UNIVERSIDADE DO ESTADO DO AMAZONAS - UEA
ESCOLA SUPERIOR DE TECNOLOGIA
ENGENHARIA ELÉTRICA

KEVEN DE CASTRO GOMES

DESENVOLVIMENTO DE UM DISPOSITIVO ELETRÔNICO PARA A MEDIÇÃO DE DISTORÇÃO HARMÔNICA CONFORME A NORMA IEC 61000-4-7

Manaus
2019

KEVEN DE CASTRO GOMES

DESENVOLVIMENTO DE UM DISPOSITIVO ELETRÔNICO PARA A MEDIÇÃO DE DISTORÇÃO HARMÔNICA CONFORME A NORMA IEC 61000-4-7

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Orientador: Jozias Parente de Oliveira, Dr

Manaus
2019

Universidade do Estado do Amazonas – UEA
Escola Superior de Tecnologia - EST

Reitor:
Cleinaldo de Almeida Costa
Vice-Reitor:
Cleto Cavalcante de Souza Leal
Diretor da Escola Superior de Tecnologia:
Ingrid Sammyne Gadelha Figueiredo
Coordenador do Curso de Engenharia Elétrica:
Walfredo da Costa Lucena Filho

Banca Avaliadora composta por: *Data da defesa: 18/12/2019.*
Prof. Jozias Parente de Oliveira, Dr *(Orientador)*
Prof. José Gilson Siqueira, Dr
Prof. Karlo Homero Ferreira Santos, Ms

CIP – Catalogação na Publicação

Gomes, Keven de Castro

Desenvolvimento de um dispositivo eletrônico para a medição de distorção harmônica conforme a norma iec 61000-4-7/ Keven de Castro Gomes; [orientado por] Jozias Parente de Oliveira. – Manaus: 2019.
64 p.: il.

Trabalho de Conclusão de Curso (Graduação em Engenharia Elétrica). Universidade do Estado do Amazonas, 2019.

1. Distorção Harmônica. 2. Transformada rápida de Fourier. 3. Microcontrolador. I. Oliveira, Jozias Parente de.

KEVEN DE CASTRO GOMES

DESENVOLVIMENTO DE UM DISPOSITIVO ELETRÔNICO PARA A MEDIÇÃO DE DISTORÇÃO HARMÔNICA CONFORME A NORMA IEC 61000-4-7

Pesquisa desenvolvida durante a disciplina de Trabalho de Conclusão de Curso II e apresentada à banca avaliadora do Curso de Engenharia Elétrica da Escola Superior de Tecnologia da Universidade do Estado do Amazonas, como pré-requisito para a obtenção do título de Engenheiro Eletricista.

Nota obtida: _____ (_____)

Aprovada em _____/_____/_____.

Área de concentração: Eficiência Energética

BANCA EXAMINADORA

_____
Orientador: Jozias Parente de Oliveira, Dr.

_____
Avaliador: José Gilson Siqueira, Dr.

_____
Avaliador: Karlo Homero Ferreira Santos, Ms.

Manaus 2019

Dedicatória

Aos meus pais, por terem dedicado suas vidas para me tornar a pessoa que sou hoje. Agradeço imensamente por todo o amor e por todo o estímulo para chegar a este momento tão importante. Dedico-lhes essa conquista, como forma de gratidão.

Agradecimentos

Agradeço aos meus pais, por todo o apoio e incentivo.
Agradeço ao meu orientador Jozias, por terem me ajudado com tanta dedicação e paciência.
Agradeço aos meus amigos Claudionor, Joaci e Keven por todos os ensinamentos ao longo dos anos.

RESUMO

A distorção harmônica em sistemas elétricos acaba afetando diretamente a qualidade de energia no qual as cargas serão alimentadas. Um dos efeitos dos harmônicos em motores e geradores elétricos, que são máquinas rotativas (indução e sincronia), seria o aumento do aquecimento devido ao aumento das perdas no ferro e no cobre, além do aumento do ruído audível. A proposta deste projeto é desenvolver um protótipo que seja capaz de medir harmônicas em tempo real, de acordo com a norma IEC 61000-4-7 – Técnicas de medição e ensaio – Guia geral sobre medições de harmônicas e inter-harmônicas e instrumentação, para sistemas de fornecimento de energia e equipamentos conectados a eles. Para isso, será utilizado o circuito integrado de monitoramento de qualidade de energia - ADE9000, que será responsável por fazer a amostragem de 12 ciclos de 60Hz com 128 amostras por ciclo, e um microcontrolador, que se comunicará via SPI com o ADE9000 para obtenção da forma de onda e fazer o processamento da transformada rápida de Fourier, obtendo as raias de tensão e corrente trifásica e enviando para o computador, onde será mostrado os gráficos de harmônicas.

Palavras-Chave: Distorção Harmônica, Transformada Rápida de Fourier, Microcontrolador.

ABSTRACT

The harmonic distortion affects directly the power quality, which loads will be fed. Harmonic damages in motors and electric generators, which are rotating machines (induction and synchrony), are the increase in the heating due to the increase of losses on iron and copper, besides the increase of audible volume. The purpose of this project is to develop an prototype capable of measuring harmonics in real time according to IEC 61000-4-7 rule - Testing and measurement techniques - General guide on harmonics and interharmonics measurements and instrumentation, for power supply systems and equipment connected thereto. For this project, a power quality monitoring integrated circuit - ADE9000 will be used, which will be responsible for sampling 12 cycles of 60Hz with 128 samples per cycle, and a microcontroller, which will communicate via SPI with the ADE9000 to obtain the waveform and make the fast Fourier transform processing, obtaining the harmonics of the three-phase voltage and current, sending to the computer, where the harmonic graphs will be shown.

Palavras-Chave: Harmonic Distortion, Fast Fourier Transform, Microcontroller.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

O tema deste trabalho é desenvolvimento de um dispositivo eletrônico para a medição de distorção harmônica conforme a norma IEC 61000-4-7.

Atualmente, para que o Sistema Elétrico de Potência (SEP) opere de forma satisfatória, sem prejuízo de desempenho, é necessário que a qualidade da energia elétrica atenda a critérios como: valor eficaz, frequência, distorções na forma de onda, entre outros (AFONSO & MARTINS, 2003).

A perda da qualidade da energia é considerada, pelos especialistas no assunto, como um desvio na forma de onda, na amplitude e na frequência da tensão e/ou da corrente elétrica. Sendo que esses desvios podem ocorrer simultaneamente ou não, podendo resultar em uma operação indevida de equipamentos ou falhas nos mesmos (PAULILO, 2013).

Equipamentos e fenômenos na eletrônica de potência, tais como cargas não lineares, conversores, fornos a arco, dentre outros, produzem distorção no sinal elétrico, devido aos surgimentos de componentes não almejáveis no sistema. Componentes como essas, podem ser definidas como harmônicos e inter-harmônicos.

Uma distorção de forma de onda é dita harmônica quando a deformação se apresenta de forma similar em cada ciclo da frequência fundamental. Neste caso, o espectro contém apenas frequências múltiplas inteiras da fundamental. Esse tipo de deformação geralmente é imposto por dispositivos que apresentam relação não linear entre tensão e corrente como, por exemplo, transformadores e motores, cujos núcleos ferromagnéticos são sujeitos à saturação. Outros elementos não lineares são as cargas eletrônicas que produzem descontinuidades na corrente devido ao chaveamento dos conversores (PHIPPS, NELSON, & SEN, 1994).

Cargas que, além de serem não lineares, também variam ao longo do tempo produzem distorções variáveis no tempo, o que pode levar ao aparecimento de frequências inter-harmônicas, além de harmônicas moduladas. É o caso de fornos a arco e compensadores reativos controlados por tiristores. Por esse motivo e por sua elevada potência (dezenas de MW) os fornos elétricos a arco são considerados cargas problemáticas para a operação de sistemas elétricos (POMILIO & DECKMANN, 1997).

Estes componentes são indesejáveis no sistema uma vez que podem reduzir a vida útil de equipamentos (como transformador, relés e fusíveis) devido ao sobreaquecimento, ou de motores, ocasionando torques indevidos. Podem comprometer o funcionamento de equipamentos eletrônicos sensíveis a distorções na forma de onda de tensão. Podem provocar também a ocorrência de ressonância dos capacitores, além de outros problemas (LI, XU, & TAYJASANANT, 2003).

Por isso, é importante o monitoramento destes componentes no sistema elétrico a fim de detectá-los e consequentemente eliminá-los ou reduzi-los. Isso proporciona maior tempo de vida útil aos dispositivos, bem como o melhor funcionamento dos mesmos.

Com o intuito de identificar a existência destes componentes, muitas técnicas têm sido propostas para monitorar o valor dos mesmos no sistema de potência. Dentre os métodos empregados, a Transformada Rápida de Fourier do inglês *Fast Fourier Transform* é o mais utilizado. Esta ferramenta é utilizada para a monitoração do sistema elétrico de potência por causa da sua eficiência computacional.

A distorção harmônica em sistemas elétricos acaba afetando diretamente a qualidade de energia no qual as cargas serão alimentadas. Entretanto, os medidores de distorção harmônica que seguem a norma IEC 61000-4-7 são extremamente caros, além de não poderem ser integrados a outros sistemas embarcados de medição da qualidade de energia.

Este trabalho tem por hipótese a ideia de que é possível desenvolver um sistema que seja capaz de medir harmônicas em tempo real com base na norma IEC 61000-4-7, utilizando um circuito integrado de monitoramento de qualidade de energia - ADE9000, juntamente com um microcontrolador que será responsável pela aquisição dos dados provenientes do CI ADE9000, além de com o mesmo microcontrolador, ser possível realizar a transformada rápida de Fourier, resultando em dados de frequência harmônicas, que poderão ser acessados por um computador (ou qualquer outro sistema de medição da qualidade de energia).

Os objetivos deste trabalho consistem em: desenvolver um circuito de condicionamento do sinal elétrico; configurar o ADE9000 para amostrar as formas de onda do sinal trifásico; realizar a transferência das formas de onda do ADE9000 para o microcontrolador; calcular a transformada rápida de Fourier no microcontrolador; utilizar uma comunicação serial para a transferência dos dados do microcontrolador

para o computador; criar uma interface gráfica para a visualização dos resultados no computador.

O desenvolvimento de um sistema capaz de efetuar a medição de harmônicas em uma rede elétrica se mostra importante no âmbito industrial, visto que torna possível localizar equipamentos que as geram, abrindo margem para o cálculo de filtros que reduzem as harmônicas.

Além disso, por se tratar de um projeto de eficiência energética, que engloba diversas áreas dentro da Engenharia Elétrica, tais como Eletrônica de Potência para o condicionamento do sinal provido da rede elétrica, Sistemas Microcontrolados na utilização de um microcontrolador para processar os dados, assim como Processamento Digital de Sinais com o cálculo da transformada rápida de Fourier visando a obtenção das componentes harmônicas.

Para ordenação dos assuntos a serem abordados de forma clara e objetiva, este trabalho está dividido em 4 capítulos, além das referências.

Capítulo 1 – Referencial Teórico: apresenta conceito fundamentais, transformada de Fourier de tempo discreto, definições da norma IEC 61000-4-7, princípios básicos de sistemas microcontrolados e o circuito de medição trifásica ADE9000.

Capítulo 2 – Metodologia: neste capítulo é descrito os dispositivos e softwares utilizados no projeto e o funcionamento do projeto.

Capítulo 3 – Implementação do projeto: descreve os procedimentos realizados durante a implementação do projeto, mostrando cada passo seguido.

Capítulo 4 – Resultados: descreve os resultados obtidos decorrente da análise, gerando informações necessárias para a conclusão que é apresentada no fim do trabalho.

1 REFERENCIAL TEÓRICO

1.1 ANÁLISE DE FOURIER

1.1.1 Transformada de Fourier de Tempo Discreto

A análise de Fourier consiste na análise de sinais no domínio da frequência. Para os sinais serem analisados neste domínio, é necessário que haja um mecanismo de transformação e, para isso, é utilizada a transformada de Fourier de tempo discreto (DFT, do inglês *Discrete Fourier Transform*). Basicamente, a transformada de Fourier de tempo discreto serve tanto para sinais periódicos, quanto para sinais aperiódicos. Todavia, existe uma generalização para sinais periódicos chamada de Série de Fourier.

Na representação do espectro de sinais periódicos, a série de Fourier requer apenas uma parte do sinal, ou seja, a parte de um período igual a N. Com os valores das amostras nesse intervalo de tempo, é possível montar os coeficientes espectrais da série de Fourier, já que para esse tipo de sinais, temos um espectro também discreto (OPPENHEIM & WILLSKY, 2010).

$$a_k = \frac{1}{N}\sum_{n=0}^{N-1} x[n]e^{-jkw_0 n} \quad (1)$$

Sendo:

a, o valor no domínio da frequência;

k, o coeficiente de frequência;

N, o período do sinal;

n, os valores de tempo discreto;

x, o sinal analisado;

w_0, a frequência fundamental;

Com a equação 1, é possível fazer a representação de todos os valores de a_k, onde k é o que multiplica a frequência fundamental do sinal. Algo interessante sobre essa representação é que, já que temos um sinal $x[n]$ periódico com período N, o a_k também será periódico com o mesmo período N.

Uma vez que temos uma representação no domínio da frequência, é importante que exista alguma maneira de recuperar o sinal no domínio do tempo, para isso, existe

a operação inversa, em que o sinal no domínio do tempo pode ser expresso como uma combinação linear das exponenciais complexas, sendo que o valor que as multiplicam são exatamente os nossos coeficientes espectrais.

$$x[n] = \sum_{k=0}^{N-1} a_k e^{jkw_0 n} \quad (2)$$

Sendo:

x, o sinal recuperado;

n, os valores de tempo discreto;

N, o período do sinal;

a, o valor no domínio da frequência;

k, o coeficiente de frequência;

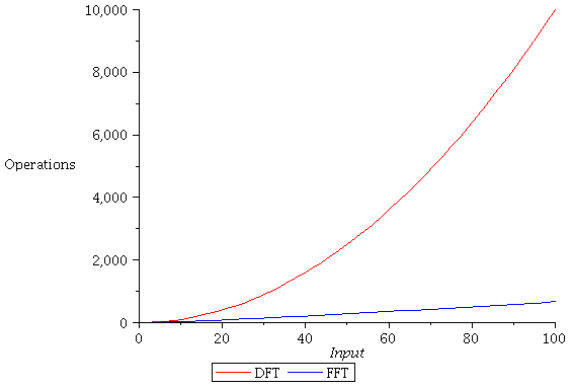
w_0, a frequência fundamental;

1.1.2 Transformada Rápida de Fourier

O algoritmo da transformada rápida de Fourier foi desenvolvida com o intuito de calcular a transformada discreta de Fourier de forma a utilizar menos recursos computacionais. A transformada rápida de Fourier (FFT, do inglês *Fast Fourier Transform*) reaproveita resultados computados anteriormente, reduzindo o número total de operações aritméticas, melhorando significativamente a sua eficiência.

Os algoritmos que utilizam a FFT utilizam a simetria e a periodicidade das funções trigonométricas para calcular a transformada discreta de Fourier. Com isso, a quantidade de operações é reduzida de N^2 operações para $N \log_2 N$, comparativo na Figura 1. No caso de implementações em sistemas embarcados, onde o poder computacional é limitado, a FFT é extremamente necessária.

Figura 1: Comparação entre Número de Operações da DFT e FFT



Fonte: (PAULILO, 2013)

1.1.3 Harmônicos na Rede Elétrica

A tensão e corrente elétrica alternada, é representada por uma forma de onda senoidal que, no domínio do tempo, pode ser representada pela equação 3 e equação 4, respectivamente.

$$v(t) = V \sin(2\pi f_0 t) \tag{3}$$

$$i(t) = I \sin(2\pi f_0 t \pm \phi) \tag{4}$$

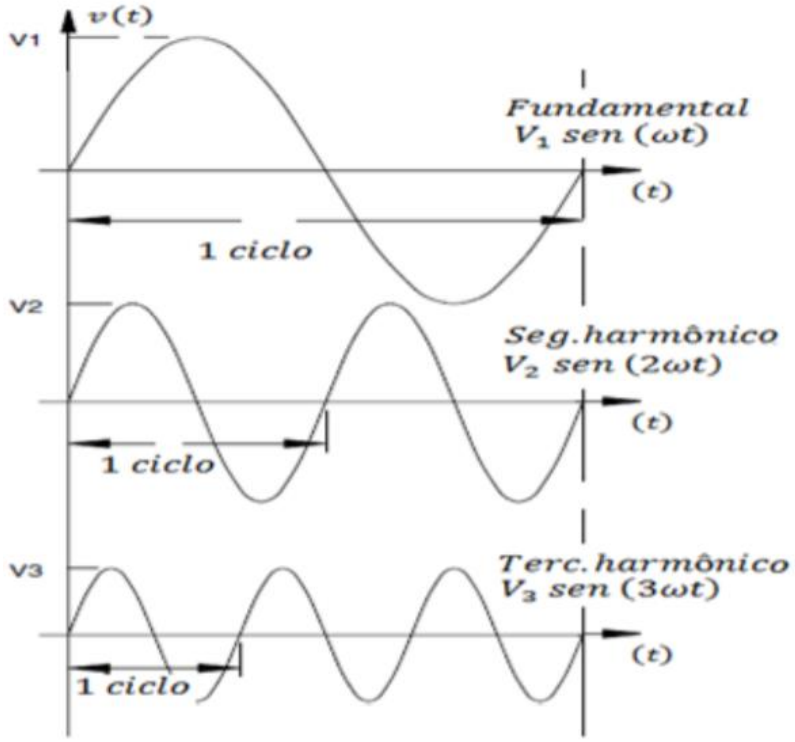
Sendo:

f_0, a frequência fundamental;

ϕ, o ângulo de defasagem entre a tensão e corrente.

Os harmônicos são componentes senoidais de um sinal periódico, na qual sua frequência é um múltiplo inteiro da frequência fundamental desse sinal, como na Figura 2.

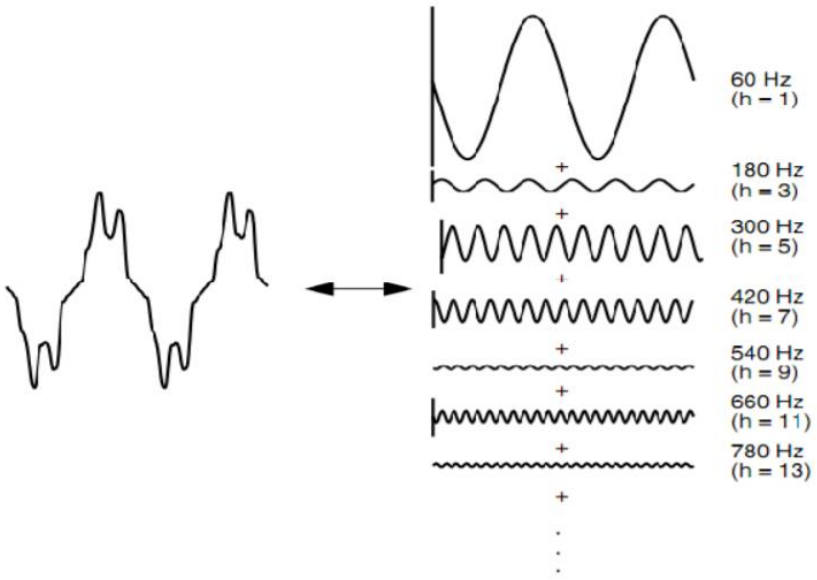
Figura 2: Fundamental, Segundo e Terceiro Harmônico



Fonte: (LI, XU, & TAYJASANANT, 2003)

Quando há presença de harmônicos no sinal da rede elétrica, é possível ver a distorção na onda, uma vez que os harmônicos estão somados com a senoide na frequência fundamental conforme na Figura 3, sendo possível fazer a obtenção desses dados para realizar a FFT.

Figura 3: Sinal Resultante com Soma de Harmônicas



Fonte: (LI, XU, & TAYJASANANT, 2003)

1.1.4 Distorção Harmônica Total

Para indicar o conteúdo harmônico de um sinal da rede elétrica, é utilizado o índice distorção harmônica total (DHT). A partir deste índice, é possível caracterizar o sinal baseado nas distorções apresentadas nele. A DHT é uma medida do valor das componentes harmônicas em relação a componente fundamental. Quando a DHT for igual a zero, o sinal não possui distorções harmônicas. A equação 5 representa a Distorção Harmônica Total.

$$DHT = \frac{\sqrt{\sum_{h=2}^{h_{max}} \left(G_{h_{RMS}}\right)^2}}{G_{1_{RMS}}} \tag{5}$$

Sendo:

$G_{h_{RMS}}$, a amplitude da harmônica de ordem h;

$G_{1_{RMS}}$, a amplitude da componente fundamental;

h, a ordem harmônica considerada no cálculo;

h_{max}, a ordem da maior componente harmônica considerada.

1.2 IEC 61000-4-7

A IEC (*Internacional Electrotechnical Commission*) 61000-4-7 é uma norma estabelecida para a detecção de componentes de frequência contidos em sinais de sistemas de potência. Nesta norma são considerados alguns métodos, denominados como grupos e subgrupos de harmônicos e inter-harmônicos, que têm como função agrupar a energia dos componentes espalhados ao longo do espectro de frequência e estabelecer sua respectiva amplitude. As equações 6 e 7 representam os cálculos dos grupos de harmônicos e subgrupos de harmônicos, respectivamente, para sistemas de 60Hz. (IEC-61000-4-7, 2002)

$$G_{g,h}^2 = \frac{C_{k_h-6}^2}{2} + \sum_{i=-5}^{5} C_{k_h+i}^2 + \frac{C_{k_h+6}^2}{2} \tag{6}$$

$$G_{sg,h}^2 = \sum_{i=-1}^{1} C_{k_h+i}^2 \tag{7}$$

Sendo:

$G_{g,h}^2$, o valor quadrático dos grupos harmônicos;

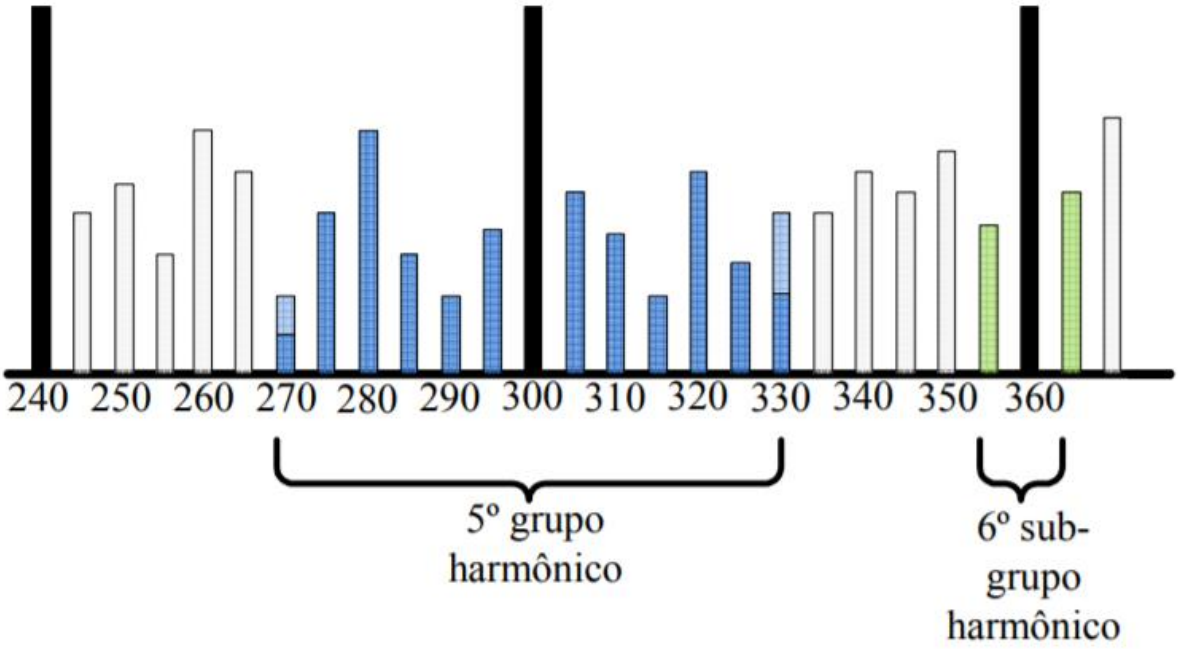
$G_{sg,h}^2$, o valor quadrático dos subgrupos harmônicos;

k_h, a ordem harmônica considerada no cálculo;

C, o valor no domínio da frequência;

A partir disto, a divisão desses grupos e subgrupos é representada na Figura 4. No quinto grupo harmônico, é possível visualizar o agrupamento de energia contida entre as frequências de 270Hz e 330Hz, sendo que as frequências das extremidades pertencem a outra harmônica também, por isso é apenas considerada metade no cálculo. Já no sexto subgrupo, é apenas considerada as duas frequências adjacentes a frequência fundamental, no caso da harmônica de 360Hz.

Figura 4: Representação dos Grupos e Subgrupos de Harmônicos



Fonte: (IEC-61000-4-7, 2002)

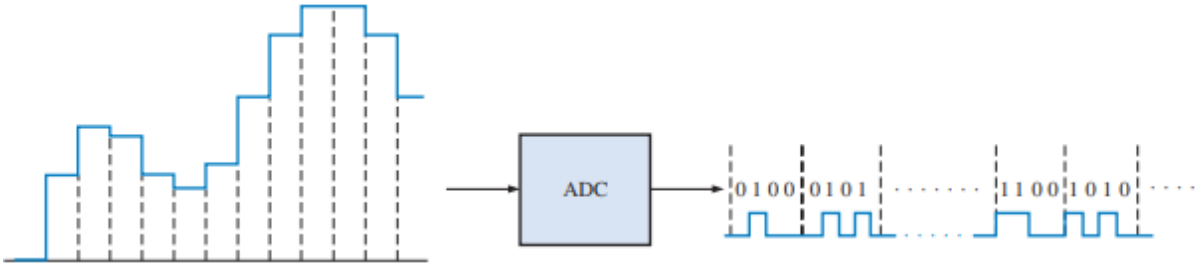
Para a realização destes grupos a norma IEC 61000-4-7 propõe a utilização de 12 ciclos do sinal, o que corresponde a uma resolução de frequência de 5 Hz e deve ser aplicada para detectar até o 50º harmônico.

1.3 SISTEMAS MICROCONTROLADOS

1.3.1 Conversor Analógico-Digital

A conversão analógico-digital é o processo de conversão da saída do circuito de amostragem e retenção em uma série de códigos binários que representam a amplitude do sinal de entrada analógico em cada instante amostrado. O processo de amostragem e retenção mantém a amplitude do sinal de entrada analógico constante entre os pulsos de amostragem; portanto, a conversão analógico-digital pode ser feita usando um valor constante em vez de um sinal analógico que varia durante o intervalo de conversão, o qual corresponde ao tempo entre os pulsos de amostragem. A Figura 5 ilustra a função básica de um conversor analógico-digital (ADC). Os intervalos de amostragem são indicados por linhas tracejadas (FLOYD, 2006).

Figura 5: Conversão Analógico-Digital



Fonte: (FLOYD, 2006)

1.3.2 Protocolo SPI

Levando em consideração que os recursos disponíveis em um microcontrolador são limitados, por vezes há necessidade de os expandir. Assim, existem circuitos integrados com as mais variadas funções: memórias *EEPROM*, *shift registers*, conversores A/D. Para efetuar a comunicação com estes periféricos é necessário um protocolo de comunicação para servir como interface para a comunicação. Dos muitos protocolos disponíveis, um que foi massivamente adotado para comunicação entre microcontroladores e periféricos externos foi o SPI (*Serial Peripheral Interface*) (ROCHA, 2007).

O protocolo SPI é um protocolo série que permite transmissão síncrona bidirecional de 8 bits. O SPI necessita de um *master* que controle o sinal de *clock*. Por conta do protocolo SPI ser bidirecional, o *master* pode enviar ou receber informação dos *slaves*, sendo que o mesmo não controla com quem irá se comunicar.

Para determinar o destinatário da informação é necessária uma ligação direta a cada um dos *slaves*, que será a ligação que conecta os pinos SS (*Slave Select*). Assim, quando a informação tiver como destinatário o *slave* 1, o pino SS1 deve ser ativado para que este aceite a informação.

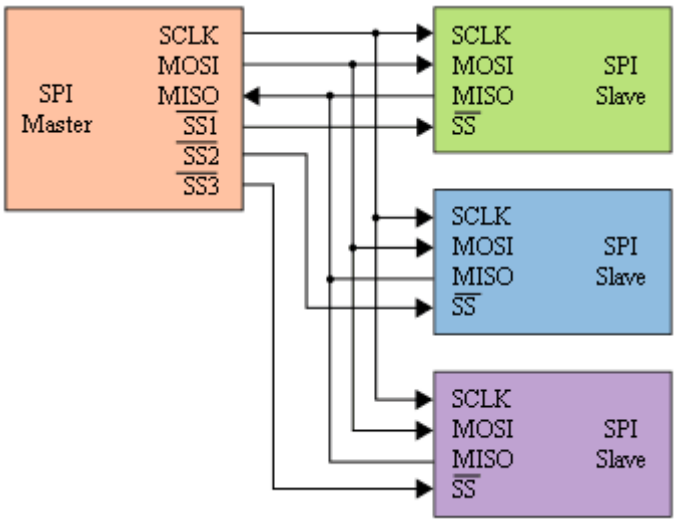
Os pinos básicos de comunicação entre dispositivos SPI se encontram na Tabela 1 e o esquema padrão de ligação na Figura 6.

Tabela 1: Pinos da Comunicação SPI

Pino	Nome Padrão	Significado
Do Master para o Slave	MOSI	Master Output Slave Input
Do Slave para o Master	MISO	Master Input Slave Output
Clock	SCLK	Serial Clock
Seleção de Slave	SS	Slave Select

Fonte: (ROCHA, 2007)

Figura 6: Esquema Padrão da Comunicação SPI



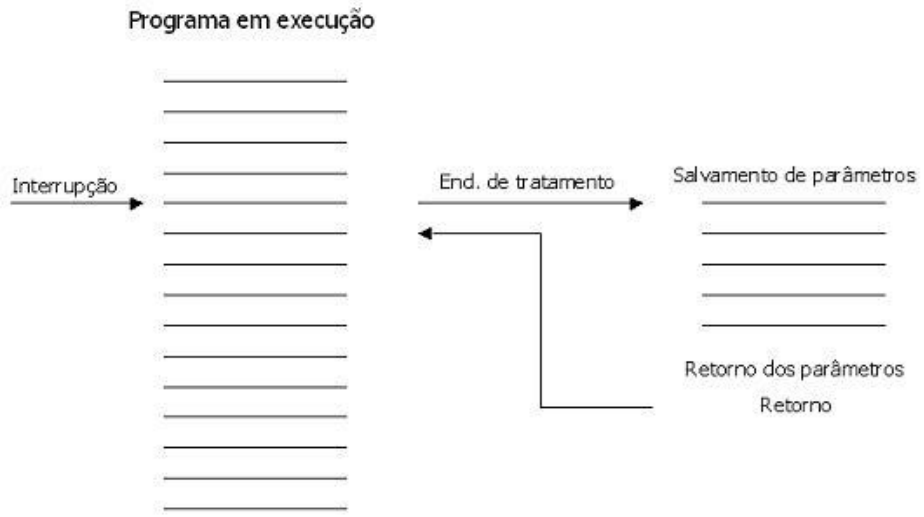
Fonte: (ROCHA, 2007)

1.3.3 Interrupções

Uma interrupção é um evento externo que força a suspensão da execução do programa corrente, desviando a execução para um bloco de código chamado rotina de interrupção. Ao terminar o tratamento de interrupção, o controle retorna ao programa interrompido exatamente no mesmo estado em que estava quando ocorreu a interrupção.

A requisição de interrupção pode ocorrer a qualquer momento (assincronamente), sendo ela indicada pela ativação de um *flag* pelo dispositivo periférico. O microprocessador reconhece a interrupção, enviando sinais de controle, completa a execução da instrução corrente, salva o conteúdo dos registradores de interesse (contador de programa, status, etc.), e atende ao dispositivo periférico que solicitou a interrupção, transferindo o controle para a rotina de tratamento da interrupção. Ao término da execução desta rotina, o microprocessador desativa o *flag* de indicação de interrupção, restaura os registradores que foram salvos, e transfere o controle para a instrução seguinte ao ponto de interrupção do programa, conforme mostrado na Figura 7 (KINOSHITA, CUGNASCA, & HIRAKAWA, 2004).

Figura 7: Funcionamento da Interrupção



Fonte: (KINOSHITA, CUGNASCA, & HIRAKAWA, 2004)

1.3.4 Transferência Direta à Memória

Como mostrado, a interrupção libera a CPU de tarefa de aguardar por ocorrência de evento de E/S. Entretanto a CPU ainda continua sendo responsável por

realizar essa transferência de dados. Para liberar a CPU totalmente desta tarefa, é necessário utilizar a transferência direta à memória (DMA).

A transferência direta à memória utiliza outro dispositivo controlador, no qual já existem em microcontroladores modernos, que será encarregado de fazer a transferência de dados, deixando a CPU livre para realizar outros tipos de tarefa que consomem mais processamento. Existem 3 passos em uma transferência DMA:

1. CPU programa o controlador de DMA com a identificação do dispositivo solicitante, operação a ser realizada no dispositivo (escrita/leitura), endereço de origem e destino (na memória principal e no periférico) e o número de bytes a serem transmitidos;
2. O controlador de DMA assume o barramento, iniciando a operação. A transferência inicia quando o dado está disponível (no dispositivo ou memória). O controlador de DMA fornece o endereço de leitura ou escrita na memória. Se a requisição necessita mais de uma transferência no barramento, o DMA gera o próximo endereço de memória. Por meio deste mecanismo o DMA pode transferir milhares de bytes do disco para a memória, sem interromper a CPU;
3. Concluída a transferência de DMA, o controlador interrompe o processador, que poderá verificar se a operação foi realizada com sucesso examinando a memória ou interrogando o controlador de DMA.

1.4 ADE9000

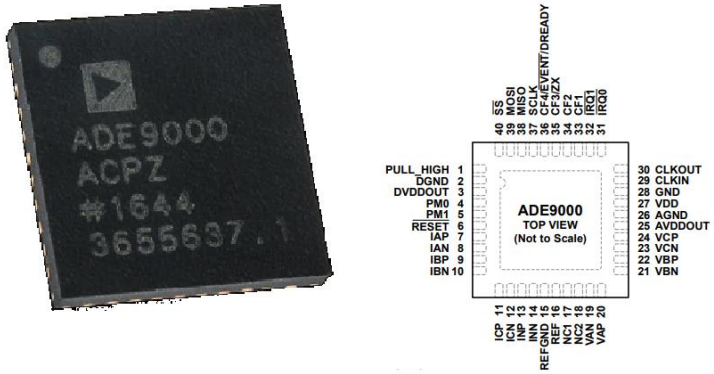
Segundo o datasheet do ADE9000:

> O ADE9000 é um CI medidor trifásico de energia elétrica de alta precisão que possui uma interface de comunicação serial e duas saídas de pulsos. O ADE9000 possui ADCs do tipo Sigma Delta, integradores digitais, circuito de referência, sensor de temperatura e todo o processamento de sinal requerido para executar a medição de energias ativa, reativa e aparente efetivos, além de oferecer um buffer de forma de onda flexível e integrado (ANALOG-DEVICES, 2017).

A Figura 8 mostra o encapsulamento e a pinagem desse circuito integrado, que apresenta canais de tensão e corrente que podem ter seus valores lidos por meio de registradores próprios. Além disso, apresenta também conversores analógico-digitais

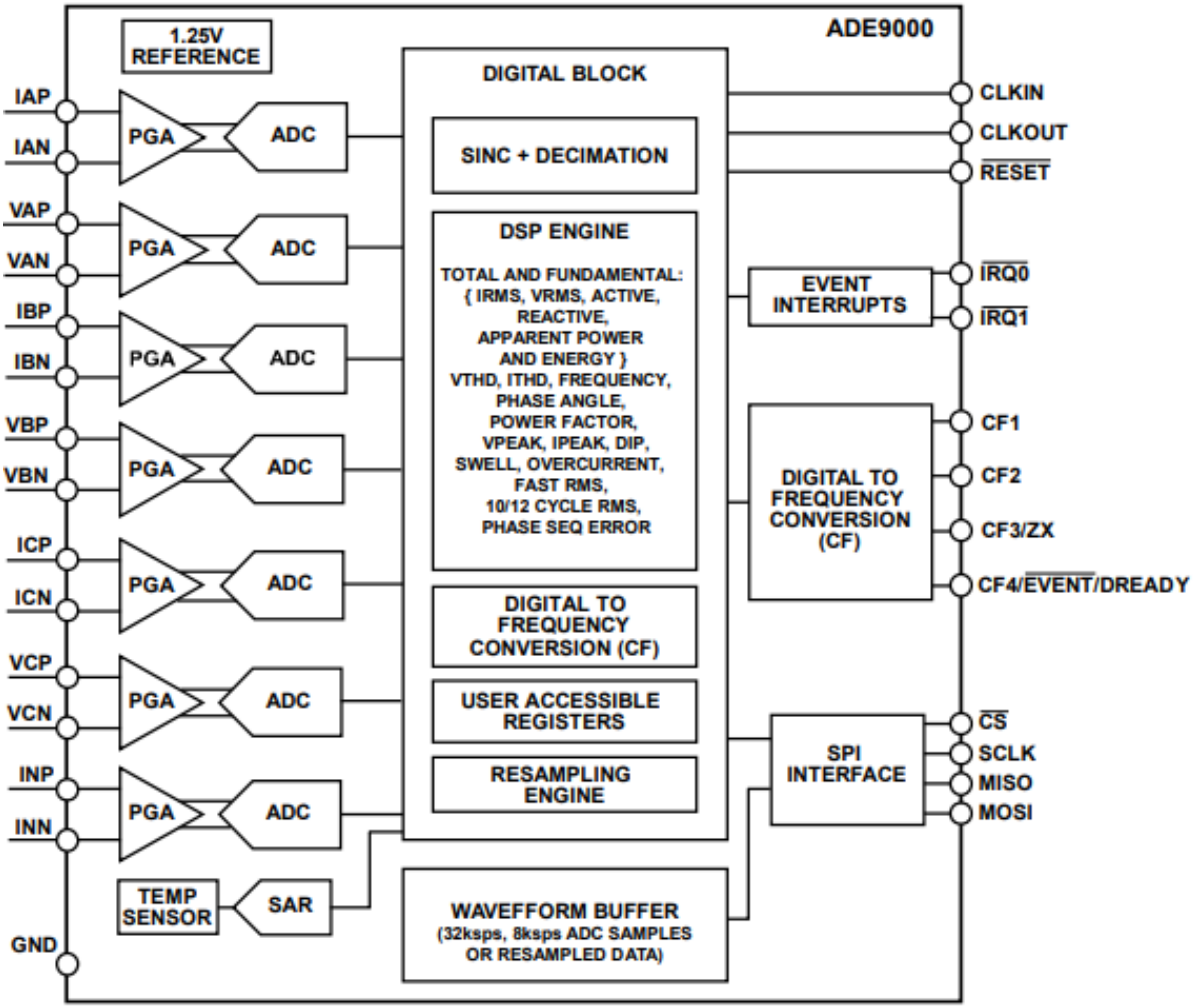
responsáveis por digitalizar as grandezas elétricas lidas e realiza todo o processamento de sinal necessário, conforme o diagrama em blocos apresentados na Figura 9.

Figura 8: Encapsulamento e pinagem do ADE9000



Fonte: (ANALOG-DEVICES, 2017)

Figura 9: Diagrama de Blocos Funcional



Fonte: (ANALOG-DEVICES, 2017)

1.4.1 Buffer de Forma de Onda

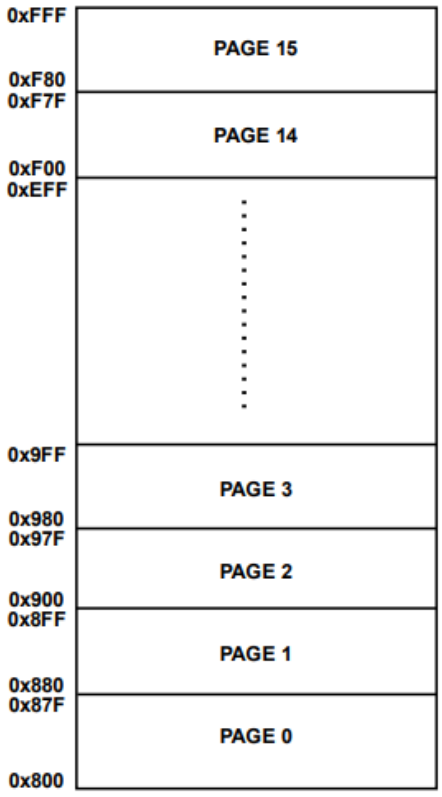
O ADE9000 possui um *buffer* de forma de onda que é composto por 2048, 32-bits de localizações de memória com endereços de 0x800 à 0xFFF. Os dados do *buffer* de forma de onda podem ser originados de quatro diferentes entradas do chip:

1. Saídas sinc4, providas de uma taxa de atualização de 32ksps.
2. Saídas sinc4+, providas de uma taxa de atualização de 8ksps.
3. Tensão e corrente são processadas pelo processador digital de sinais a uma taxa de atualização de 8ksps.
4. Formas de onda reamostradas com 128 pontos por linha de ciclo processadas pelo processador digital de sinais.

Há 256 (2048/8) grupos de amostras que podem ser armazenadas no *buffer*. No caso de uma taxa de atualização de 32ksps, o *buffer* contém 8ms de dados, já na taxa de 8ksps, possui 32ms de dados.

Quando utilizado uma taxa de atualização fixa, o *buffer* de forma de onda é dividido em 16 páginas, de 0 a 15. Cada página contém 128, 32-bits localizações de memórias conforme a Figura 10.

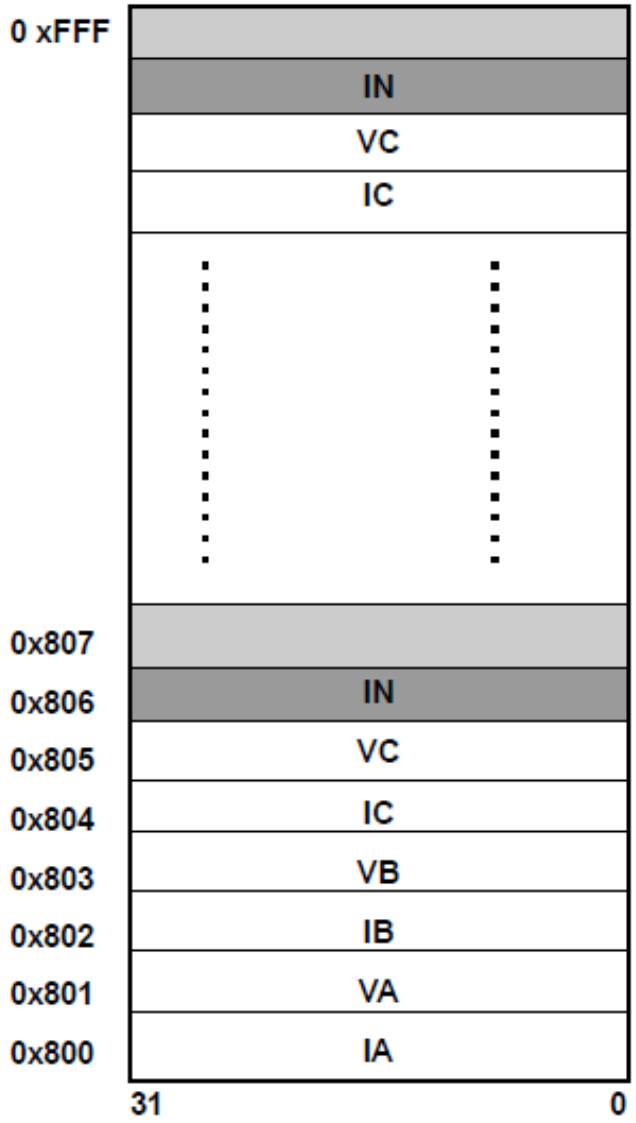
Figura 10: Arranjo de páginas do *buffer* de forma de onda



Fonte: (ANALOG-DEVICES, 2017)

A Figura 11 mostra como um *buffer* de taxa de atualização fixa é armazenado. Cada grupo de amostra é separado por uma célula de reposição que não tem dados.

Figura 11: *Buffer* de forma de onda com taxa de atualização fixa



Fonte: (ANALOG-DEVICES, 2017)

O *buffer* de forma de onda oferece dois tipos de preenchimentos para o uso de taxas de atualizações constantes: Parar quando o *buffer* está cheio e continuar preenchendo.

Além disso, o ADE9000 permite a seleção de eventos que geram uma interrupção no caso do modo de continuar preenchendo. É possível configurar para que haja uma interrupção quando o ADE9000 termine de preencher uma página específica, guardando o valor da última página preenchida em um registrador que pode ser acessado via SPI.

2 METODOLOGIA

O Trabalho apresentado é uma Pesquisa Aplicada, e tem como objetivo a realização de Pesquisa exploratória sobre o material bibliográfico e de laboratório. Os procedimentos técnicos de pesquisa e bibliografia experimental são utilizados. O método de abordagem hipotético-dedutivo e o método de procedimento monográfico são utilizados em sua elaboração. A coleta de dados é feita através da observação direta intensiva e documentação indireta, sendo estes dados qualitativos e interpretados de forma global.

Para a implementação deste projeto, foram adotados os seguintes passos:

1. Estudo sobre as especificações da norma IEC 61000-4-7 com o intuito de designar a quantidade necessária de amostras para ser realizado o cálculo da FFT.
2. Estudo do protocolo de comunicação do ADE9000, para que o sistema seja capaz de configurar o circuito integrado e, por conseguinte, obter as amostras de tensão e corrente trifásica.
3. Desenvolvimento no microcontrolador, dos processos de transferência de amostras do ADE9000, cálculo da FFT e distorção harmônica, e de uma comunicação serial para a transferência dos dados para um computador.
4. Desenvolvimento de uma interface gráfica no computador, com a finalidade de apresentar os gráficos de forma visual, juntamente com a validação dos dados de harmônicas.

2.1 ESPECIFICAÇÃO DOS DISPOSITIVOS E SOFTWARES UTILIZADOS NO PROJETO

Este tópico detalha os dispositivos utilizados para a geração de sinais, medição das formas de onda equivalentes e o microcontrolador utilizado. Também serão citados os softwares utilizados para realização dos cálculos apresentação dos gráficos de forma visual.

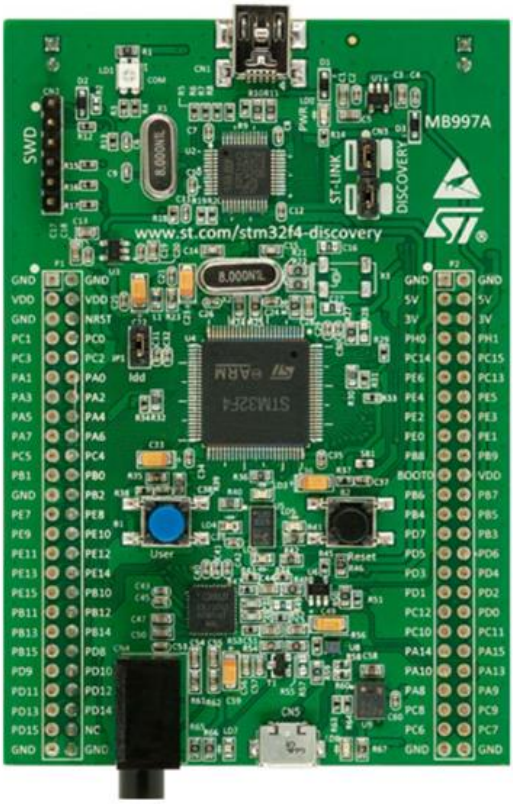
2.1.1 Microcontrolador STM34F4-Discovery

O kit STM32F4-discovery, ilustrado na figura Figura 12, permite que os usuários desenvolvam facilmente aplicativos com o microcontrolador de alto desempenho STM32F407VG com o núcleo de 32 bits ARM Cortex-M4. Inclui tudo o que é necessário para iniciantes ou para usuários experientes começarem rapidamente (ST, 2017).

Baseado no STM32F407VG, ele inclui uma ferramenta de depuração embutida ST-LINK/V2 ou ST-LINK/V2-A, dois acelerômetros digitais ST-MEMS, um microfone digital, um DAC de áudio com driver de alto-falante de classe D integrado, LEDs, botões e um conector micro-AB USB OTG (ST, 2017).

Por se tratar de um microcontrolador robusto, todos os periféricos são normalmente desativados. Por conta disso, para utilizar algum desses periféricos, é necessário ativar seus respectivos clocks. Além disso, todos os recursos são definidos em formado de Struct, uma vez que a linguagem de programação utilizada para programar o STM32F4-discovery é o C.

Figura 12: Microcontrolador STM34F4-Discovery



Fonte: (ST, 2017)

2.1.2 Gerador de Função Digital ITGFB-2002

Para a geração dos sinais necessários nos ensaios, foi utilizado o gerador de função digital ITGFB-2002, ilustrado na Figura 13. De acordo com o manual, este gerador possui 3 características principais (INSTRUTEMP, 2010):

1) Fácil e simples de utilizar, com processador único para controle das funções e exibição no display;
2) Ampla escala do gerador de função com alta precisão integrada e pontas de teste para alto desempenho;
3) Desenvolvido com tecnologia que assegura alta confiabilidade e estabilidade em seu funcionamento.

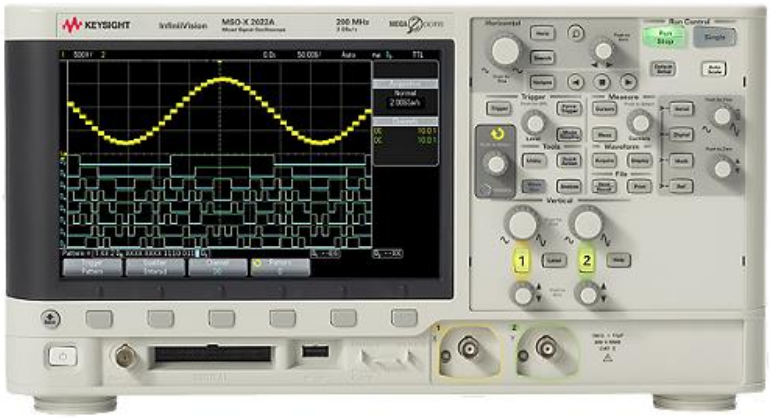
Figura 13: Gerador de função digital ITGFB-2002

Fonte: (INSTRUTEMP, 2010)

2.1.3 Osciloscópio Digital DSO-X 2012A

Para a visualização analógica das formas de onda injetadas no sistema, foi utilizado o Osciloscópio Digital DSO-X 2012A, ilustrado na Figura 14. De acordo com o manual, o osciloscópio possui uma largura de banda de 100 Mhz, 2 canais, uma memória máxima de 1 Mpts, uma taxa de amostragem de 2 GSa/s e 8 bits de ADC. Além disso, o mesmo possui uma interface para *flash drive* que pode ser usada para salvar em formato de imagem, o que está sendo visto no visor (KEYSIGHT, 2012).

Figura 14: Osciloscópio Digital DSO-X 2012A



Fonte: (KEYSIGHT, 2012)

2.1.4 Matlab

Para realizar a aquisição dos dados da serial foi utilizada a linguagem de programação de Matlab, símbolo representando na Figura 15, que consiste em uma ferramenta matemática poderosa, que possui módulos de comunicação serial, além de existir funções nativas para plotar os gráficos do espectro de frequência em tempo real. A utilização dessas tecnologias se deu devido a familiaridade do autor com a linguagem (SALVADOR, 2016).

Figura 15: Matlab

MATLAB

Fonte: (SALVADOR, 2016)

2.1.5 Conversor USB/Serial PL2303

Para que seja possível a comunicação entre o microcontrolador, foi utilizado o conversor USB/Serial PL2303, ilustrado na Figura 16. Este possui uma comunicação via USB, tendo um circuito que faz a conversão para UART, protocolo muito utilizado em microcontroladores (Prolific, 2019).

Figura 16: Conversor USB/Serial PL2303

Fonte: (Prolific, 2019)

2.1.6 Biblioteca de Transformada Rápida de Fourier em C

Para realizar a transformada rápida de Fourier em um microcontrolador de forma rápida e eficiente, foi necessária a busca de uma biblioteca que fizesse esse cálculo seguindo esses padrões.

A biblioteca escolhida foi a KISS FTT, uma vez que ela é eficiente, moderadamente útil que pode utilizar dados do tipo *float* e pode ser incorporada em programas em C em alguns minutos (MBORGERGING, 2019).

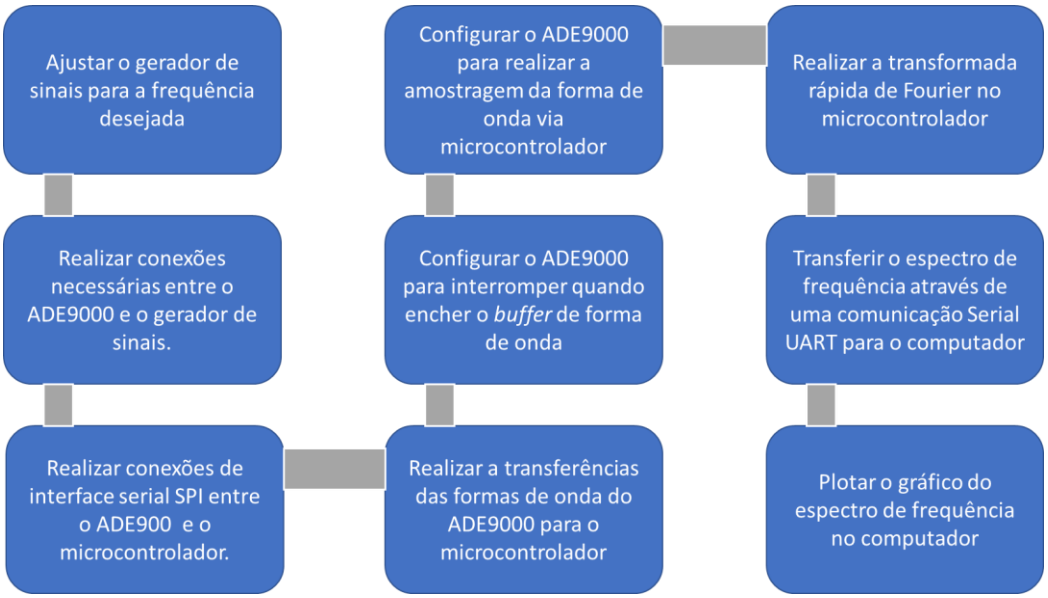
Em um processador Athlon XP 2100+ com 1 GHz de frequência de *clock*, a biblioteca teve um desempenho de 10 mil ffts de 1024 pontos em 0.63 segundos, transformando 5 minutos de áudio da qualidade de CD em menos de 1 segundo.

2.2 FLUXOGRAMA DO FUNCIONAMENTO DO PROJETO

O fluxograma da Figura 17 demonstra o funcionamento do sistema como um todo. Esse processo inicia-se com o ajuste do gerador sinais, escolhendo frequências para que sejam analisadas. Feito isso, é necessário realizar as conexões para que o ADE9000 possa medir a forma de onda provida do gerador de sinais, juntamente com as conexões necessárias para a comunicação entre o ADE9000 e o microcontrolador, que se conectam por meio de uma comunicação serial SPI.

Após isso, o ADE9000 precisa ser configurado para que ele faça a amostragem da forma de onda e também para que ele gere uma interrupção no caso de um *buffer* de forma de onda cheio. Uma vez configurado, a cada interrupção, o microcontrolador realizará a transferência dos dados armazenados no ADE9000 para ele. Quando os dados forem suficientes para atender a norma, será feita a transformada rápida de Fourier dentro do microcontrolador e, no final de cada transformada, será transferido o espectro de frequência para o computador via comunicação UART Serial. Por fim, o computador através do Matlab, receberá os dados da forma de onda e espectro de frequência e o plotará o resultado feito pelo microcontrolador, e o resultado utilizando o próprio Matlab.

Figura 17: Fluxograma do funcionamento do projeto



Fonte: (Elaborado pelo autor, 2019)

3 IMPLEMENTAÇÃO DO PROJETO

O capítulo de implementação do projeto está dividido basicamente em sete etapas:

a) diagrama em blocos do hardware;
b) estudo sobre a norma IEC 61000-4-7;
c) fluxograma do algoritmo desenvolvido;
d) ajuste do gerador de sinais;
e) conexões e comunicação do ADE9000;
f) transformada rápida de Fourier no microcontrolador;
g) transferência de dados para o computador.

3.1 DIAGRAMA EM BLOCOS DO HARDWARE

O diagrama em blocos da Figura 18 demonstra a ligação física entre os componentes utilizados neste projeto. Primeiramente, a saída do gerador de sinais, onde a forma de onda se encontra, é ligada na entrada do ADE9000. Após isso, o ADE9000 está conectado por meio da comunicação SPI com o microcontrolador STM32F4, sendo que este contém uma saída que serial para enviar o resultado final, precisando ser conectado ao conversor USB-serial e, por fim, conectando ao computador através de uma entrada USB.

Figura 18: Diagrama em blocos do hardware

Fonte: (Elaborado pelo autor, 2019)

3.2 ESTUDO SOBRE A NORMA IEC 61000-4-7

Para atender a norma IEC 61000-4-7, foi necessário que tenhamos 200 ms de dados para que possa ser feita a transformada rápida de Fourier. Pensando nisso, uma vez que o ADE9000 será configurado para uma taxa de 8 ksps, o cálculo foi feito

para determinar o número de amostras necessária para a representação no domínio da frequência.

$$Quantidade\ de\ amostras = Frequência\ de\ Amostragem * Tempo \quad (8)$$

$$Quantidade\ de\ amostras = 8000 * 0.2$$

$$Quantidade\ de\ amostras = 1600\ amostras$$

Esta quantidade de 1600 amostras, equivale a 12 ciclos de uma forma de onda de 60 Hz, amostradas a uma frequência de 8 kHz.

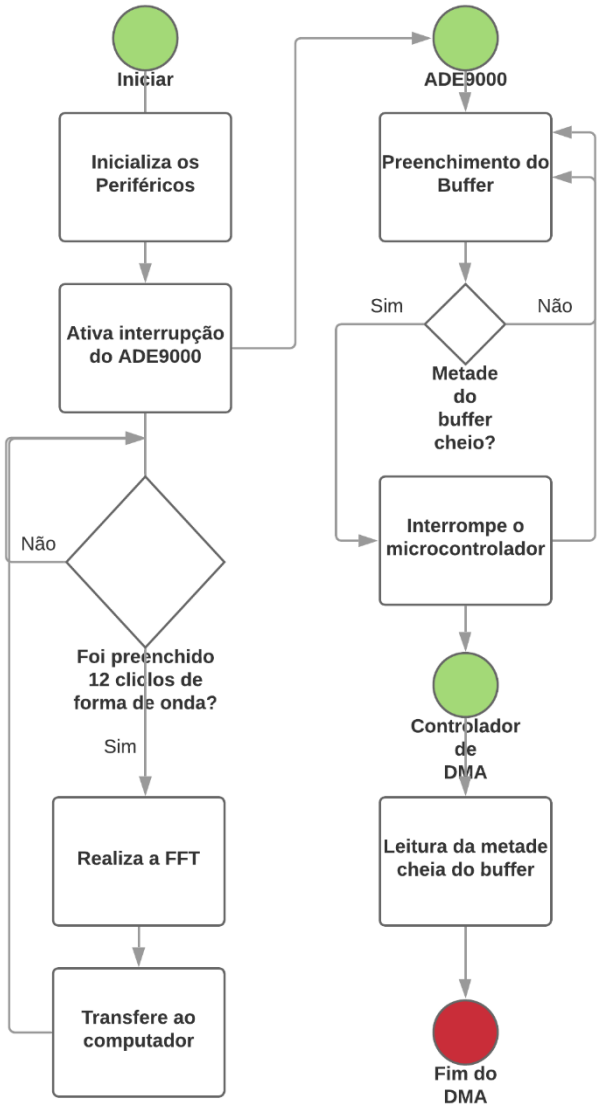
3.3 FLUXOGRAMA DO ALGORITMO DESENVOLVIDO

O algoritmo, ilustrado na Figura 19, se divide em 3 diferentes processos: o principal, ADE9000 e o controlador de DMA. O programa principal, primeiramente será responsável pela inicialização dos periféricos. Após isso, ele inicializará o ADE9000 para que faça uma interrupção quando a página final e a página da metade forem preenchidas, resultando sempre em uma metade de buffer cheia.

Com isso, o ADE9000 será responsável pela a amostragem das formas de onda e, assim que as páginas configuradas estiverem cheias, o ADE9000 interromperá o microcontrolador de forma assíncrona, informando qual parte do buffer está cheia, iniciando o terceiro processo.

No controlador de DMA, será feita a transferência dos dados do buffer do ADE9000 para o microcontrolador, independentemente do processamento do programa principal, preenchendo um buffer interno à memória do microcontrolador. Quando atingir o equivalente a 12 ciclos de forma de onda, o microcontrolador realizará a transformada rápida de Fourier, entregando todos esses dados ao computador.

Figura 19: Fluxograma do Algoritmo

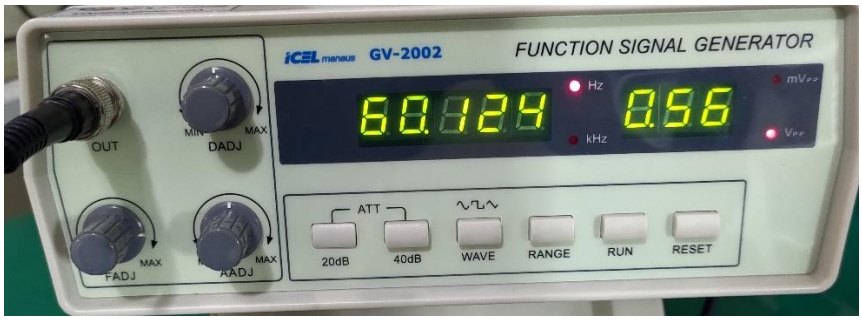


Fonte: (Elaborado pelo autor, 2019)

3.4 AJUSTE DO GERADOR DE SINAIS

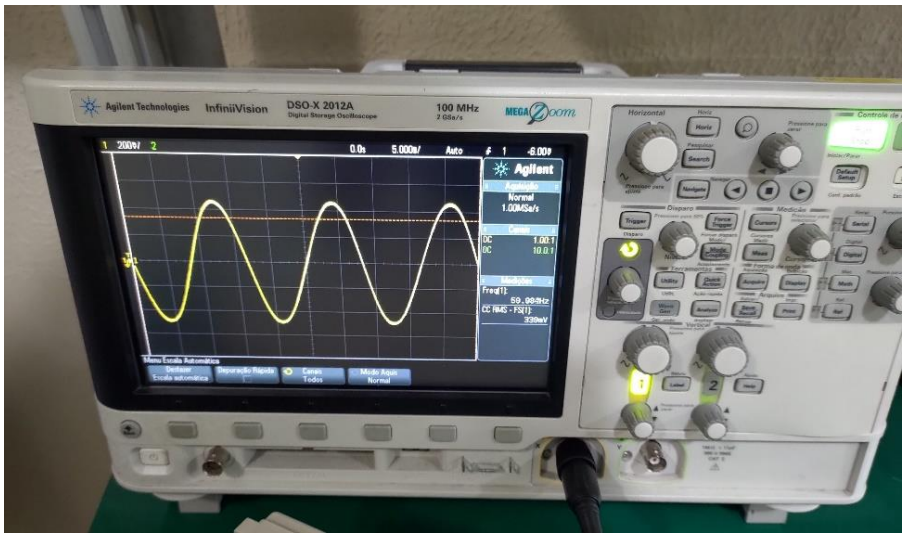
Utilizando o visor e os botões que se encontram no gerador de sinais, primeiramente foi feito o ajuste do gerador para que ele trabalhasse com uma tensão senoidal de 60 Hz de acordo com a Figura 20. Entretanto, para que seja realmente validade a forma de onda de entrada, foi utilizado o osciloscópio digital em conjunto, conforme a Figura 21.

Figura 20: Ajuste do gerador de sinais



Fonte: (Elaborado pelo autor, 2019)

Figura 21: Forma de onda no osciloscópio



Fonte: (Elaborado pelo auto, 2019)

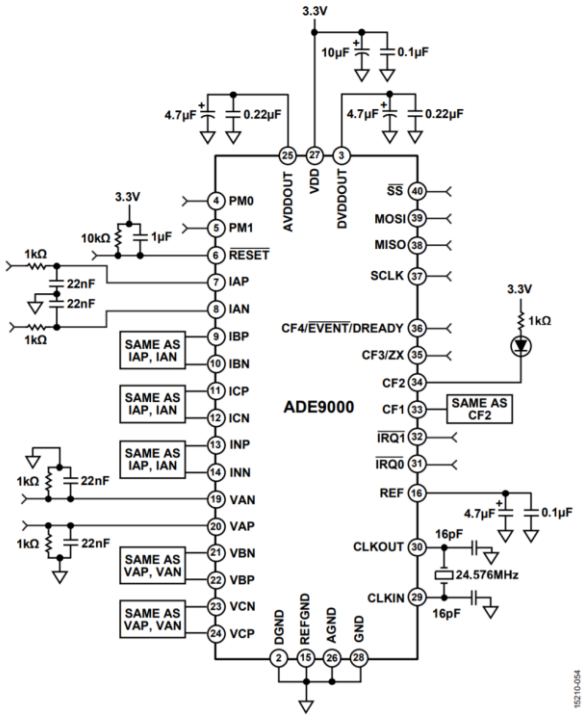
Para a validação dos resultados, outras formas de onda e outras frequências foram utilizadas, sempre utilizando o procedimento descrito acima.

3.5 CONEXÕES DO ADE9000

3.5.1 Circuito de Teste do ADE9000

Inicialmente, para que os sinais sejam normalmente condicionados, foi necessário a utilização do circuito de teste demonstrado no *datasheet* do ADE9000, conforme Figura 22. Uma vez se trata de um sinal a ser amostrado, é necessário a utilização de alguns filtros, além da utilização de *clock* e tensão de referência.

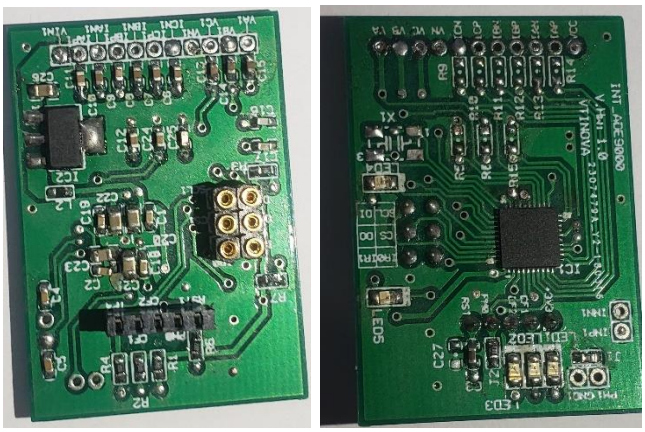
Figura 22: Circuito de teste do ADE9000



Fonte: (ANALOG-DEVICES, 2017)

A interface do ADE9000, que tem como base o circuito de teste apresentado, foi desenvolvida e fabricada, onde o esquema elétrico se encontra no APÊNDICE A – INTERFACE ADE9000. A placa de circuito impressa e montada se encontra na Figura 23. Nesta placa, é possível notar que todas as entradas de sinais estão externadas e de fácil acesso, juntamente com a interface de comunicação serial SPI e pinos de interrupção externos.

Figura 23: Placa de circuito impresso da interface ADE9000

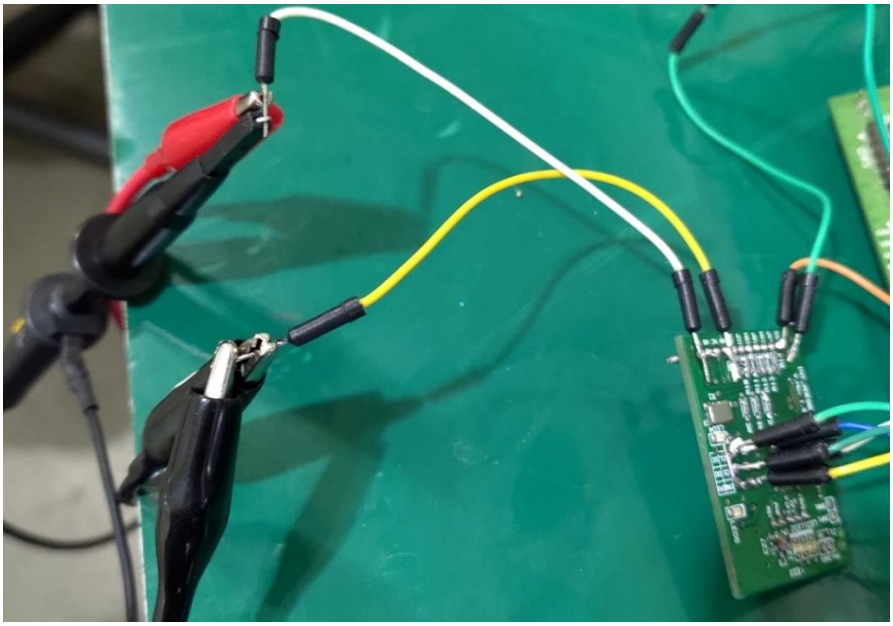


Fonte: (Elaborado pelo autor, 2019)

3.5.2 Ligação Elétrica do ADE9000

Para fins de teste, apenas uma das entradas foi utilizada, uma vez que o processo é replicado para as demais entradas. A entrada escolhida foi a tensão VA. Primeiramente, foi feito a ligação entre o gerador de sinais com a entrada VA e VN, representando uma fase um neutro, conforme Figura 24.

Figura 24: Ligação entre ADE9000 e gerador de sinais



Fonte: (Elaborado pelo autor, 2019)

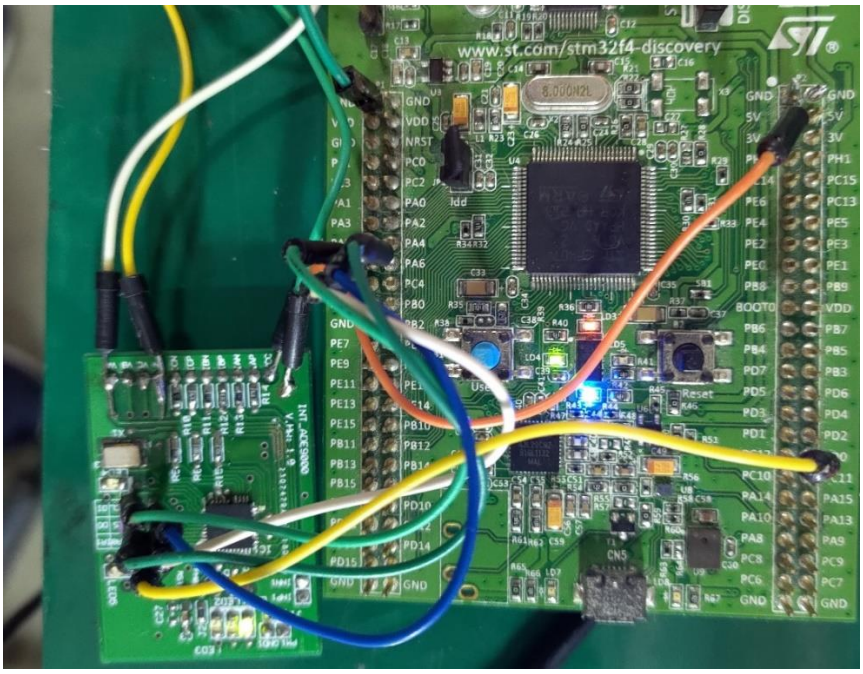
No quesito de microcontrolador, foi utilizado um STM32F4-Discovery, que possui um *clock* de 168Mhz, entretanto, para simular um dispositivo com capacidade menor, foi reduzido o *clock* por 3, resultando em 56Mhz. A ligação entre o microcontrolador e o ADE9000 se dá por meio de 5 fios, conforme Tabela 2 e demonstrado ilustrativamente na Figura 25.

Tabela 2: Pinos de ligação entre microcontrolador e ADE9000

Pino do Microcontrolador	Pino do ADE9000	Significado
PA7	MOSI	Master Output Slave Input
PA6	MISO	Master Input Slave Output
PA5	SCLK	Serial Clock
PC5	SS	Slave Select
PD0	IRQ0	Interrupção

Fonte: (Elaborado pelo autor, 2019)

Figura 25: Ligação entre microcontrolador e ADE9000



Fonte: (Elaborado pelo autor, 2019)

3.6 COMUNICAÇÃO COM O ADE900

3.6.1 Inicialização da Comunicação SPI

Uma vez definidos os pinos de ligação entre o microcontrolador e o ADE9000, iniciou-se a inicialização dos periféricos a serem utilizados para a comunicação ocorrer. Para isso, foi criada uma biblioteca em C responsável por toda comunicação SPI, denominada *spi.c*.

Nesta biblioteca, primeiramente foi criada uma função que faz a inicialização das GPIO's que serão utilizadas.

```
/* Incialização dos GPIOS SPI */
gpio_init(GPIOA, GPIO_Pin_5 | GPIO_Pin_6 | GPIO_Pin_7, GPIO_OType_PP,
GPIO_PuPd_NOPULL, GPIO_High_Speed, GPIO_AF_SPI1);

/*Incialização do GPIO Interrupção*/
irq0_gpio_init();

/*Incilização do Chip Select*/
GPIOx_CS = GPIOC;
GPIO_Pin_CS = GPIO_Pin_5;
chip_select_init();
```

Com os pinos de GPIO já inicializados, na mesma função, foi inicializado o periférico de comunicação SPI propriamente dito, definindo os parâmetros como descritos no *datasheet* do ADE9000.

```
/* Inicialização da SPI*/
RCC_APB2PeriphClockCmd( RCC_APB2Periph_SPI1, ENABLE);
SPI_InitTypeDef SPI_InitStruct;
SPI_StructInit(&SPI_InitStruct);
SPI_InitStruct.SPI_DataSize = SPI_DataSize_16b;
SPI_InitStruct.SPI_BaudRatePrescaler = SPI_BaudRatePrescaler_2;
SPI_InitStruct.SPI_Direction = SPI_Direction_2Lines_FullDuplex;
SPI_InitStruct.SPI_FirstBit = SPI_FirstBit_MSB;
SPI_InitStruct.SPI_Mode = SPI_Mode_Master;
SPI_InitStruct.SPI_NSS = SPI_NSS_Soft;
SPI_InitStruct.SPI_CPOL = SPI_CPOL_High;
SPI_InitStruct.SPI_CPHA = SPI_CPHA_2Edge;
SPI1->CR1 &= ~SPI_CR1_SPE;
SPI_Init(SPI1, &SPI_InitStruct);
SPI1->CR1 |= SPI_CR1_SPE;
```

O tamanho do dado foi definido como 16 bits, ou seja, 2 bytes. Uma vez que o *clock* do microcontrolador está configurado como 56 MHz, o *clock* a ser empregado no periférico é no mínimo 2 vezes menor, resultando em 26 MHz. Entretanto, a frequência máxima suportada pelo ADE9000 é de 20 Mhz e, por esta razão foi definido um *Prescaler* igual a 2, que divide a frequência na metade, ou seja, 13 MHz.

Além disso, a comunicação acontece nas duas direções e simultaneamente, ou seja, *Full Duplex*. Por fim, temos a configuração de modo *Master*, com a segunda borda em nível alto.

3.6.2 Escrita e Leitura de Registradores do ADE9000

Como descrito no tópico 1.3.2 Protocolo SPI para que a comunicação ocorra, é necessário que o pino de *chip select* seja ativado a cada operação de escrita e leitura, e desativado assim que esta operação termine. Com base nisso foram criadas funções que ativam e desativam o *chip select*.

```
void chip_select()
{
	GPIO_WriteBit(GPIOx_CS, GPIO_Pin_CS, 0);
}

void chip_deselect()
```

```
{
	GPIO_WriteBit(GPIOx_CS, GPIO_Pin_CS, 1);
}
```

3.6.2.1 Escrita de Registradores

O ADE9000 possui diversos registradores no qual podem ser configurados. Para escrever dados nestes registradores, é necessário primeiramente escrever o endereço do registrador a ser configurado, e em seguida, o dado propriamente dito.

```
void set_registrador(uint16_t endereco_registrador, uint32_t
tamanho_dado, uint16_t* dado)

{
	chip_select();
	endereco_registrador = ((endereco_registrador << 4) & 0xFFF0);
	SPI_CHECK_ENABLED(SPI1);
	SPI_WAIT(SPI1);
	SPI1->DR = endereco_registrador; //Escrita
	SPI_WAIT(SPI1);
	(void)SPI1->DR; //Leitura

	for (uint32_t i = 0; i < tamanho_dado; i++) {
		SPI1->DR = dado[i]; //Escrita
		SPI_WAIT(SPI1);
		(void)SPI1->DR; //Leitura
	}
	chip_deselect();

}
```

Nota-se que primeiramente é utilizada a função *chip_select()*, que é necessária para a comunicação, após isso é feita uma operação com o endereço do registrador que habilita o modo de escrita. Após isso, é feita a escrita do endereço do registrador, juntamente com a leitura de um desprezível, uma vez que a comunicação é sempre *Full Duplex*. Em sequência, este processo é repetido para cada posição do vetor dado. Por fim, é utilizada a função *chip_deselect()* para a finalização de uma comunicação.

3.6.2.2 Leitura de Registradores

De forma muito semelhante a escrita de registradores, os registradores também podem ser lidos através da comunicação SPI. Uma diferença é que no endereço do registrador é habilitado o modo de leitura e a outra, é que ao invés da leitura de valores

não importantes, é feita a escrita de valores quaisquer, no caso repetindo o endereço do registrador.

```
void get_registrador(uint16_t endereco_registrador, uint32_t
tamanho_dado, uint16_t* dado)
{
	chip_select();
	endereco_registrador = (((endereco_registrador << 4) &
0xFFF0)+8);
	SPI_CHECK_ENABLED(SPI1);
	SPI_WAIT(SPI1);
	SPI1->DR = endereco_registrador;
	SPI_WAIT(SPI1);
	(void)SPI1->DR;
	for (uint32_t i = 0; i < tamanho_dado; i++) {
		SPI1->DR = endereco_registrador;
		SPI_WAIT(SPI1);
		dado[i] = SPI1->DR;
	}
	chip_deselect();
}
```

3.6.3 Configuração do ADE9000

Uma vez que é possível ler e escrever nos registradores, é possível também configurar o ADE9000 de forma que ele amostre os sinais de entrada e interrompa quando desejado. Para isso diversos registradores foram configurados, sempre utilizando a função *set_registradores()* seguida da função *get_registradores()* para garantir que o valor está correto, conforme o exemplo a seguir.

```
set_registrador(MASK0, TAM_MASK0, DEFAULT_MASK0);
get_registrador(MASK0, TAM_MASK0, temp);
for (int i = 0; i < TAM_MASK0; i++)
{
	if(temp[i] != DEFAULT_MASK0[i])
		return 0;
}
return 1;
```

O primeiro registrador configurado foi o WFB_CFG, responsável pela configuração da forma de onda. Nele, foi configurado a saída sinc4 com uma taxa de atualização de 8ksps no modo de sempre continuar preenchendo e a leitura dos *buffers* de forma de onda de todos os canais.

Como o ADE9000 está configurado para continuar preenchendo o *buffer* de forma de onda, quando encher, automaticamente retornará ao início e começará a

substituir os dados dentro do *buffer*. Para isso, foi necessário definir dois pontos de interrupção, um quando o *buffer* atinge a metade da sua capacidade e outra quando atinge o máximo da sua capacidade, sendo assim necessário ser feita a leitura de uma metade enquanto o ADE9000 preenche a outra.

Por conseguinte, o registrador WFB_PG_IRQEN foi configurado com dois valores em nível alto, um representando a página 0 e o outro representando a página 8.

Por fim, o registrador MASK0, responsável pela interrupção no pino IRQ0, foi habilitado para interromper apenas quando uma página ativada no registrador WFB_PG_IRQEN está cheia.

3.6.4 Transferências das Formas de Onda para o Microcontrolador

Visto que o ADE9000 está devidamente configurado e trabalhará de forma independente, sendo informado ao microcontrolador quando uma página é preenchida, é possível fazer a transferência desses dados de forma de onda.

Quando acontece uma interrupção no IRQ0, o primeiro passo é descobrir qual foi a última página preenchida, para que seja possível saber qual parte do *buffer* será transferida. Para isso, foi criada a função *get_last_page()* que se encontra a seguir. Esta função faz a leitura do registrador WFB_TRG_STAT que informa a última página preenchida, sendo necessária a conversão para valores inteiros.

```
int get_last_page ( void )
{
	uint16_t temp[TAM_WFB_TRG_STAT];
	get_registrador(WFB_TRG_STAT, TAM_WFB_TRG_STAT, temp);
	int page = (temp[0] & 0xF000) >> 12;
	if (page < 8) return 0;
	else return 1;
}
```

Com esta informação, o controlador de DMA pode ser iniciado com o endereço do *buffer* desejado. Para iniciar uma operação de escrita e leitura, novamente é necessário a ativação do *chip select*. Além disso, conforme o *datasheet* do ADE9000, temos que a metade do *buffer* teria 2048 posições de 16 bits, sendo 2 adicionais para o comando de escrita, resultando em 2050.

```
void dma_init(uint32_t* pTmpBuf1 , int page)
{
	chip_select();
	if (page == 0)
		pTmpBuf1[0] = 0x8008;
	else
		pTmpBuf1[0] = 0xC008;

	DMA_InitTypeDef DMA_InitStructure;

	DMA_StructInit(&DMA_InitStructure);
	RCC_AHB1PeriphClockCmd( RCC_AHB1Periph_DMA2, ENABLE);

	DMA_DeInit(DMA2_Stream3);
	DMA_DeInit(DMA2_Stream2);

	DMA_InitStructure.DMA_BufferSize = (uint16_t)2050;
```

A partir disso, o controlador de DMA trabalhará de forma independente, interrompendo o microcontrolador quando esses dados já estivem sidos transferidos. Quando esses dados estão no microcontrolador, é necessário que eles sejam preparados para a transformada de Fourier, pois eles chegam no formato sinc4. Para a decodificação desses dados foi implementada a função *sinc4_decoder*().

```
kiss_fft_scalar sinc4_decoder(uint32_t burst)
{
	kiss_fft_scalar temp = 0.0;
	int signal = (burst & 0x80000000) >> 31;
	int32_t man = (burst >> 4);
	if (signal == 1)
	{
		man |=  0xFF000000;
	}
	temp = ((float) man);
	return temp;
}
```

Como o *buffer* de forma de onda é preenchido de acordo com os canais de forma sequencial, foi necessário varrer o vetor que fez a leitura através do DMA, decodificando os canais na respectiva sequência.

```
void burst_to_buffer( controller* ct)
{
	for(int i = 0; i < 897; i+= 7)
	{
		ct->buffer_read.IA[ct->count] = sinc4_decoder(ct-
>burst_read[i]);
		ct->buffer_read.VA[ct->count] = sinc4_decoder(ct-
>burst_read[i+1]);
```

```
            ct->buffer_read.IB[ct->count] = sinc4_decoder(ct-
>burst_read[i+2]);
            ct->buffer_read.VB[ct->count] = sinc4_decoder(ct-
>burst_read[i+3]);
            ct->buffer_read.IC[ct->count] = sinc4_decoder(ct-
>burst_read[i+4]);
            ct->buffer_read.VC[ct->count] = sinc4_decoder(ct-
>burst_read[i+5]);

            ct->count++;
            if(ct->count == 1600)
            {
                  ct->buffer = 0;
                  ct->full_buffer = 1;
            }
            else if (ct->count >= 3200)
            {
                  ct->buffer = 1;
                  ct->full_buffer = 1;
                  ct->count = 0;
            }
      }
}
```

O *buffer* interno do microcontrolador foi composto por 3200 amostras, uma vez que enquanto 1600 amostras estão sendo preenchidas, as outras 1600 vão ser utilizadas na transformada rápida de Fourier.

3.7 TRANSFORMADA RÁPIDA DE FOURIER NO MICROCONTROLADOR

Para o processo de transformada rápida de Fourier no microcontrolador, foi criada uma biblioteca denominada *fft.h*. Primeiramente, foi feita a importação da biblioteca de transformada rápida de Fourier denominada KISS_FFT.

.

```
#include "kiss_fft.h"
#include "kiss_fftr.h"
```

Após isso, foi necessário alocar espaços de memória que a biblioteca necessita para fazer o cálculo, que depende exclusivamente do número de amostras utilizadas na transformada.

```
void init_fft ( controller* ct )
{
      cfg = kiss_fftr_alloc( WFB_ELEMENT_ARRAY_SIZE/2 ,0 ,NULL,NULL );
}
```

Por fim, foi utilizada a função *kiss_fftr()* que tem como parâmetros a configuração alocada anteriormente, o endereço dos dados no domínio do tempo, e o endereço dos dados a serem calculados no domínio da frequência. Lembrando sempre de que quando um *buffer* interno ao microcontrolador é preenchido, o outro que será utilizado na transformada de Fourier.

```
void burst_to_buffer( controller* ct)
{
for(int i = 0; i < 897; i+= 7)
{
ct->buffer_read.IA[ct->count] = sinc4_decoder(ct->burst_read[i]);
ct->buffer_read.VA[ct->count] = sinc4_decoder(ct->burst_read[i+1]);
ct->buffer_read.IB[ct->count] = sinc4_decoder(ct->burst_read[i+2]);
ct->buffer_read.VB[ct->count] = sinc4_decoder(ct->burst_read[i+3]);
ct->buffer_read.IC[ct->count] = sinc4_decoder(ct->burst_read[i+4]);
ct->buffer_read.VC[ct->count] = sinc4_decoder(ct->burst_read[i+5]);
ct->count++;
if(ct->count == 1600)
{
	ct->buffer = 0;
	ct->full_buffer = 1;
}



else if (ct->count >= 3200)
{
	ct->buffer = 1;
	ct->full_buffer = 1;
	ct->count = 0;
		}
	}
}
```

3.8 TRANFERÊNCIA DOS DADOS PARA O COMPUTADOR

3.8.1 Ligação Elétrica do Conversor USB/Serial

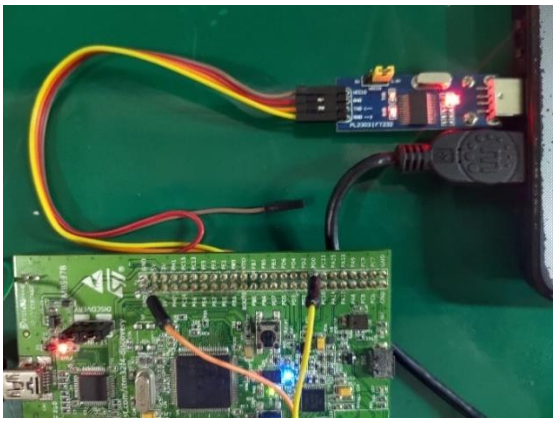
A ligação entre o microcontrolador e o Conversor USB/Serial se dá por meio de 2 fios, conforme Tabela 3 e demonstrado ilustrativamente na Figura 26.

Tabela 3: Pinos de ligação entre microcontrolador e conversor

Pino do Microcontrolador	Pino do Conversor	Significado
PB6	RX	Microcontrolador para Computador
PB7	TX	Computador para microcontrolador

Fonte: (Elaborado pelo autor, 2019)

Figura 26: Ligação entre microcontrolador e conversor



Fonte: (Elaborado pelo autor, 2019)

3.8.2 Envio de Dados Pelo Microcontrolador

Uma vez definidos os pinos que serão utilizados pelo microcontrolador para a comunicação serial, foi utilizada uma biblioteca do STM32F4 que trata justamente de fazer o envio desses dados. Além disso, a transferência de dados também é de forma semelhante à transferência dos dados do ADE9000, uma que vez que será utilizado novamente um controlador de DMA, para que os processos do microcontrolador não sejam comprometidos.

```
#include "tm_stm32f4_usart.h"
#include "tm_stm32f4_usart_dma.h"
```

Após a biblioteca ser importada, foram configurados os pinos utilizados, juntamente com a frequência de *baudrate* utilizada, para ser feito o casamento dessas configurações no computador.

```
        TM_USART_Init(USART1, TM_USART_PinsPack_2, 345600);
        TM_USART_DMA_Init(USART1);
```

Por fim, como apenas o canal de tensão VA está sendo injetado por uma forma de onda, os dados da forma de onda, juntamente com o espectro calculado dentro do microcontrolador serão enviados via UART.

```
array = (uint8_t*)(&ct->espectro.VA);
TM_USART_DMA_Send(USART1, array, 801*4*2);
while((TM_USART_DMA_Sending(USART1)));
array = (uint8_t*)(&ct->buffer_read.VA);
TM_USART_DMA_Send(USART1, &array[(ct->buffer+1)*1600], 800*4*2);
```

3.8.3 Recebimento dos Dados pelo Computador

Utilizando a ferramenta Matlab foi criado um script para que seja possível receber dados vindos de uma porta USB do computador. Para isso, é necessário fazer a configuração desta porta, juntamente com a frequência configurada pelo microcontrolador.

```
s = serial('COM7', 'BaudRate', 115200);
fopen(s);
```

Como primeiramente é enviado o espectro calculado pelo microcontrolador, foi utilizada a função *fread()* para receber os dados do eixo real e do eixo imaginário, conforme a quantidade de raias no domínio da frequência.

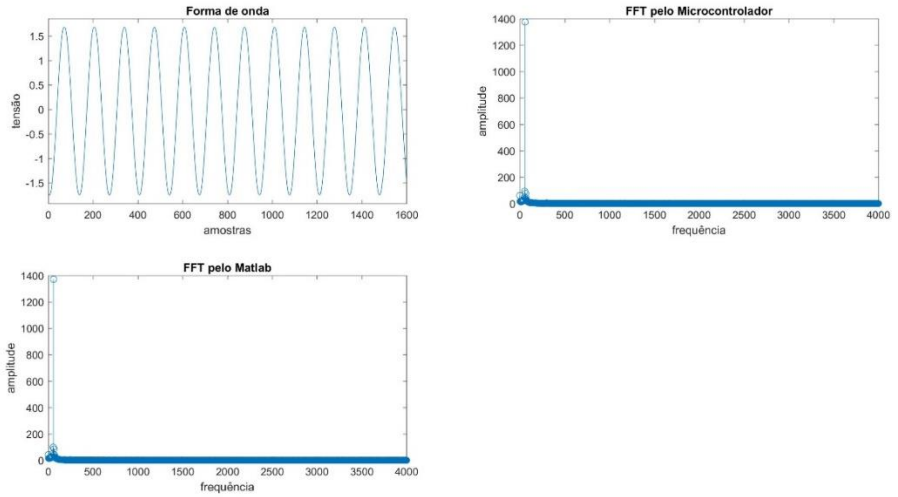
```
for j=1:qnt
espectro_r(j) = fread(s,1,'float');
espectro_i(j) = fread(s,1,'float');
end
```

Após isso, o mesmo método foi empregado para os dados da forma de onda, sendo possível efetuar o mesmo cálculo de transformada rápida de Fourier no computador já validado pelo Matlab.

```
for j=1:qnt_s
wf(j) = fread(s,1,'float');
end
espectro_matlab = fft(wf, 1600, 1);
```

Por fim, o gráfico da forma de onda da entrada, o espectro de frequências calculado pelo microcontrolador e o espectro de frequência calculado pelo Matlab foram plotados para a visualização dos dados, conforme o exemplo na Figura 27.

Figura 27: Exemplo de Resultado



Fonte: (Elaborado pelo autor, 2019)

Por fim, o índice de distorção harmônica total foi calculador para ambos espectros, com o intuito de comparar o cálculo feito pelo microcontrolador e feito pelo Matlab.

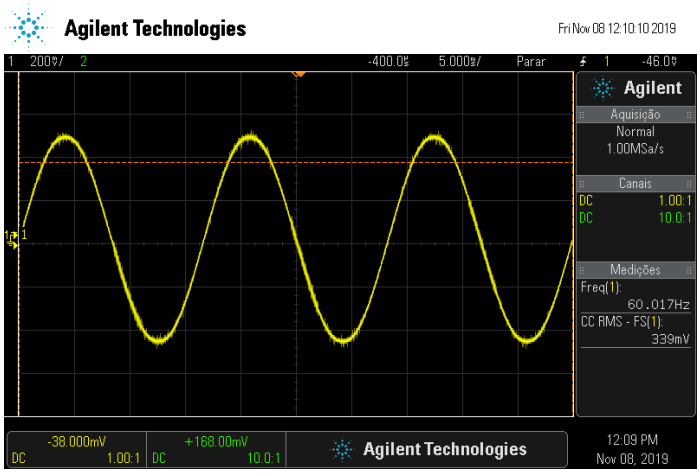
$$DHT\ (Microcontrolador) = 0{,}024$$

$$DHT\ (Matlab) = 0{,}022$$

4 RESULTADOS

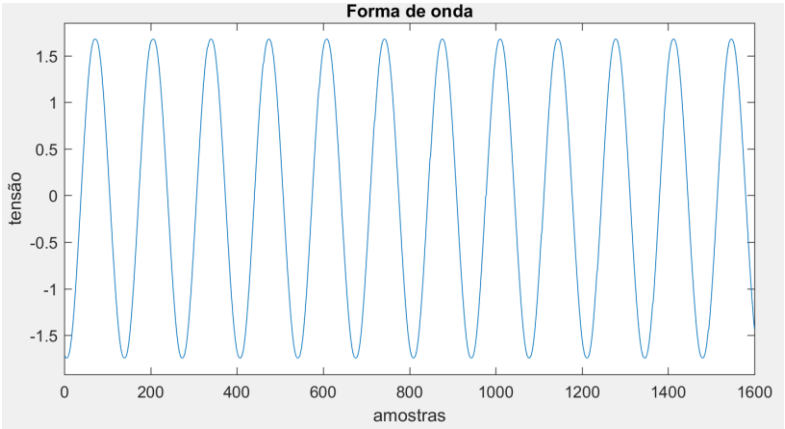
Realizada a implementação do algoritmo proposto, diferentes casos foram testados para a validação dos resultados. O primeiro deles é justamente o caso de uma forma de onda senoidal com frequência de 60Hz, pois é o caso de como as redes elétricas no Brasil se comportam, ilustradas como forma de onda configurada no gerador de sinais e forma de onda amostrada pelo ADE9000, nas Figura 28 e Figura 29 respectivamente.

Figura 28: Forma de onda de 60 Hz senoidal no gerador



Fonte: (Elaborado pelo autor, 2019)

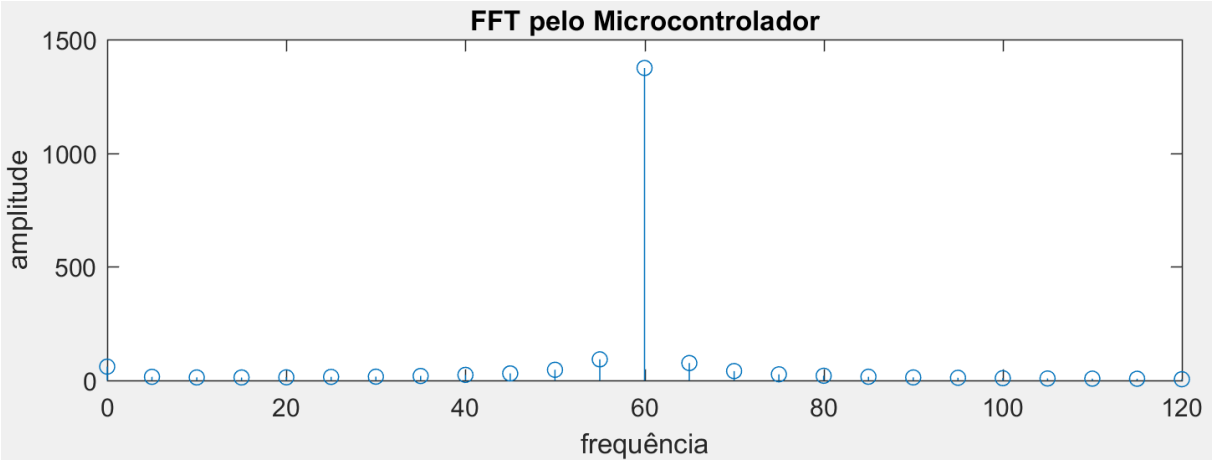
Figura 29: Forma de onda de 60 Hz senoidal amostrada pelo ADE9000



Fonte: (Elaborado pelo autor, 2019)

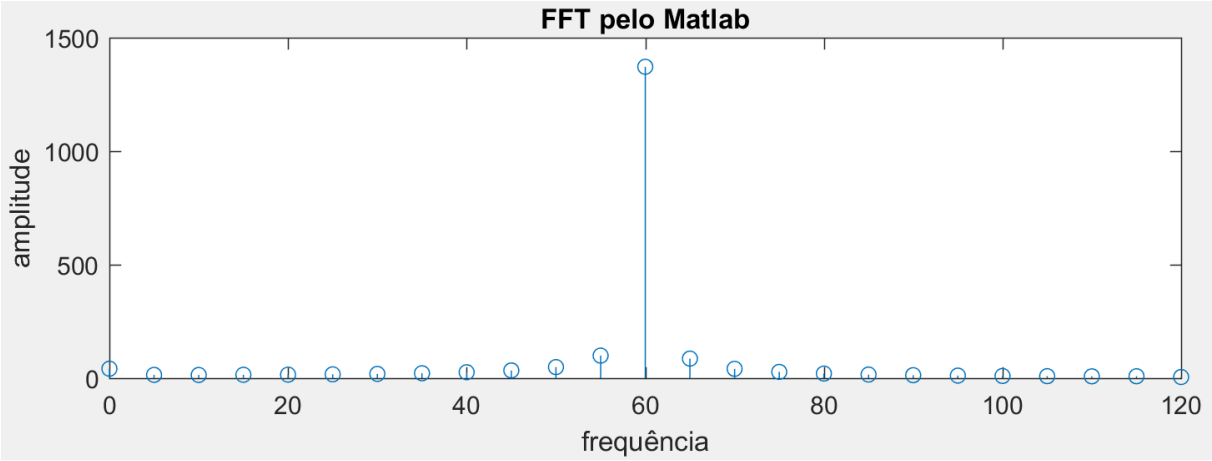
Para fins de avaliação da frequência de 60 Hz, foram ampliados os gráficos de espectro de frequência para um intervalo que ficasse melhor a visualização dos resultados. Os gráficos de espectro de frequência apresentadas pelo microcontrolador e pelo Matlab, se encontram nas Figuras 30 e 31, respectivamente.

Figura 30: Espectro da forma de onda de 60 Hz senoidal pelo microcontrolador



Fonte: (Elaborado pelo autor, 2019)

Figura 31: Espectro da forma de onda de 60 Hz senoidal pelo Matlab



Fonte: (Elaborado pelo autor, 2019)

Apesar da frequência configurada ser 60 Hz, é possível notar algumas raias ao entorno desta frequência também possuem amplitude. Isso se dá por justamente o gerador de sinais ter algum pequeno desvio de frequência, o que acontece nas redes elétricas. Para isso, existe o grupo de harmônicas que pode ser calculado de acordo com a norma IEC 61000-4-7.

Outras formas de onda, seguindo a metodologia apresentada, foram testadas e se encontram no APÊNDICE D – RESULTADOS OBTIDOS PARA OUTRAS FORMAS DE ONDA. Com base nas ondas que não são senoidais, elas podem ser representadas como um somatório de ondas senoidais harmônicas no qual o sistema

conseguirá detectará, dando margem a criação de filtros para melhorar a e eficiência da rede.

Para todas as formas de onda, os dados que se referem ao índice de distorção harmônica total calculados pelo microcontrolador e pelo Matlab se encontram na Tabela 4.

Tabela 4: Índice de distorção harmônica total

Forma de Onda	DHT (microcontrolador)	DHT (Matlab)
Senoidal de 60 Hz	0,024	0,022
Triangular de 60 Hz	0,1207	0,1193
Quadrada de 60 Hz	142,8282	138,8886

Fonte: (Elaborado pelo autor, 2019)

Com isso, é possível perceber que quanto menor a semelhança da forma de onda com a onda senoidal pura de 60 Hz, maior é o índice de distorção harmônica total, uma vez que apresenta mais componentes harmônicos com maior amplitude em relação a componente fundamental.

CONCLUSÃO

Este trabalho teve o intuito de desenvolver um sistema eletrônico para a medição de harmônicas de acordo com a norma IEC 61000-4-7, utilizando a amostragem de um circuito integrado medidor trifásico de energia elétrica de alta precisão, o ADE9000 e a transformada rápida de Fourier sendo calculado em um microcontrolador.

Foi apresentando um breve referencial teórico acerta dos conceitos fundamentais para realização dessa pesquisa, o qual abrangeu a análise de Fourier, o estudo da norma IEC 61000-4-7, os métodos utilizados em sistemas microcontrolados e o chip de medição de energia utilizado para capturar as formas de onda da rede, o ADE9000, da Analog Devices.

O método proposto consistiu na captura da forma de onda de sinais providos do gerador de sinais. O próximo passo consistiu em realizar a transferência dos dados armazenados no ADE9000 para o microcontrolador. Após isso, o microcontrolador foi responsável pelo cálculo da transformada rápida de Fourier, tendo como resultado o espectro de frequência, que foi transferido ao computador através de uma comunicação serial. Por fim, foi utilizado o Matlab para a validação dos resultados, recebendo os valores do espetro de frequência e da forma de onda, plotando os respectivos gráficos.

Por conta da utilização destes grupos de harmônicas, utilizando 12 ciclos de sinal, correspondentes a uma resolução de frequência de 5 Hz, e podendo ser detectada até a 65º harmônica, o sistema é capaz de fazer a medição de harmônicas em tempo real de acordo com a norma IEC 614000-4-7.

Em consequência de estar se trabalhando com baixo nível de programação em C, o entendimento dos registradores de *buffer* do microcontrolador são complicados, juntamente com a depuração dos mesmos. Além disso, outra dificuldade encontrada foi a comunicação serial, por ser necessário casar os algoritmos do microcontrolador e do computador, uma vez que eles precisam estar sincronizados para os dados enviados e recebidos serem o mesmo.

Por fim, sugere-se a continuidade deste trabalhando, avaliando outras formas de mostrar o resultado ao usuário final, como por exemplo em um *display LCD*, criando um sistema totalmente independente.

REFERÊNCIAS BIBLIOGRÁFICAS

AFONSO, J. L., & MARTINS, J. S. (2003). *As Oportunidades das Ameaças: A Qualidade da Energia Elétrica.* Portugal: Ciclo de Seminários, Departamento de Electrónica Industrial, Universidade do Minho Campus de Azurém, Guimarães.

ANALOG-DEVICES. (2017). *ADE9000*. Fonte: High Performande, Multiphase Energy and Power Quality Monitoring IC Data Sheet: https://www.analog.com/media/en/technical-documentation/data-sheets/ade9000.pdf

FLOYD, T. (2006). *Sistemas Digitais: Fundamento e Aplicações.* Santana: Pearson Education.

IEC-61000-4-7. (2002). *Testing and measurement techniques.* General guide on harmonics and interharmonics measurements and instrumentation, for power supply systems and equipment connected thereto.

INSTRUTEMP. (Setembro de 2010). *Gerador de Função Digital ITGFB-2002.* Fonte: Manual de Instruções: http://instrutemp.provisorio.ws/2010_09/ITGFB-2002.pdf

KEYSIGHT. (2012). *DSOX2012A Osciloscópio.* Fonte: KEYSIGHT TECHNOLOGIES: https://literature.cdn.keysight.com/litweb/pdf/5990-6618PTBR.pdf?id=2002854

KINOSHITA, J., CUGNASCA, C. E., & HIRAKAWA, A. R. (2004). *INTERRUPÇÕES.*

LI, C., XU, W., & TAYJASANANT, T. (2003). *Interharmonics: basic concepts and techniques for their detection and measurement.* Elsevier Science, p. 39-48.

MBORGERGING. (2019). *KISS FFT*. Fonte: Github: https://github.com/mborgerding/kissfft

OPPENHEIM, A. V., & WILLSKY, A. S. (2010). *Sinais e Sistemas.* Cap. 3: Representação de sinais periódicos em série de Fourier. Ed 2: p 104 – 107.

PAULILO, G. (2013). *Conceitos Gerais Sobre Qualidade da Energia.* Cap. 1: Qualidade de Energia. O Setor Elétrico. Ed 84: p 28 – 35.

PHIPPS, J. K., NELSON, J. P., & SEN, P. K. (1994). *Power Quality and Harmonic Distortion on Distribution Systems.* IEEE Trans. on Industry Applications, vol. 30, no. 2, pp; 476-485.

POMILIO, J. A., & DECKMANN, S. M. (1997). *Efeito Flicker produzido pela Modulação Harmônica.* São Lourenço, MG: SBQEE'97 - Seminário Brasileiro de Qualidade da Energia Elétrica.

Prolific. (2019). *PL2303.* Fonte: Prolific: http://www.prolific.com.tw/US/ShowProduct.aspx?p_id=225&pcid=41

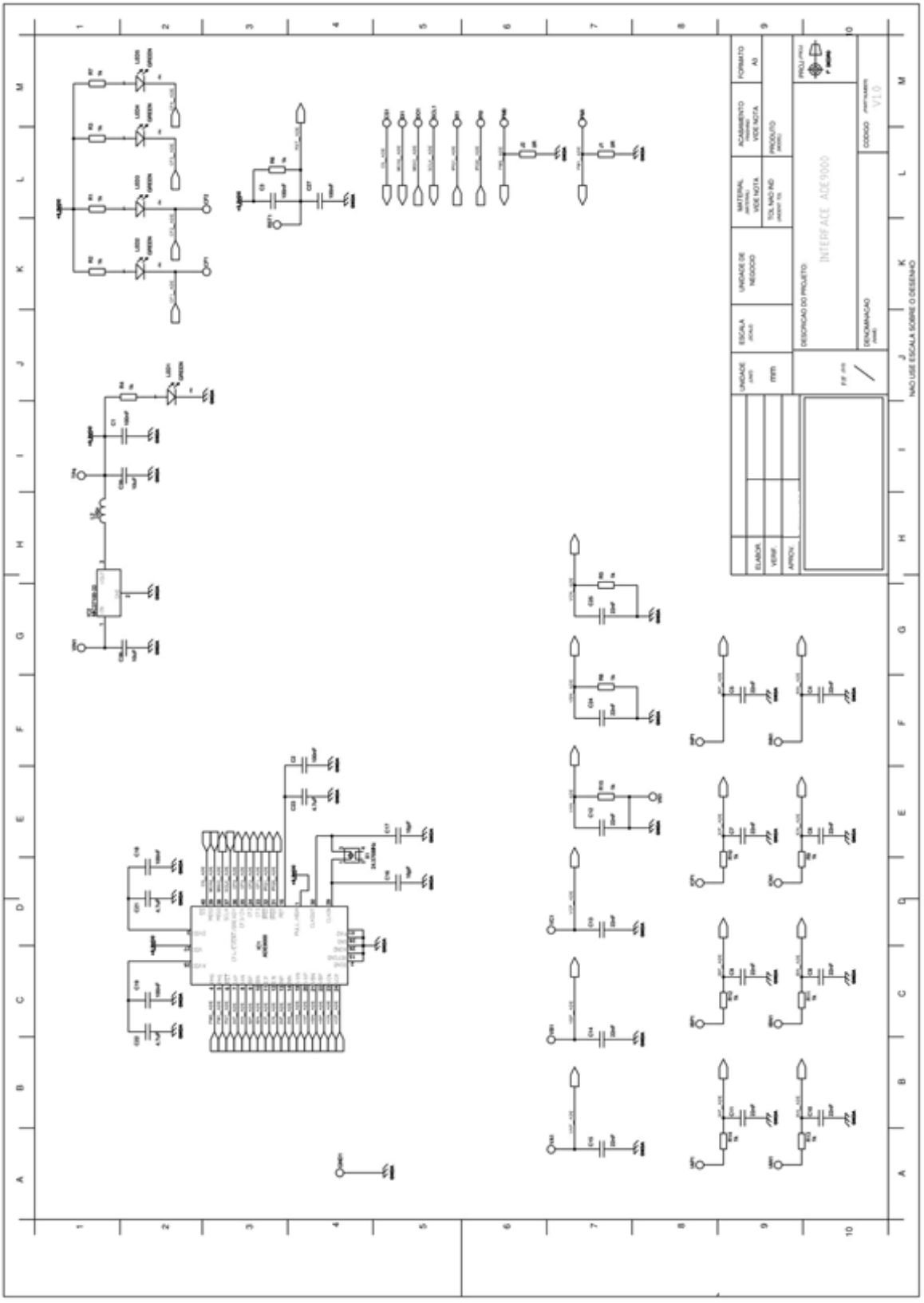
ROCHA, R. (2007). Microcontroladores: Protocolos SPI. *Programar*, 34.

SALVADOR, V. (Agosto de 2016). *O que é o Matlab?* Fonte: Portal CSTI: https://www.portalgsti.com.br/2016/08/o-que-e-o-matlab.html

ST. (2017). *Discovery kit with STM32f407VG MCU.* Fonte: ST: https://www.st.com/content/ccc/resource/technical/document/user_manual/70/fe/4a/3f/e7/e1/4f/7d/DM00039084.pdf/files/DM00039084.pdf/jcr:content/translations/en.DM00039084.pdf

APÊNDICE A – INTERFACE ADE9000

Figura 32: Interface ADE9000



Fonte: (Elaborado pelo autor, 2019)

APÊNDICE B – ALGORITIMO PRINCIPAL DO MICROCONTROLADOR

```
#include "stm32f4xx.h"
#include "led.h"
#include "fft.h"
#include "structs.h"
#include "spi.h"
#include "defines.h"
#include "tm_stm32f4_usart.h"
#include "tm_stm32f4_usart_dma.h"
#define PI 3.14159265359
controller* ct;
int main(void)
{
	uint8_t *array;
	SystemInit();
	TM_USART_Init(USART1, TM_USART_PinsPack_2, 345600);
	TM_USART_DMA_Init(USART1);
	uint8_t sinc[] = {0xaa};
	ct = (controller *) malloc(sizeof(controller));
	ct->count = 0;
	led_init();
	spi_init();
	init_fft(ct);
	led_write(LED6_PIN, configuracao_default());
	irq0_init();
	while (1)
	{
		if (ct->full_buffer == 1)
		{
			ct->full_buffer = 0;
			led_write(LED5_PIN, 1);
			calc_fft(ct);
			led_write(LED5_PIN, 0);
			if (!TM_USART_DMA_Sending(USART1))
			{
				TM_USART_Send(USART1, sinc, 1);
				array = (uint8_t*)(&ct->espectro.VA);
				TM_USART_DMA_Send(USART1, array, 801*4*2);
				while((TM_USART_DMA_Sending(USART1)));
				array = (uint8_t*)(&ct->buffer_read.VA);
				TM_USART_DMA_Send(USART1, &array[(ct-
>buffer+1)*1600], 800*4*2);
			}
		}
	}
}
void DMA2_Stream2_IRQHandler(void)
{
	led_write(LED3_PIN, 1);
	DMA_ClearITPendingBit(DMA2_Stream2, DMA_FLAG_TCIF2);
	chip_deselect();
	if (!TM_USART_DMA_Sending(USART1))
	burst_to_buffer(ct);
	irq0_init();
	led_write(LED3_PIN, 0);
}
```

```
void EXTI0_IRQHandler(void) {
    EXTI_ClearFlag(EXTI_Line0);
    EXTI_ClearITPendingBit(EXTI_Line0);
    EXTI_DeInit();
    int last_page = get_last_page();
    led_write(LED4_PIN, last_page);
    dma_init( ct->burst_read, last_page);
}
```

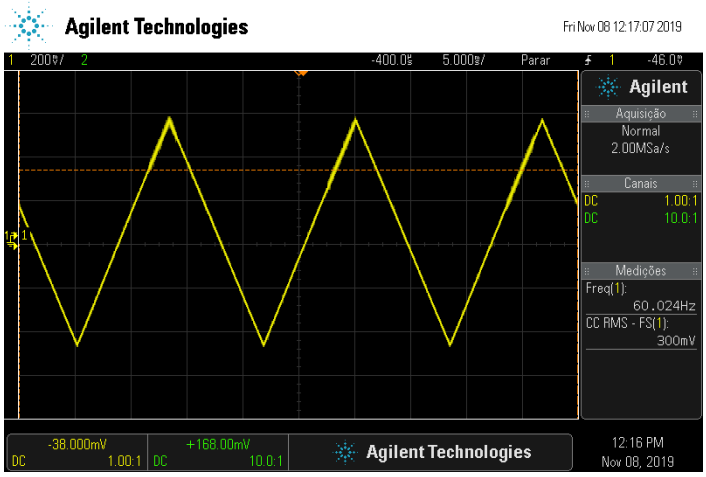
APÊNDICE C – ALGORITIMO DE VALIDAÇÃO NO MATLAB

```
clear
close all
clc

qnt = 801;
qnt_s = 1600;
espectro_r = zeros(qnt, 1);
espectro_i = zeros(qnt, 1);
espectro = zeros(qnt, 1);
wf = zeros(qnt_s, 1);
coordenada = 0:5:qnt*5-1;
s = serial('COM7', 'BaudRate', 115200);
fopen(s);
try
    while true
        if fread(s,1) == 170
            for j=1:qnt
                espectro_r(j) = fread(s,1,'float');
                espectro_i(j) = fread(s,1,'float');
            end
            for j=1:qnt_s
                wf(j) = fread(s,1,'float');
            end
            espectro = espectro_r + 1i*espectro_i;
            subplot(2,2,1);
            plot(wf);
            axis([0 1600 min(wf)*1.1 max(wf)*1.1]);
            title('Forma de onda');
            xlabel('amostras');
            ylabel('tensão');
            subplot(2,2,2);
            stem(coordenada, abs(espectro));
            title('FFT pelo Microcontrolador');
            xlabel('frequência');
            ylabel('amplitude');
            subplot(2,2,3);
            a = fft(wf, 1600, 1);
            stem(coordenada, abs(a(1:801)));
            title('FFT pelo Matlab');
            xlabel('frequência');
            ylabel('amplitude');
            drawnow;
        end
    end
catch
end
fclose(s);
```

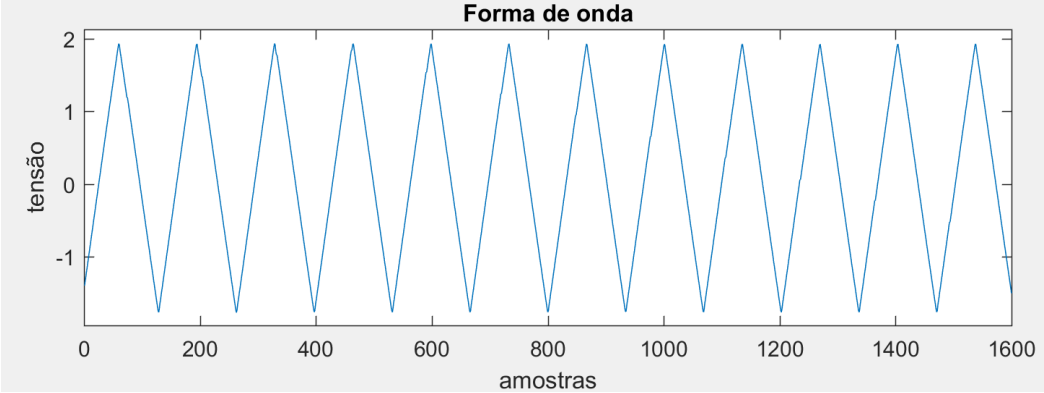
APÊNDICE D – RESULTADOS OBTIDOS PARA OUTRAS FORMAS DE ONDA

Figura 33: Forma de onda de 60 Hz triangular no gerador



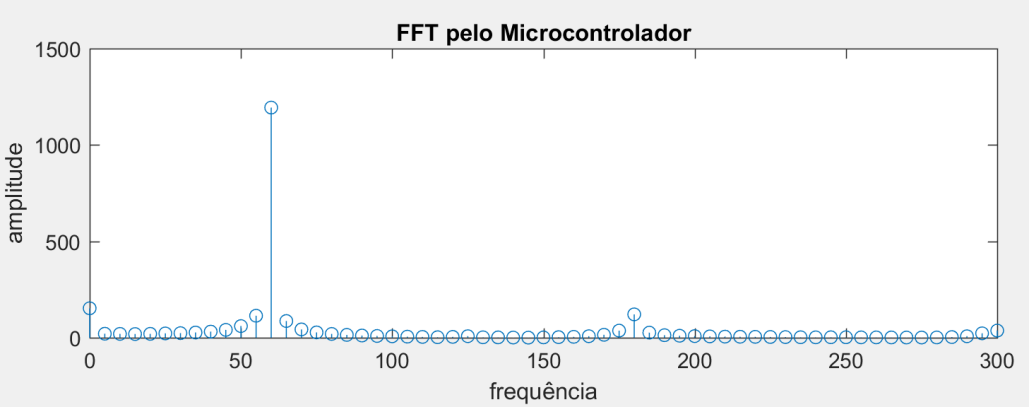
Fonte: (Elaborado pelo autor, 2019)

Figura 34: Forma de onda de 60 Hz triangular amostrada pelo ADE9000



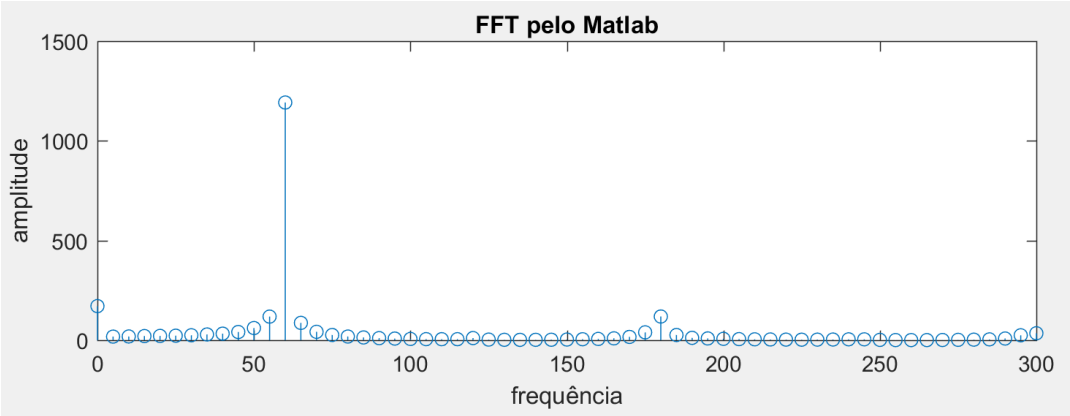
Fonte: (Elaborado pelo autor, 2019)

Figura 35: Espectro da forma de onda de 60 Hz triangular pelo microcontrolador



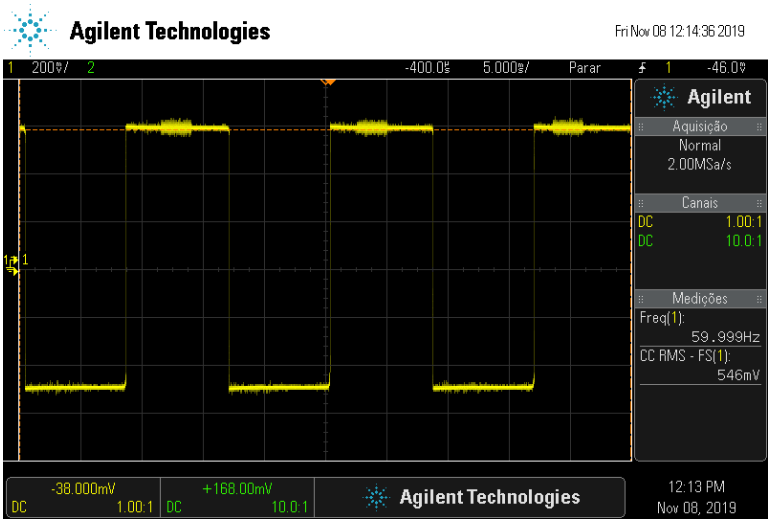
Fonte: (Elaborado pelo autor, 2019)

Figura 36: Espectro da forma de onda de 60 Hz triangular pelo Matlab



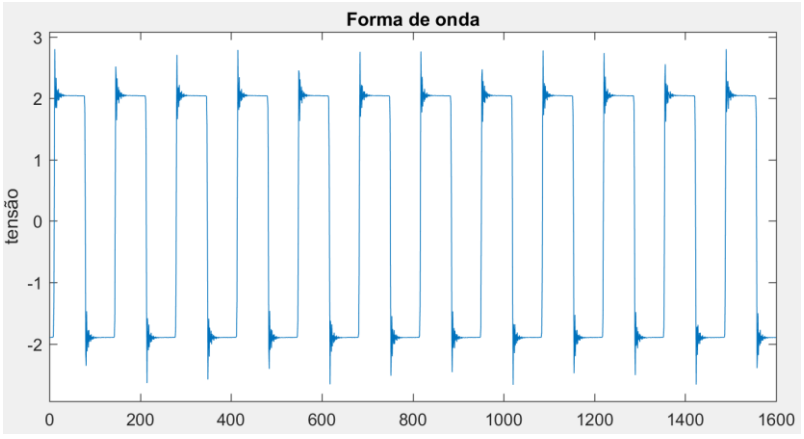
Fonte: (Elaborado pelo autor, 2019)

Figura 37: Forma de onda de 60 Hz quadrada no gerador



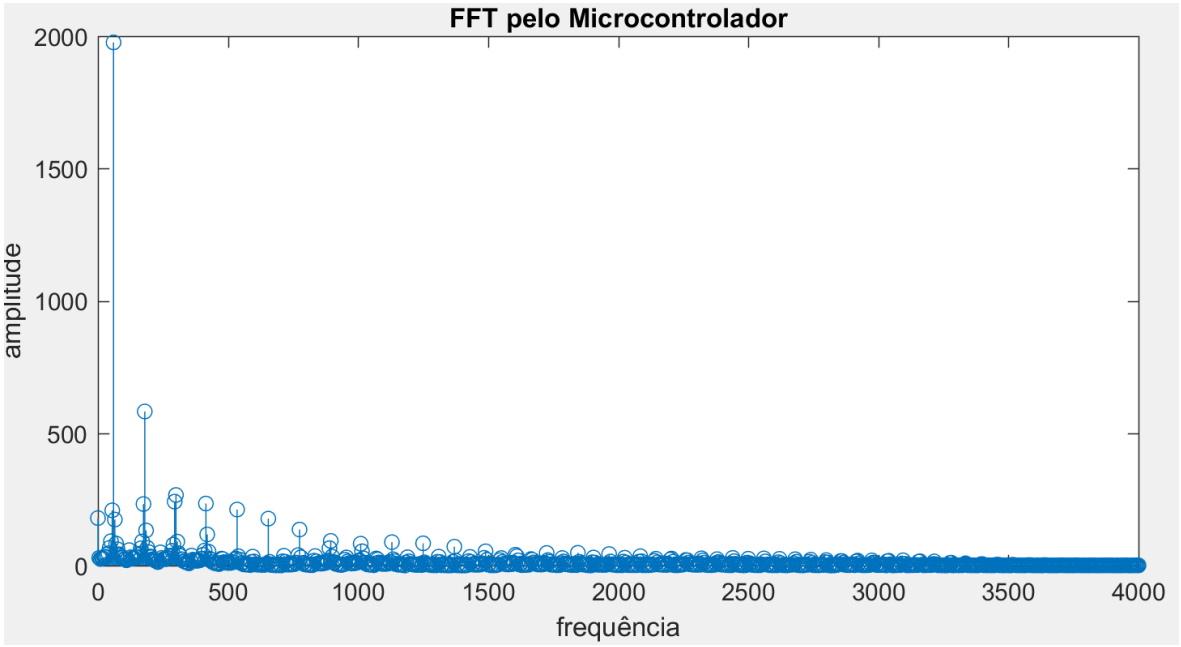
Fonte: (Elaborado pelo autor, 2019)

Figura 38: Forma de onda de 60 Hz quadrada amostrada pelo ADE9000



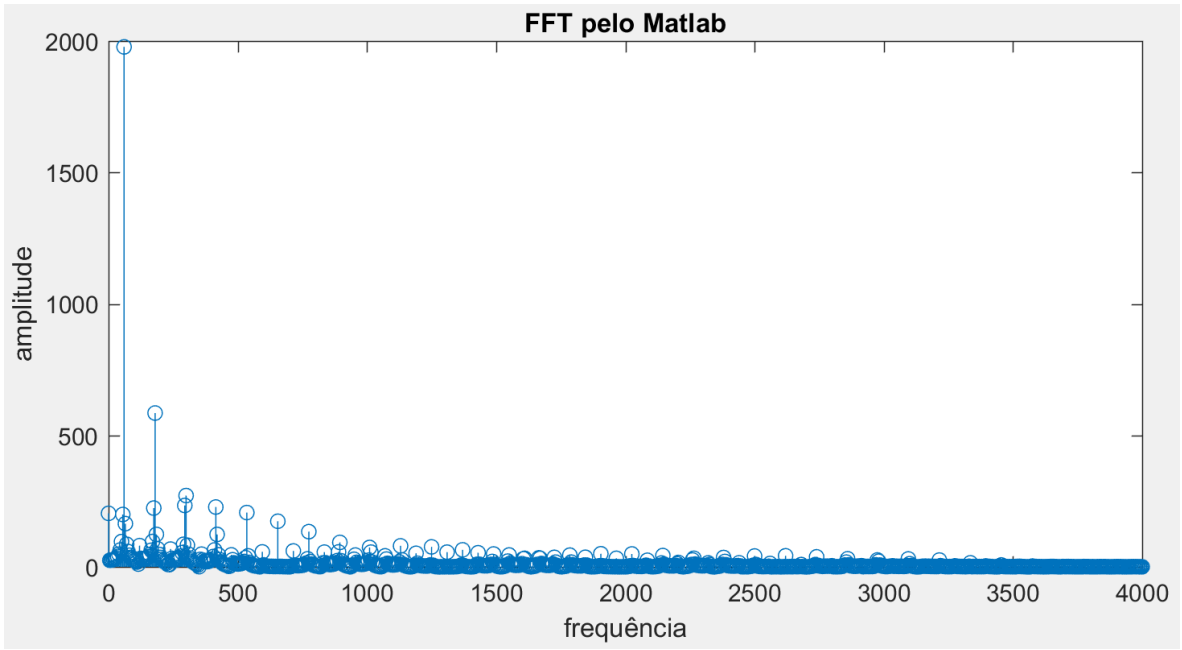
Fonte: (Elaborado pelo autor, 2019)

Figura 39: Espectro da forma de onda de 60 Hz quadrada pelo microcontrolador



Fonte: (Elaborado pelo autor, 2019)

Figura 40: Espectro da forma de onda de 60 Hz quadrada pelo Matlab



Fonte: (Elaborado pelo autor, 2019)