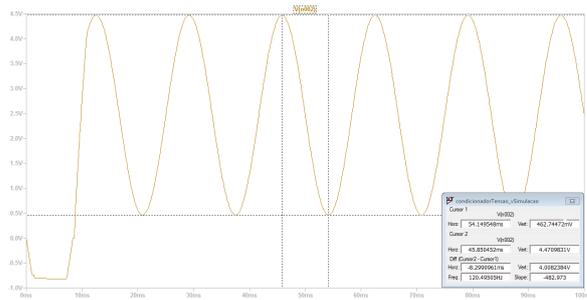
Figura 3.19: Gráficos obtidos para a saída do amplificador operacional com a presença do capacitor para filtragem.

início desta sub-seção, as tensões presentes em 3.19(a) se caracterizaram por um intervalo menor ($483.57mV$ a $4.45V$, em $60Hz$) quando comparadas às tensões presentes em 3.19(b) ($160mV$ a $4.80V$, em $60.2Hz$), justamente em função da presença na tomada em teste de voltagens maiores do que o previsto para uma instalação de $220V_{rms}$ perfeita. Contudo e felizmente, essa alta na voltagem ainda permaneceu dentro da margem de erro de 25% projetada para o sub-sistema e a componente DC introduzida centralizou verticalmente bem a senoide em torno do valor desejado de $2.5V$, permitindo a visualização e subsequente leitura do sinal condicionado sem maiores distorções ou perdas de informação.

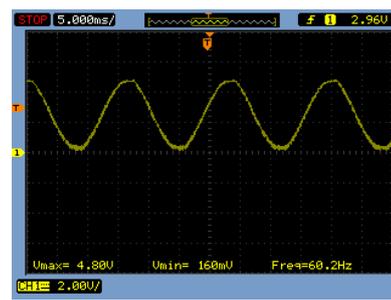
3.1.4 Testes adicionais

De forma a complementar e reforçar a importância dos resultados obtidos até o momento, seguem os produtos de alguns outros ensaios realizados:

Fora um pequeno aumento no intervalo de tensão em ambiente simulado 3.20(a) ($462.75mV$ - $4.47V$, $60Hz$) - na bancada de testes obteve-se $160mV$ - $4.80V$, $60.2Hz$ (3.20(b)) - não houveram maiores diferenças no sinal de saída para o circuito final sem a



(a) Simulado



(b) Experimental

Figura 3.20: Gráficos obtidos para a saída do amplificador operacional sem a presença do capacitor para filtragem.

presença do capacitor para a filtragem da Figura 3.20 quando comparado à saída obtida anteriormente em 3.19.

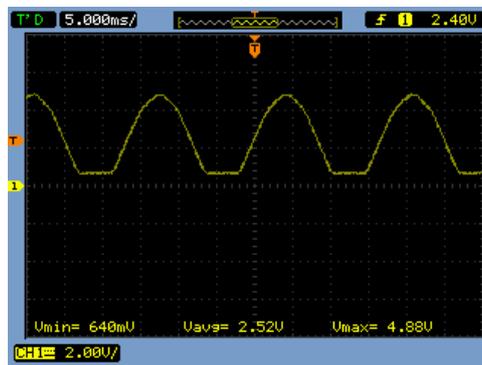


Figura 3.21: Curva obtida na saída do circuito implementado sem a devida alimentação negativa.

Demonstra-se na Figura 3.21 a consequência do uso da alimentação negativa inadequada no dispositivo eletrônico de amplificação do filtro ativo - no caso o aterramento desse pino de alimentação não permite ao CI que alcance tensões adjacentes à 0V em sua saída, limitando a operação do circuito projetado a um nível de saturação "negativa" em 640mV.

Por fim, para o circuito projetado e implementado sem demais alterações é possível ver que no caso de uma falha no fornecimento de energia por parte da tomada, ou simplesmente na ocasião do sub-sistema estar desconectado da rede elétrica, a voltagem apresentada na saída do circuito fica em torno de $-720mV$ até $-680mV$ (Figura 3.22) - ou seja, dentro do limite estabelecido anteriormente e garantido pelo diodo zener com tensão de polarização direta de 1V, de forma que o valor lido pelo microcontrolador será considerado como 0V.



Figura 3.22: Sinal obtido na saída do circuito implementado com o terminal primário do transformador desconectado da tomada.

3.2 Sub-sistema digital

Nesta secção serão introduzidos resultados relativos à digitalização do sinal analógico de interesse (especificamente à quantização e amostragem), às métricas de qualidade de energia geradas e à resposta do monitor de falhas a diferentes situações emuladas em ambiente controlado e comunicadas pela plataforma de prototipagem utilizada (Figura 3.23).

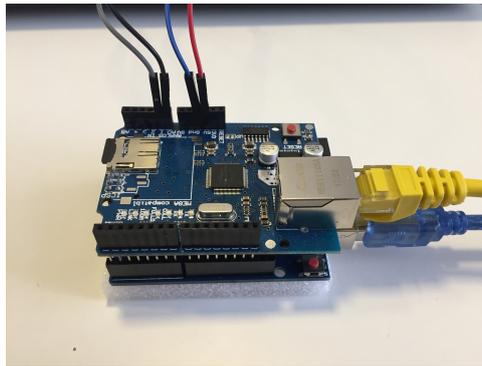


Figura 3.23: Arduino Uno utilizado com o *Shield* de Ethernet montado.

Dito isso e seguindo a ordem das tarefas como apresentadas no código desenvolvido (Figura 3.24) - digitalização, cálculo de parâmetros e comunicação:

```
monitorTensao | Arduino 1.8.7
Arquivo Editar Sketch Ferramentas Ajuda
monitorTensao
/*
 * Sub-EHE, Dezembro 2019, Brasília
 * Projeto Final em Engenharia de Computacao 2
 * Titulo: Deteccao de falhas no fornecimento de energia
 * Autor: Felipe Teixeira Cardoso
 * Orientador: Daniel Chaves Cafe
 */

#include <SPI.h>
#include <Ethernet.h>
#include <PubSubClient.h>

// #define DEBUG_MONITOR // Para exibir informacoes textuais no monitor serial
// #define DEBUG_PLOTTER // Para exibir informacoes graficas no plotter serial
// #define LOCAL_SERVERS // Para utilizar o broker de um servido local pre-estabelecido

// Variaveis relativas a amostragem dos sinais
const int nMaxBytesPck = 32; // Tamanho em bytes de cada pacote de dados lidos
byte pckV[nMaxBytesPck]; // Pacote de dados a respeito da tensao lida
byte pckI[nMaxBytesPck]; // Pacote de dados a respeito da corrente lida
short nSampPerSec = 1000; // Frequencia de amostragem em hertz
unsigned long uSecSampPer = 1000000/nSampPerSec; // Período de amostragem em microssegundos
float secPerPck = (float)nMaxBytesPck/nSampPerSec; // Tempo que cada pacote representa em segundos
unsigned long mSecDelayPck = 5000; // Tempo de espera entre o envio de pacotes consecutivos em milisegundos

// Variaveis relativas a conexao MQTT
byte mac[] = {0x0E, 0x0E, 0x92, 0x27, 0x06, 0x67}; // Endereco MAC do arduino no formato de vetor de bytes
char macStr[] = "0e0e0e92270667"; // Endereco MAC do arduino no formato de vetor de chars
byte ip[] = {192, 168, 100, 042}; // Endereco IP do arduino no formato de vetor de bytes
char serverName[] = "test.mosquitto.org"; // URL do servidor de testes do mosquitto, onde esta o broker remoto
```

Figura 3.24: Código desenvolvido no ambiente Arduino.

3.2.1 Digitalização da entrada analógica

Como definido, a resolução escolhida para cada amostra foi, por simplicidade, de 1 byte - 8 bits, cada uma podendo representar $2^8 = 256$ níveis de tensão entre $0V$ e $5V$ - e a frequência de amostragem de $f_s = 1kHz$ - cumprindo bem o critério de Nyquist para a situação em questão ($f_s > 2f_m = 720Hz$).

A primeira escolha para a quantização pode ser comparada a uma outra alternativa de 2 bytes (de forma a utilizar todos os 10 bits/1024 níveis de resolução possíveis por parte do conversor A/D usado) como na Figura 3.25, com ambas amostradas na maior taxa permitida pelo microcontrolador para melhor visualização. Com precisões de $19.5mV$ para 1 byte e de $4.88mV$ para 10 bits, conclui-se que a diferença entre os dois sinais é pouco perceptível e não justifica o aumento no tamanho dos dados a serem enviados em função do byte adicional de informação por amostra ou mesmo o esforço computacional de codificar e decodificar os 10 bits de uma amostra entre dois bytes diferentes na mensagem - por exemplo, utilizar os últimos 6 bits restantes de uma primeira amostra em 2 bytes e os 4 primeiros bits dos 2 bytes seguintes para enviar a próxima amostra.

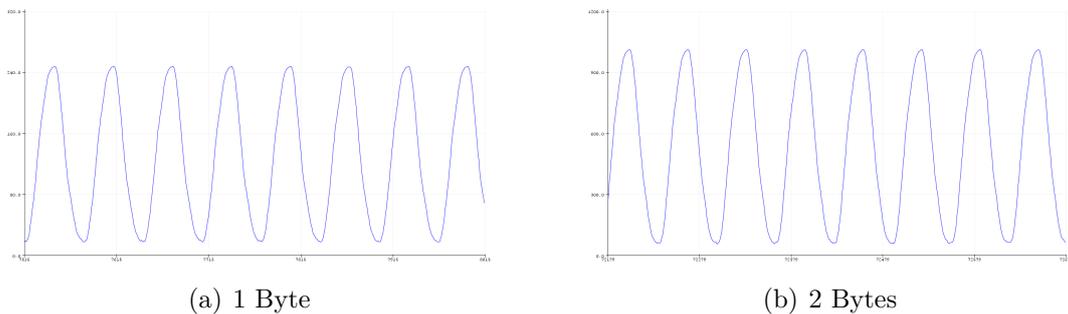


Figura 3.25: Avaliação gráfica da quantização do sinal condicionado para diferentes resoluções.

3.2.2 Métricas de qualidade de energia geradas

Para avaliação das métricas de qualidade de energia calculadas em *software*, realizou-se cinco ensaios em horários diferentes para uma mesma tomada de um prédio residencial do Distrito Federal gerando os cinco vetores/pacotes de dados referentes à tensão originada do circuito condicionador de tensão e lida pelo microcontrolador. Os valores obtidos para tais métricas relativas aos pacotes gerados estão dispostos na Tabela 3.1, na qual os valores calculados f_m , V_{rms} , V_{max} e V_{min} representam, respectivamente, a frequência do sinal, o valor eficaz, o valor de pico e o valor de vale no pacote em questão (com **byte** indicando a representação do valor lido com a resolução escolhida, V_c esse mesmo valor em termos da

Tabela 3.1: Resultados de ensaios feitos em rede elétrica residencial.

Pacote	f_m - byte/Hz	V_{rms} - byte/V_c	V_{max} - byte/V_c/V_t	V_{min} - byte/V_c/V_t
1	58/58	83/252.18	243/4.75/352.45	18/0.35/-331.18
2	58/58	81/246.11	240/4.69/343.33	19/0.37/-328.14
3	58/58	82/249.15	244/4.77/355.49	18/0.35/-331.18
4	58/58	81/246.11	243/4.75/352.45	19/0.37/-328.14
5	62/62	80/243.07	243/4.75/352.45	22/0.43/-319.03

tensão proporcional na saída do condicionador e V_t o mesmo valor em termos da tensão correspondente de saída na tomada).

Nota-se na tabela apresentada que, além de estarem satisfatoriamente perto do esperado na prática - denunciando uma tensão na tomada pouco maior do que os $220V_{rms}$ a $60Hz$ esperados, assim como nos experimentos feitos em laboratório - os resultados obtidos mantiveram-se relativamente próximos entre diferentes pacotes, o que pode ser um indicativo de certa estabilidade da rede elétrica local.

3.2.3 Respostas do monitor de falhas

Como descrito no capítulo anterior, o monitor de falhas executado sobre a plataforma Windows de um computador pessoal (representando aqui o servidor no LDCI) receberá mensagens do Arduino por meio da interface serial conectada por USB apenas nos casos de falta de conexão da placa com a internet (*NI*) e de falta de energia no edifício (*NE*). Para testar esta funcionalidade, emularam-se tais situações simplesmente desconectando o cabo de rede do *shield Ethernet* no caso da primeira e desconectando ambos o cabo de rede do *shield* e o *plug* de alimentação da tomada simultaneamente no segundo caso. Para uma exploração mais completa desse recurso, também emularam-se as situações com operação normal tendo as duas redes, elétrica e de internet, em funcionamento (OK) e com *plug* de alimentação da tomada desconectado da rede elétrica (*NP*) - as quais não devem gerar respostas por parte do monitor de falhas. Os resultados desses testes do ponto de vista do monitor podem ser conferidos na Figura 3.26, que confirma o sucesso do procedimento implementado.

```
C:\Users\Filipe\Documents\UnB\9º Semestre\Projeto Final em Engenharia de Computação 2\entreg...
Setting ConnMask successfull
Waiting for Data Reception...
-----
Data Received:
Sampling Configuration:
  nMaxBytesPck (bytes): 32
  nSampPerSec (Hz): 1000
  uSecSampPer (uSec): 1000
  secPerPck (sec): 0.03
  mSecDelayPck (mSec): 5000
Connectivity configuration:
Data Received:
Data Received:
  mac:      macStr: deed92270667
  ip:      serverName: test.mosquitto.org
  clientName: ac:monitor:arduino:deed92270667
  topicName: voltageCurrentMonitor
Data Received:
```

(a) *OK/NP* - sem mensagem recebida (como previsto)

```
C:\Users\Filipe\Documents\UnB\9º Semestre\Projeto Final em Engenharia de Computação 2\entreg...
Waiting for Data Reception...
-----
Data Received:
Sampling Configuration:
  nMaxBytesPck (bytes): 32
  nSampPerSec (Hz): 1000
  uSecSampPer (uSec): 1000
  secPerPck (sec): 0.03
  mSecDelayPck (mSec): 5000
Connectivity configuration:
Data Received:
Data Received:
  mac:      macStr: deed92270667
  ip:      serverName: test.mosquitto.org
  clientName: ac:monitor:arduino:deed92270667
  topicName: voltageCurrentMonitor
Data Received:
Data Received:
-2
```

(b) *NI* - mensagem recebida '-2' (indicando falta de conexão, como previsto)

```
C:\Users\Filipe\Documents\UnB\9º Semestre\Projeto Final em Engenharia de Computação 2\entreg...
Waiting for Data Reception...
-----
Data Received:
Sampling Configuration:
  nMaxBytesPck (bytes): 32
  nSampPerSec (Hz): 1000
  uSecSampPer (uSec): 1000
  secPerPck (sec): 0.03
  mSecDelayPck (mSec): 5000
Connectivity configuration:
Data Received:
Data Received:
  mac:      macStr: deed92270667
  ip:      serverName: test.mosquitto.org
  clientName: ac:monitor:arduino:deed92270667
  topicName: voltageCurrentMonitor
Data Received:
Data Received:
0
```

(c) *NE* - mensagem recebida: '0', (indicando falta de energia, como previsto)

Figura 3.26: *Display* do monitor de falhas para as diferentes situações possíveis com relação a energia e conexão com a internet.

3.3 Sub-sistema de rede

Finalmente, relatam-se nesta última secção os resultados relacionados à publicação da página *web* para o amplo acesso às informações do sensor, ao envio e recebimento de mensagens por meio do protocolo MQTT, tanto pela rota normal de funcionamento como pela rota de segurança (Figura 3.27), e à interpretação da mensagem recebida e disposição dos dados que ela carrega em uma interface intuitiva para interação com o usuário - novamente emulados em ambiente controlado. Vale observar que além do sinal de tensão desejado, para os testes realizados nesta discussão, os dados obtido relativos à corrente elétrica foram emulados pela simples conexão da segunda entrada analógica com o pino de fornecimento DC de 3.3V nominais da própria placa Arduino. Nesse caso, a tensão lida pelo microcontrolador foi em média de $3.44V_{DC}$ representado digitalmente em um byte de valor 176 - indicando assim o que seria a passagem de cerca de 1.88A pelo ACS712-5A em uma situação real.



Figura 3.27: Hardware da interface do sub-sistema digital com o de redes na rota de segurança do sistema.

3.3.1 Publicação da página *web*

Dessa forma, segue primeiramente na Figura 3.28 o *web site* desenvolvido quando inicializado em um *browser* qualquer (apontando desconhecimento com relação ao estado do sistema, como esperado).

Ainda, observa-se na página a URL “<https://filipe5214957.github.io>”, indicando a publicação da página pelo serviço de *hosting* do GitHub. Isto pode ser também conferido no próprio domínio do fornecedor desse serviço, como demonstrado na Figura 3.29.

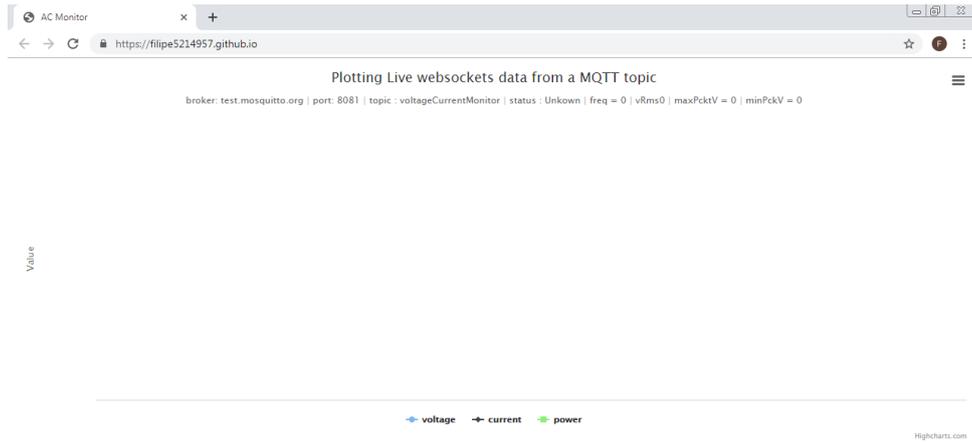


Figura 3.28: Página criada iniciada indicando estado desconhecido (*Unkown*) do sistema.

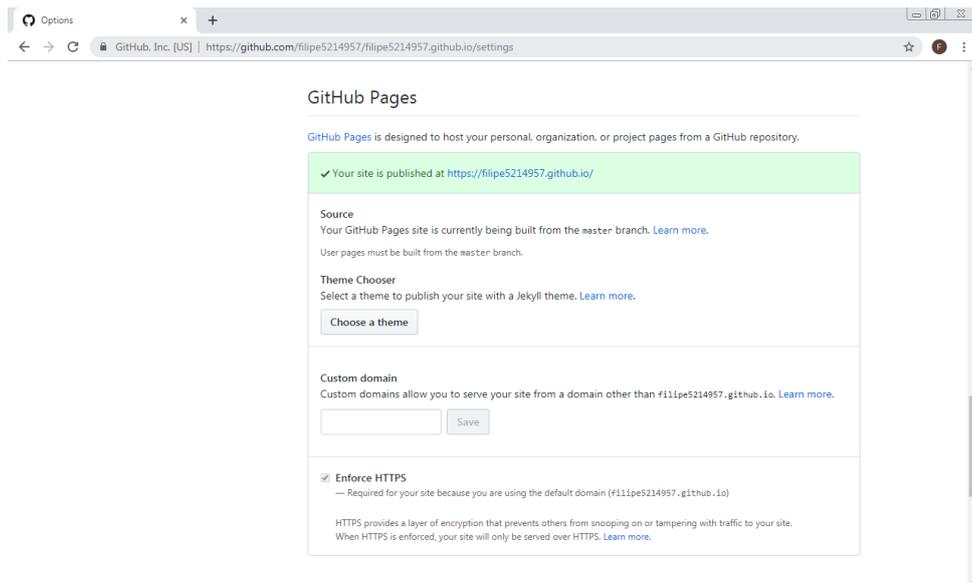
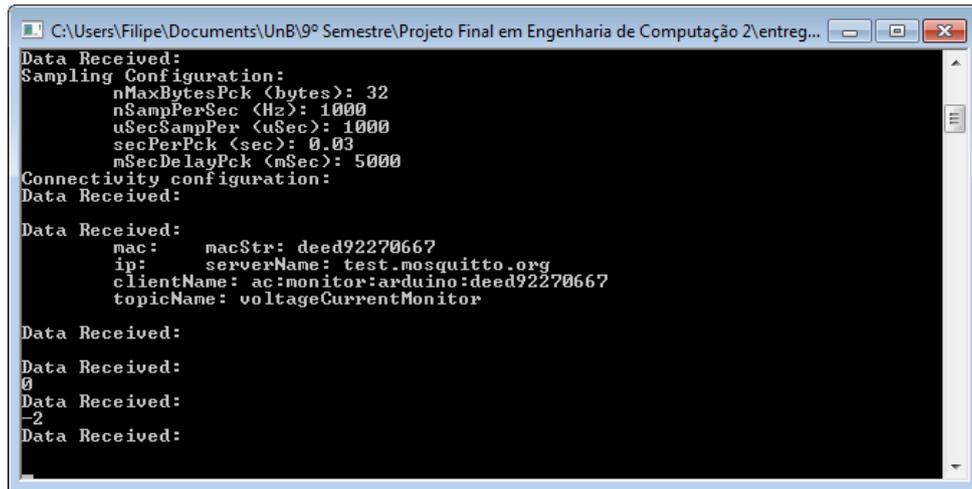


Figura 3.29: Página publicada com sucesso pelo *host* GitHub Pages.

3.3.2 Troca de mensagens por meio do MQTT

A maneira como se conduziu o teste da troca unidirecional de mensagens pelas duas rotas estabelecidas se deu pela passagem por cada um dos quatro possíveis estados de funcionamento do sistema de forma subsequente e na ordem *NE*, *NI*, *OK* e *NP* - com cada transição entre estados sendo emulada de acordo com as ações apresentadas ao final da secção anterior, quando tratou-se das respostas do monitor de falhas.



```
C:\Users\Filipe\Documents\UnB\9º Semestre\Projeto Final em Engenharia de Computação 2\entreg...
Data Received:
Sampling Configuration:
  nMaxBytesPck (bytes): 32
  nSampPerSec (Hz): 1000
  uSecSampPer (uSec): 1000
  secPerPck (sec): 0.03
  mSecDelayPck (mSec): 5000
Connectivity configuration:
Data Received:
Data Received:
  mac:      macStr: deed92270667
  ip:      ip:      serverName: test.mosquitto.org
  clientName: ac:monitor:arduino:deed92270667
  topicName: voltageCurrentMonitor
Data Received:
Data Received:
0
Data Received:
-2
Data Received:
```

Figura 3.30: *Display* do monitor de falhas para os testes de envio de mensagens realizados.

Presente na Figura 3.30 estão os resultados desse procedimento relativos ao *endpoint* do *publisher*, especificamente do ponto de vista da rota de segurança estabelecida - a qual enviará (ou não) para o *broker* no tópico “*voltageCurrentMonitor*” um alerta com o estado atual de acordo com os bytes recebido do Arduino Uno. Percebe-se em um primeiro momento o recebimento de um byte com valor ‘0’ (indicando ao monitor o estado *NE*), seguido de dois bytes ‘-2’ (indicando o estado *NI*) e, a partir de então, não se verifica a entrada de qualquer outro byte (como esperado para os dois estados remanescentes *OK* e *NP*, nos quais a rota de funcionamento normal ainda está disponível para o envio de mensagens).

Esse comportamento em conjunto com o que é mostrado na Figura 3.31 validam a comunicação implementada: a imagem em questão informa que, relativo ao *endpoint* do *subscriber* - inscrito no tópico “*voltageCurrentMonitor*” mencionado - as mensagens recebidas do *broker* indicaram a passagem pelos estados *NE* (2 bytes recebidos por meio da rota de segurança), *NI* (outros 2 bytes vindos da rota de segurança), *OK* (69 bytes recebidos pela rota de funcionamento normal) e *NP* (2 bytes recebidos da rota normal), nessa ordem.

3.3.3 Disposição dos dados e informações na interface e interação com o usuário

A tradução das mensagens recebidas em informação visual para interação com o usuário é verificada na Figura 3.32, que não só demonstra o uso das ferramentas de mostrar/esconder para os três sinais *plotados* e de acesso à informação de amostras individuais, mas também o acesso ao sistema por meio de dispositivos *mobile* fora da rede local.

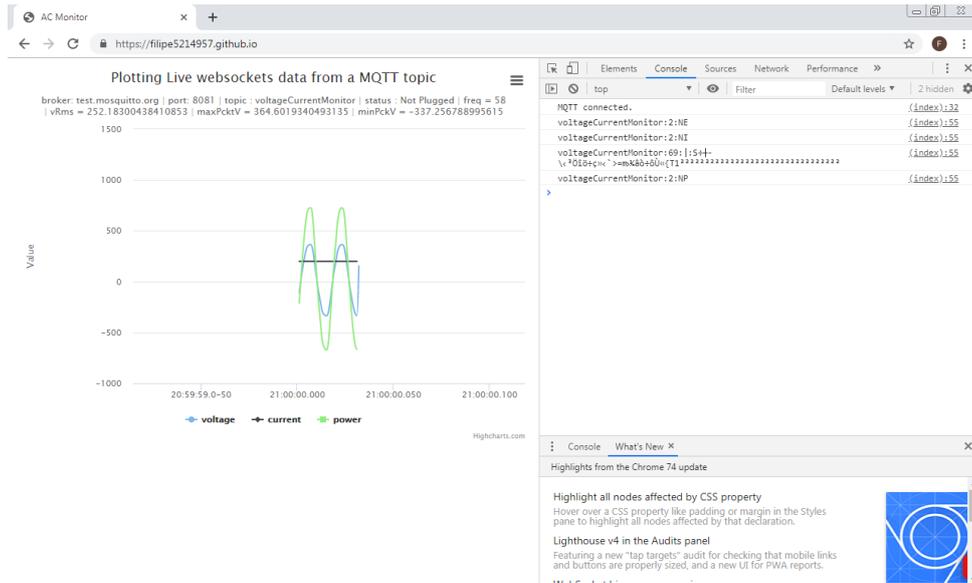


Figura 3.31: Página *Web* com ferramentas de desenvolvedor a mostra para os testes de recebimento de mensagens realizados.

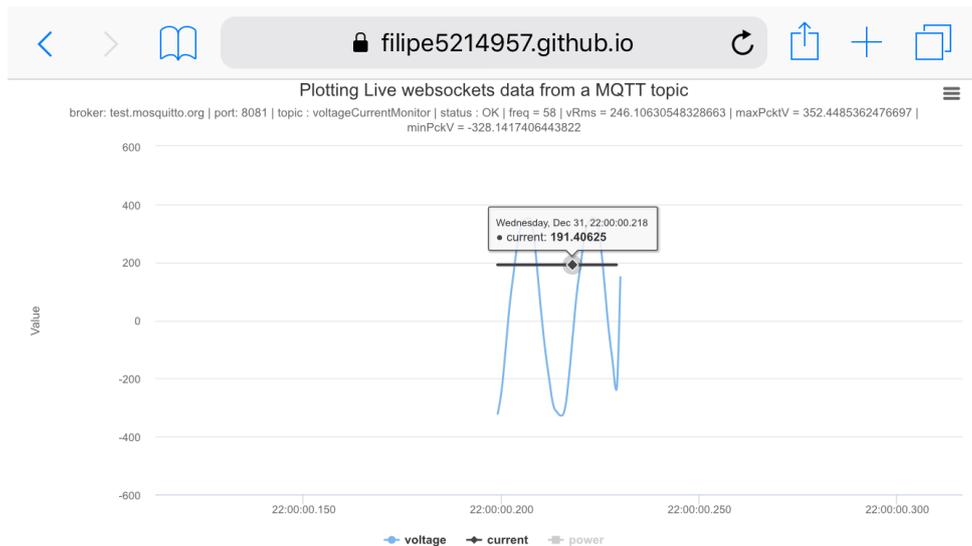
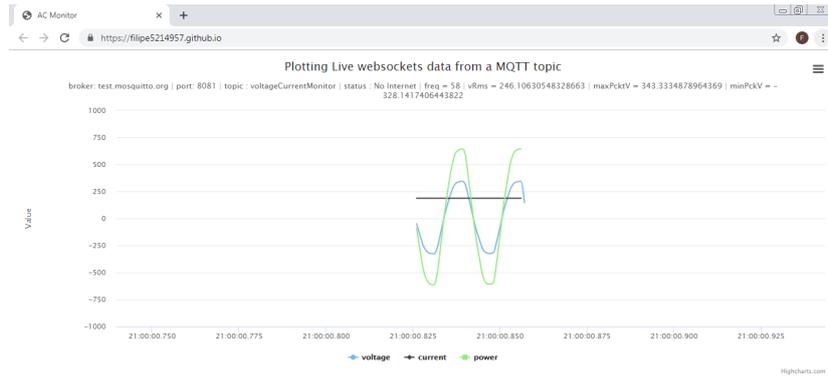


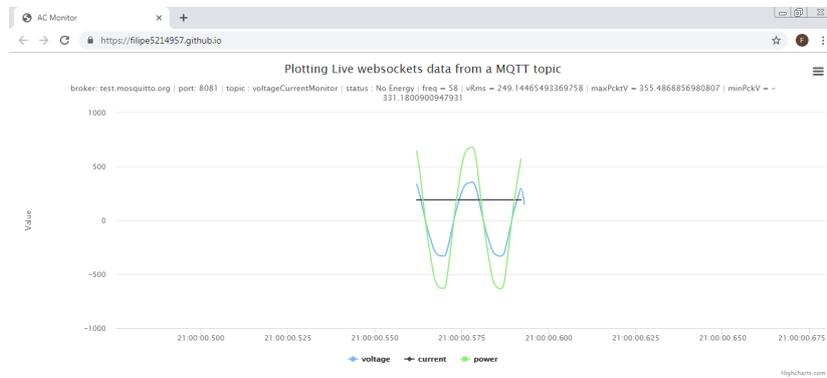
Figura 3.32: Demonstração das ferramentas para análise de amostras com o sistema em estado de funcionamento normal (*OK*), visualizada por meio de um navegador em *smartphone*.

É interessante notar também que a forma de onda da potência instantânea - habilitada nas capturas da Figura 3.33 e calculada a partir dos outros dois sinais - condiz com o esperado para essa situação (já que se mantém em fase com o sinal de tensão enquanto seu módulo é modificado pelo produto com o sinal de corrente DC).

Das figuras 3.33(a) e 3.33(b), assim como da figura Figura 3.31 exposta anteriormente,



(a) NI



(b) NE

Figura 3.33: Página *Web* visualizada em navegador de *notebook* para duas das diferentes situações possíveis com relação a energia e conexão.

também é possível ver que os indicadores de frequência (*freq*), tensão eficaz (*vRms*), tensão de pico (*maxPcktV*) e de vale (*minPcktV*) sobre as ondas *plotadas* estão na página mesmo com o indicador de *status* apontando que este não está no modo de funcionamento normal (OK), o que simplesmente significa que a página está mostrando os últimos dados recebidos do Arduino antes da mudança na situação de operação do sistema.

Capítulo 4

Considerações Finais

Finalmente, o presente capítulo encerra o trabalho com uma breve consideração a respeito dos resultados obtidos nos testes em comparação aos objetivos inicialmente definidos para o projeto, seguida de uma conclusão relatando as maiores dificuldades encontradas no desenvolver do sistema junto aos aspectos financeiros do protótipo implementado e, em um último momento, algumas perspectivas futuras para uma possível evolução do trabalho aqui apresentado.

4.1 Resultados e Objetivos Cumpridos

Em geral, os resultados obtidos para cada sub-sistema concebido mostraram-se bem satisfatórios. Apesar de discrepâncias notáveis, principalmente no funcionamento do circuito analógico real e na interface deste com o circuito digital em relação ao esperado teoricamente, é possível afirmar que o sistema prototipado atendeu a todos os três objetivos gerais postulados ao início deste texto, bem como aos cinco objetivos específicos de cada objetivo geral, de maneira satisfatória:

- De forma a garantir uma medida adequada, precisa e eficiente dos valores de tensão em uma tomada da rede de energia AC com o circuito condicionador de tensão:
 - utilizou-se um trafo-transformador para gerar um novo referencial terra, desacoplar da malha elétrica o circuito analógico e, em conjunto com um amplificador operacional na configuração inversora, reduzir a tensão para leitura;
 - desenvolveram-se fontes de tensão DC positiva e negativa para fornecer a alimentação adequada aos componentes eletrônicos utilizados;
 - projetou-se um filtro passa-baixa ativo com deslocamento DC e componentes de proteção adequado para filtrar ruídos indesejados e isolar/proteger a carga do dispositivo responsável pela medida.

- Para ler e obter, a partir dos dados mensurados, informações que caracterizem a situação da energia fornecida pela malha e que sejam úteis na identificação de anomalias com o dispositivo Arduino Uno:
 - utilizou-se o conversor A/D do processador ATmega328P acoplado à placa do Arduino para converter o sinal analógico condicionado em digital;
 - desenvolveu-se um programa em C++ para o microprocessador ATmega328P da placa capaz de coordenar a leitura e processamento das amostras em três etapas;
 - reservaram-se vetores de bytes para representar forma de onda equivalente à tensão e à corrente;
 - obteve-se por meio de procedimentos no programa escrito os bytes com valores estimados de frequência, tensão eficaz, tensão de pico e tensão de vale.
- Para estabelecer uma forma de comunicação rápida, confiável e segura para troca de dados, informações e avisos entre o sistema em laboratório e computadores (locais ou não) gerenciados por um administrador com o periférico de *ethernet* do Arduino e a estrutura de rede adotada:
 - elegeu-se o protocolo MQTT para o envio e recepção em função do tipo de mensagem;
 - padronizou-se dois formatos gerais para as mensagens: pacotes de 2 bytes, contendo apenas alertas relativos ao estado de funcionamento do sistema, e pacotes de 70 bytes, contendo os dados e informações gerados em um intervalo de amostragem;
 - baseado na estrutura *publisher - broker - subscriber* do protocolo MQTT criou-se uma interface cliente-servidor-cliente intermediada por um servidor de *broker* local ou remoto (Mosquitto) alcançável por duas rotas de comunicação (uma pelo *Ethernet Shield* e outra por um Mini-modem 3G ligado ao servidor e acionado por um programa escrito em linguagem C) para garantir a robustez do sistema;
 - utilizando as linguagens HTML e JavaScript em conjunto com bibliotecas gráficas e de MQTT prontas disponibilizadas *online* criou-se um interface gráfica para cliente *subscriber* em uma página web abrigada por um *server* no domínio do GitHub;
 - por meio de opções de autenticação presentes na ferramenta MQTT escolhida (Mosquitto) definiu-se uma restrição de acesso ao sistema em laboratório (usuário e senha);

De um ponto de vista mais crítico, o objetivo específico cumprido com menor grau de confiança em função da falta de testes específicos para verificar o seu funcionamento foi o último apresentado - a definição de restrições de acesso ao sistema em laboratório. Se apresentaram questionáveis também algumas das formas de onda resultantes dos experimentos em laboratório, com a acurácia do sinal condicionado final não podendo ser verificada em um teste de controle, isto é, em uma situação ideal de tensão na rede elétrica.

4.2 Conclusão

Tratando-se de um projeto relativamente extenso, englobando diversas áreas do conhecimento em engenharia - desde *design* de circuitos eletrônicos e qualidade de energia, passando por processamento de dados em microcontroladores programáveis e por interfaces/protocolos de rede até o desenvolvimento de interface gráfica em aplicações para monitoramento em tempo real - é de se esperar que algumas dificuldades tenham surgido ao longo deste trabalho, as mais evidentes destas relacionadas ao sub-sistema analógico.

Em um primeiro momento, a implementação de um protótipo para o condicionador de tensão demonstrou ser uma tarefa desafiadora pois os cálculos teóricos e simulações computacionais, ainda que feitos corretamente, não necessariamente se traduziram bem para a prática. Nesse sentido, os maiores problemas encontrados foram relativos à escolha de um trafo e um amplificador operacional adequados para o circuito. Em versões iniciais do projeto desta parte do sistema, a idéia de manter o circuito analógico simples e compacto levou à decisão de utilizar as alimentações já disponíveis pelo Arduino ($5V$ e *Ground*) para fornecer energia a um amp-op *rail-to-rail*, de forma que seriam desnecessários dois terminais de saída no transformador e as duas fontes projetadas na versão final do sub-sistema. Na prática, a dificuldade em encontrar amplificadores com essa característica específica ou similares que proporcionassem resultados satisfatórios levou à necessidade de reprojeter o circuito diversas vezes para, ultimamente, incluir as devidas alimentações, o que por sua vez levou à outros problemas relativos a referência do circuito, geração de tensões negativas a partir de tensões positivas, utilização de componentes altamente não lineares e dimensionamento da corrente proveniente do transformador e presente nos demais eletrônicos do circuito.

Para os demais sub-sistemas, as maiores dificuldades estavam relacionadas à inexperiência com algumas das ferramentas - como os protocolos MQTT e HTTPS e as linguagens HTML e JavaScript - e à pouca familiaridade com certos conceitos - como qualidade de energia, parâmetros de comunicação serial (baud rate, timeouts de leitura/escrita, entre outros utilizados pelas bibliotecas Win32 API e windows.h) e padrões Unicode de codifi-

cação binária (UTF8, utilizado pela biblioteca `mqtts31.js`). Com relação ao sub-sistema digital especificamente, encontrar uma forma de estabelecer a comunicação serial com o programa do monitor de falhas executado por um computador pessoal (o servidor monitorado) foi particularmente desafiador. Já com relação ao sub-sistema de rede, o maior desafio foi elaborar um código para dispor os dados e informações na página que fosse rápido o suficiente para acompanhar o *throughput* de mensagens recebidas do *broker* encaminhadas do *publisher*. Mesmo assim, são obstáculos que puderam ser ultrapassados por meio de intensivas pesquisas e até abstraídos por meio da modificação de certos programas prontos.

Apesar dessas dificuldades, o produto final obtido foi satisfatório, com o protótipo implementado gerando um custo total de R\$278.21 ou aproximadamente US\$73.04 pela cotação atual do dólar (junho/2019, com US\$1.00 equivalendo a R\$3.85), de acordo com o orçamento disposto na Tabela 4.1 - vale observar que os itens marcados com asterisco (*) já estavam disponíveis e não foram de fato adquiridos para realização do projeto, portanto os preços foram apenas estimados de pesquisas em lojas *online*.

Dessa maneira, em forma de protótipo o sistema final demonstra um custo de fabricação possivelmente competitivo para o mercado, ao menos considerando o preço de comercialização para outros produtos com funções similares (e de qualidade variável) encontrados no cenário nacional - como o *Net Adapter II Externo Kit Gerenciador Remoto Web/SNMP 64017 - SMS*, vendido a R\$731.90 [47] - e internacional - como o *kit Base Unit SensorGateway* (US\$317), *USB Power Cable for Sensorgateway* (US\$29), *LTE/2G Communication Add-On* (US\$450) e *AC Power Quality Sensor* (US\$571), vendido a um total de US\$1367.00 [48].

4.3 Trabalhos Futuros

Por fim, sobre as perspectivas futuras para esse trabalho, algumas idéias surgem de pontos levantados ao longo do texto:

- Reprojeter o circuito de condicionamento analógico de forma a torná-lo mais compacto e barato, eliminando a necessidade das fontes implementadas para alimentação por meio do uso de componentes mais precisos e com não-linearidades menos significativas;
- Produzir um novo protótipo do circuito reprojeterado em uma placa impressa encapsulada (em conjunto com a placa do microprocessador) em uma caixa de plástico, de forma a produzir sinais condicionados mais precisos e reduzir interferências externas;

Tabela 4.1: Orçamento do projeto desenvolvido.

	Item	Preço (R\$)
Analogico (R\$109,54)	Plug de tomada com cabo de alimentação e solda	20,00
	Transf Hayonik 12VAC 800mA	27,40
	Protoboard 830 Furos MB-102	20,45
	Resistor de Filme de Carbono 33K - 1/4W	0,04
	Resistor de Filme de Carbono 3K9 - 1/4W	0,04
	Capacitor Cerâmica 100nF/50V	0,06
	LM358 3V-32V/2A	0,49
	Diodo Zener 1N4733 - 5,1V/1W	0,22
	Diodo 1N4007	0,10
	Capacitor Eletrolítico 220uF/25V	0,17
	LM7812 12V/1A	1,32
	Capacitor Poliéster 22nF/50V	0,06
	Resistor de Filme de Carbono 12K - 1/4W	0,04
	Resistor de Filme de Carbono 2K7 - 1/4W	0,04
	Resistor de Filme de Carbono 4K7 - 1/4W	0,04
	Resistor de Filme de Carbono 33K - 1/4W	0,04
	Capacitor Cerâmico 22nF/50V	0,06
	Capacitor Cerâmico 10nF/50V	0,07
	NE555	0,82
	Diodo 1N4007 (2 unidades)	0,10
Capacitor Eletrolítico 22uF/25V	0,07	
Capacitor Eletrolítico 100uF/25V	0,12	
ACS712-5A	16,99	
Cabos Macho-Macho (65 peças) *	11,90	
Cabos Fêmea-Fêmea (40 peças) *	8,90	
Digital (R\$66,55)	Arduino Uno R3 SMD + Cabo USB	66,55
Rede (R\$102,12)	Shield Ethernet/Rede - WIZnet W5100	66,59
	Cabo Flexível 2,5 metros Amarelo UTP CAT5E 28AWG*4P *	12,54
	Minimodem 3G Onda MSA190UP Desbloqueado *	12,99
	Chip pré-pago - Claro 3G Ready *	10,00
	Total	278,21

- Substituir o microcontrolador utilizado por um processador de sinais digitais (*DSP*) próprio para a geração das várias outras informações de qualidade de energia, que requerem maior memória (para o armazenamento de mais amostras com melhor resolução a cada intervalo de leitura) e maior velocidade de processamento (para obter melhores taxas de amostragem, melhor desempenho nos cálculos feitos e possibilitar análises mais complexas - que utilizam da *FFT* por exemplo);
- Eliminar a necessidade de se realizar a comunicação serial entre o servidor monitorado e o produto desenvolvido em si (e conseqüentemente do Mini Modem 3G) por meio de um periférico ligado ao *DSP* capaz de se conectar à internet via *Wi-fi* em situações de funcionamento normal ou por 3G em situações de falha no fornecimento de energia ao roteador local;

- Buscar técnicas alternativas de proteção às mensagens trocadas por meio do protocolo MQTT e de restrição de acesso ao sistema;
- Otimizar, além do que já foi alcançado, o código usado na geração de gráficos e disposição de informações da página *web* para tornar o monitoramento mais dinâmico (no sentido de haver uma variação mais rápida do que é mostrado no *site*);
- Redefinir a interface gráfica do *website*, adicionando recursos para configuração do sistema de forma remota, ou seja, transformando a comunicação de via única entre *publisher* e *subscriber* em um comunicação de duas vias.

Referências

- [1] *Inflection point*, 2019. https://en.wikipedia.org/wiki/Inflection_point, acesso em junho de 2019. ix, 33
- [2] George, W.: *Antique Electric Waffle Irons 1900-1960: A History of the Appliance Industry in 20th Century America*. Trafford Publishing, 2003, ISBN 9781553956327. <https://books.google.com.br/books?id=rW5t4MIgx40C>. 1
- [3] Logic, Daft: *List of the power consumption of typical household appliances*. <https://www.daftlogic.com/information-appliance-power-consumption.htm>, acesso em junho de 2018. 1
- [4] Hill, Jim: *The smart home: a glossary guide for the perplexed*, 2015. <https://www.t3.com/features/the-smart-home-guide>, acesso em junho de 2018. 1
- [5] *Understanding building automation and control systems*. https://web.archive.org/web/20130519124213/http://www.kmcccontrols.com/products/Understanding_Building_Automation_and_Control_Systems.aspx, acesso em junho de 2018. 1, 2
- [6] *Home automation and control market by lighting control, security & access control, hvac control, and geography - global forecast to 2020*. https://web.archive.org/web/20150123015209/http://www.researchandmarkets.com:80/research/xt7v6p/home_automation, acesso em junho de 2018. 1
- [7] *My life at x10*. <https://web.archive.org/web/20140930080338/http://hometoys.com/emagazine.php?url=%2Fhtinews%2Foct99%2Farticles%2Frye%2Frye.htm>, acesso em junho de 2018. 2
- [8] *Overview*. <http://www.bacnet.org/Overview/index.html>, acesso em junho de 2018. 2
- [9] *The future of building automation analytics: The cloud, the edge or both?* <http://connectedcityusa.com/2017/12/future-building-automation-analytics-cloud-edge/>, acesso em junho de 2018. 2
- [10] *Best home automation system*, 2014. <https://www.consumerreports.org/cro/magazine/2014/06/run-your-home-from-your-phone/index.htm>, acesso em junho de 2018. 2

- [11] *Internet of things*. <http://www.gatewaytechnolabs.co.uk/internet-things>, acesso em junho de 2018. 2
- [12] Ashton, Kevin: *That 'internet of things' thing*, 2009. <http://www.rfidjournal.com/articles/view?4986>, acesso em junho de 2018. 2
- [13] *The "only" coke machine on the internet*. https://www.cs.cmu.edu/~coke/history_long.txt, acesso em junho de 2018. 2
- [14] Palermo, Frank: *Internet of things done wrong stifles innovation*, 2014. <https://www.informationweek.com/strategic-cio/executive-insights-and-innovation/internet-of-things-done-wrong-stifles-innovation/a/d-id/1279157>, acesso em junho de 2018. 2
- [15] Köhn, Rüdiger: *Konzerne verbünden sich gegen hacker*, 2018. http://www.faz.net/aktuell/wirtschaft/diginomics/grosse-internationale-allianz-gegen-cyber-attacken-15451953-p2.html?printPagedArticle=true#pageIndex_1, acesso em junho de 2018. 3
- [16] Nordrum, Amy: *Popular internet of things forecast of 50 billion devices by 2020 is outdated*, 2016. <https://spectrum.ieee.org/tech-talk/telecom/internet/popular-internet-of-things-forecast-of-50-billion-devices-by-2020-is>-acesso em junho de 2018. 3
- [17] Hsu, Chin Lung e Judy Lin: *An empirical examination of consumer adoption of internet of things services: Network externalities and concern for information privacy perspectives*. *Computers in Human Behavior*, 62:516–527, setembro 2016. 3
- [18] *50 sensor applications for a smarter world*. http://www.libelium.com/resources/top_50_iot_sensor_applications_ranking/, acesso em junho de 2018. 3
- [19] Ramesh Jadhav, Akshay e P Rajalakshmi: *Iot enabled smart and secure power monitor*. páginas 1–4, julho 2017. 3
- [20] Chacón-Troya, D. P., O. O. González e P. C. Campoverde: *Domotic application for the monitoring and control of residential electrical loads*. Em *2017 IEEE 37th Central America and Panama Convention (CONCAPAN XXXVII)*, páginas 1–6, Nov 2017. 3
- [21] Gu, Shu Cheng, Ching Hung Cheng e Gwo Jia Jong: *The power monitor system combined green energy*. páginas 757–760, novembro 2013, ISBN 978-1-4799-0071-8. 3
- [22] Morimoto, N., Y. Fujita, M. Yoshida, H. Yoshimizu, M. Takiyamada, T. Akehi e M. Tanaka: *Smart outlet network for energy-aware services utilizing various sensor information*. Em *2013 27th International Conference on Advanced Information Networking and Applications Workshops*, páginas 1630–1635, March 2013. 3, 4

- [23] Ge, Binjie, Xin'an Wang, Zhang Xing, Xiaoxing Feng e Qingqin Wang: *Design and implementation of a novel programmable power monitor chip*. setembro 2010. 3
- [24] Lorek, M. C., F. Chraim e K. S. J. Pister: *Plug-through energy monitor for plug load electrical devices*. Em *2015 IEEE SENSORS*, páginas 1–4, Nov 2015. 3
- [25] Carvalho, Leticia: *Na volta às aulas, estudantes da unb enfrentam apagão em cursos noturnos*, 2017. <https://g1.globo.com/distrito-federal/noticia/na-volta-as-aulas-estudantes-da-unb-enfrentam-apagao-em-cursos-noturnos.html>, acesso em junho de 2018. 4
- [26] *Why you need a ups device?* <https://kb.abacuslaw.com/articles/321-Why-You-Need-a-UPS-Device>, acesso em junho de 2018. 4
- [27] *Queda de energia destrói material usado em pesquisa de alunos da unb*, 2016. <http://g1.globo.com/distrito-federal/noticia/2016/01/queda-de-energia-destrui-material-usado-em-pesquisa-de-alunos-da-unb.html>, acesso em junho de 2018. 4
- [28] Golkar, M. A.: *Electric power quality: types and measurements*. Em *2004 IEEE International Conference on Electric Utility Deregulation, Restructuring and Power Technologies. Proceedings*, volume 1, páginas 317–321 Vol.1, April 2004. 5
- [29] Rocha, Joaquim Eloir: *Qualidade da energia elétrica*, 2016. http://paginapessoal.utfpr.edu.br/joaquimrocha/qualidade-da-energia-eletrica/Qualidade%20da%20Energia%20Eletrica%20_%20Apostila.pdf/at_download/file, acesso em junho de 2018. 6
- [30] Cormen, T.H.: *Introduction to Algorithms, 3rd Edition*. MIT Press, 2009, ISBN 9780262033848. <https://books.google.com.br/books?id=VK9hPgAACAAJ>. 8
- [31] Tonin, Marcos Vinicius Prescendo: *Desenvolvimento do frontend de uma plataforma de aferimento de qualidade de energia*, 2018. 9
- [32] *Arduino uno*. <https://www.farnell.com/datasheets/1682209.pdf>, acesso em novembro de 2018. 12
- [33] *How to create a transformer using ltspice*. <http://web.cecs.pdx.edu/~tymerski/ece222/How%20to%20create%20a%20transformer%20using%20LTSpice.pdf>. 14
- [34] *1n4001 - 1n4007 general-purpose rectifiers*. <https://br.mouser.com/datasheet/2/308/1N4007-1305060.pdf>, acesso em novembro de 2018. 15, 17
- [35] *Lt1083/lt1084/lt1085 fixed 3a, 5a, 7.5a low dropout positive fixed regulators*, 2003. <https://www.analog.com/media/en/technical-documentation/data-sheets/1083ffe.pdf>, acesso em novembro de 2018. 16

- [36] *xx555 precision timers*, 2014. <http://www.ti.com/lit/ds/symlink/ne555.pdf>, acesso em novembro de 2018. 17
- [37] *Lmx58-n low-power, dual-operational amplifiers*, 2014. <http://www.ti.com/lit/ds/symlink/lm158-n.pdf>, acesso em novembro de 2018. 19, 24
- [38] Sedra, Adel S. e Kenneth C. Smith: *Microelectronic Circuits*. Oxford University Press, fifth edição, 2004. 19
- [39] *Atmega328p 8-bit avr microcontroller with 32k bytes in-system programmable flash datasheet*, 2105. http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7810-Automotive-Microcontrollers-ATmega328P_Datasheet.pdf, acesso em novembro de 2018. 19, 23, 25
- [40] *1 watt zener diodes*. http://www.baudaeletronica.com.br/Documentos/Zener_1N4728%20to%201N4764.pdf, acesso em novembro de 2018. 24
- [41] Sreedharan, Rahul: *Serial port programming on windows using win32 api and c language*, 2015. <https://github.com/xanthium-enterprises/Serial-Programming-Win32API-C>, acesso em novembro de 2018. 34
- [42] *Mqtt things and channels binding*, 2019. <https://www.openhab.org/addons/bindings/mqtt.generic/>, acesso em novembro de 2018. 36
- [43] *Pubsubclient*. <https://www.arduino-libraries.info/libraries/pub-sub-client>, acesso em novembro de 2018. 36
- [44] *Eclipse paho javascript client*, 2013. <https://www.eclipse.org/paho/clients/js/>, acesso em novembro de 2018. 39
- [45] *jquery core - all versions*. <https://code.jquery.com/jquery/>, acesso em novembro de 2018. 39
- [46] *Highcharts file service*. https://code.highcharts.com/?_ga=2.123715980.646424508.1562010321-732451812.1561747366, acesso em novembro de 2018. 39
- [47] *Net adapter ii externo kit gerenciador remoto web/snmp 64017 - sms*. https://www.pontofrio.com.br/Informatica/equipamentos-rede/AdaptadordeRede/net-adapter-ii-externo-kit-gerenciador-remoto-web-snmp-64017-sms-9234.html?utm_medium=cpc&utm_source=gp_pla&IdSku=9234578&idLojista=19709&utm_campaign=info_shopping&gclid=CjwKCAjwr8zoBRA0EiwANmvpYAXQTfhyERDdaDR_33n8eUI6CRc8obmFLZZjh2ozcGf6NJ1sr8ijcBoCn0IQAvD_BwE, acesso em junho de 2019. 65
- [48] *Simply smart monitoring. anywhere. everywhere*. https://store.serverscheck.com/cart_all.asp?cid=1008245688016441426&mop=, acesso em junho de 2019. 65