



Universidade de Aveiro Departamento de Eletrónica,
2014 Telecomunicações e Informática

**Luís Filipe Manteigas
da Cunha**

**Analisador das Características Elétricas de
um Circuito de Carga**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Eletrónica e Telecomunicações, realizada sob a orientação científica do Dr. António Ferreira Pereira de Melo, Professor Catedrático do Departamento de Eletrónica, Telecomunicações e Informática da Universidade de Aveiro

Este trabalho é dedicado especialmente aos meus avós, mãe, pai e irmão, assim como a toda a minha família que sempre acreditou e me encorajaram a realizar todos os meus objetivos.

O júri

Presidente

Prof. Dr. João Pedro Estima de Oliveira
Professor Associado da Universidade de Aveiro

Vogais

Prof. Dr. João Rafael da Costa Sanches Galvão
Professor Adjunto da Escola Superior de Tecnologia e Gestão do Instituto Politécnico de Leiria

Prof. Dr. António Ferreira Pereira de Melo
Professor Catedrático da Universidade de Aveiro

Agradecimentos

Com orgulho finalizo um dos meus grandes objetivos que sempre tive na minha vida. É impossível chegar a esta etapa e não lembrar-me das pessoas que me encheram de coragem, força, conhecimento e boa disposição para enfrentar e superar todos os obstáculos que se foram atravessando no meu percurso. Foram essas pessoas que fizeram do meu caminho um percurso memorável na qual não irei esquecer. A essas pessoas quero deixar os meus mais profundos agradecimentos por tudo o que me proporcionaram.

Aos meus avós, que apesar de já não estarem fisicamente presentes estarão sempre presentes na minha vida. Obrigado por todo o amor, carinho e afeto que me deram.

Aos meus pais e irmão, por tudo o que me proporcionaram desde sempre. Dedicção, esforço, amor, ternura, fé, compreensão, garra, confiança... são apenas algumas palavras daquilo que tiveram para comigo e que também me ensinaram o seu significado. Espero conseguir-vos compensar sempre com bastante orgulho.

A todos os meus familiares, que sempre trataram de mim da melhor maneira que podia ser tratado. Adoro cada um de vocês, adoro a nossa família.

Aos meus amigos mais chegados, pelas gargalhadas, sorrisos, confiança e toda a diversão e verdadeira amizade que temos. Foi sem dúvida essencial para conseguir chegar até aqui.

Ao meu orientador, por todo o esforço, paciência e dedicação que teve comigo. Proporcionou-me, sem dúvida, uma aprendizagem enriquecedora através dos seus vastos conhecimentos científicos. Uma pessoa que certamente não esquecerei, não só pelos conhecimentos que tem mas também como sendo uma pessoa extraordinária.

A todas as pessoas que conheci na firma, que sempre estiveram disponíveis para me ajudar e orientar da melhor maneira no meu projeto. Pessoas profissionais, amigas e sempre bem-dispostas que me trataram sempre como amigo.

Um muito obrigado a todos!

Palavras-chave

Energia, Analisador, Potência Ativa, Potência reativa, Potência Aparente, Corrente, Tensão, Fator de potência, Distorção Harmónica, FFT (Fast Fourier Transform), Processamento de Sinal, LabVIEW, Microcontrolador, Norma EN50160.

Resumo

Este trabalho consiste na elaboração de um protótipo de um analisador de energia elétrica para montagem em quadro elétrico. Este analisador tem como objetivo permitir verificar se a qualidade da energia elétrica fornecida cumpre com os parâmetros mínimos definidos na norma europeia EN50160.

Por ano estima-se que são enormes os prejuízos devido à fraca qualidade da energia elétrica [1]. Nas indústrias pode ocorrer defeitos de fabrico de produtos levando à não utilização e comercialização dos mesmos tendo estes de serem descartados.

Deste modo, a constante monitorização da energia tanto para efeitos de melhoramento da rede elétrica ou para possíveis reclamações dos consumidores por violações das cláusulas de contrato é indispensável.

Um outro fator que torna importante o uso destes analisadores é o facto de permitirem a monitorização dos custos associados à energia elétrica.

Apesar dos analisadores de energia se encontrarem maioritariamente em unidades fabris, começa-se a observar um crescimento em redes domésticas com o objetivo de reduzirem custos.

Nesta dissertação são apresentados conceitos teóricos indispensáveis à realização de todo o projeto. Aborda-se aspetos de aquisição de sinais, acondicionamento destes, montagem e prototipagem de hardware e toda a programação essencial para o correto funcionamento de um protótipo.

keywords

Energy, Analyzer, Real Power, Reactive Power, Apparent Power, Current, Voltage, Power Factor, Total Harmonic Distortion, FFT (Fast Fourier Transform), Signal Processing, LabVIEW, Microcontroller, Norm EN50160

abstract

This thesis has as objective the elaboration of a prototype of a power analyzer to be inserted into an electrical cabinet. The goal of the analyzer is to make sure that the delivered energy does meet the minimum defined in the European norm EN50160.

Per year it is estimated that the economic losses are huge due to poor power quality [1]. At the industries can occur manufacturing defects leading to products that cannot be commercialized and need to be wasted.

Thus, the constant monitoring of energy both for the purposes of improving the power grid or avoiding possible consumer complaints about violations of the terms of contract is essential.

Another advantage of the use of these analyzers is that they allow the monitoring of the cost of electricity.

Although the energy analyzers are mainly used in industry, they begin to show a growth in home networks in order to reduce costs.

In this thesis are also presented the theoretical concepts indispensable to the realization of the entire project. It deals with aspects of signal acquisition, signal conditioning, hardware assembly and prototyping and all the essential programming for the correct building of a prototype.

Conteúdo

LISTA DE FIGURAS	15
LISTA DE TABELAS	16
LISTA DE ACRÓNIMOS	17
1 INTRODUÇÃO	19
1.1 OBJETIVOS.....	19
1.2 MOTIVAÇÃO	20
1.3 ESTRUTURA DA DISSERTAÇÃO	20
2 ESTADO DA ARTE	23
3 FUNDAMENTAÇÃO TEÓRICA	25
3.1 PROCESSAMENTO DIGITAL DO SINAL	25
3.2 AMOSTRAGEM	26
3.3 QUANTIFICAÇÃO E CODIFICAÇÃO	27
3.4 SÉRIES DE FOURIER.....	27
3.5 DISTORÇÃO HARMÓNICA TOTAL (THD)	30
3.6 COMUNICAÇÃO SPI.....	30
3.7 VALOR EFICAZ (RMS)	31
3.8 POTÊNCIA.....	33
3.8.1 <i>Potência aparente e fator de potência</i>	33
3.8.2 <i>Potência ativa</i>	35
3.8.3 <i>Potência reativa</i>	36
4 NORMA EN 50160	39
5 DESENVOLVIMENTO DO PROJETO	41
5.1 AQUISIÇÃO E CONDICIONAMENTO DO SINAL	41
5.1.1 <i>Transformadores de intensidade</i>	42
5.1.2 <i>Divisor de tensão</i>	44
5.1.3 <i>Filtros</i>	45
5.1.4 <i>Multiplexagem de sinais</i>	46
5.1.5 <i>Buffer de tensão</i>	47
5.1.6 <i>Circuito somador em tensão não inversor</i>	47
5.1.7 <i>Regulador de tensão</i>	48
5.1.8 <i>Referência à massa</i>	50
5.2 CONSTRUÇÃO DO PRIMEIRO PROTÓTIPO DO PROJETO	51
5.2.1 <i>Kit de desenvolvimento Stellaris LM3S8962</i>	51
5.2.2 <i>Placa de aquisição (primeiro protótipo)</i>	53
5.2.2.1 <i>ADC (primeiro protótipo)</i>	54
5.2.2.2 <i>Divisores resistivos com filtragem (Secção A)</i>	56
5.2.2.3 <i>Transformadores de intensidade com filtragem (Secção B)</i>	58
5.2.2.4 <i>Multiplexação dos canais (Secção C e G)</i>	60
5.2.2.5 <i>Buffer de desacoplamento (Secção D)</i>	61
5.2.2.6 <i>Circuito somador de tensão e regulador de tensão com buffer (secção F e E)</i>	62
5.2.3 <i>Software LabVIEW</i>	63

5.2.3.1	Programação LabVIEW	64
5.2.4	<i>Ensaio do protótipo</i>	67
5.3	ALTERAÇÃO DO SOFTWARE	69
5.3.1	<i>Resultado via programação em C</i>	70
5.4	SEGUNDO PROTÓTIPO	71
5.4.1	<i>Kit de desenvolvimento dsPICDEM-80</i>	72
5.4.2	<i>Placa de aquisição (segundo protótipo)</i>	74
5.4.2.1	Divisores resistivos com filtragem (Secção I).....	75
5.4.2.2	Transformadores de intensidade com filtragem (Secção II)	77
5.4.2.3	Regulador de tensão do segundo protótipo (Secção V)	78
5.4.2.4	Ensaio da placa de aquisição (segundo protótipo)	79
5.4.3	<i>Software</i>	82
5.4.3.1	Primeira abordagem.....	83
5.4.3.1.1	Resultados obtidos	85
5.4.3.2	Segunda abordagem.....	87
5.4.3.2.1	Resultados obtidos pelo novo algoritmo	87
6	CONCLUSÕES FINAIS	95
7	TRABALHO FUTURO	99
8	BIBLIOGRAFIA	101
	ANEXO A – CÓDIGO C DO PRIMEIRO PROTÓTIPO	103
	ANEXO B – DIAGRAMA DE BLOCOS DO PROGRAMA ELABORADO EM LABVIEW	107
	ANEXO C – SIMULAÇÃO DE CARGAS E POTÊNCIAS (MATLAB)	110
	ANEXO D – CÓDIGO EM LINGUAGEM C DO SEGUNDO PROTÓTIPO	110

Lista de figuras

Figura 2.1 – Esquema geral do analisador	24
Figura 3.1 – Conversão analógica digital	25
Figura 3.2 – Amostragem, quantificação e codificação.	26
Figura 3.3 – Sinais sinusoidais e sua soma	28
Figura 3.4 – FFT do sinal somado	29
Figura 3.5 - Diagrama de funcionamento do SPI	31
Figura 3.6 - Triângulo de potências	34
Figura 3.7 – Característica de carga em relação de vetores tensão com o de corrente	35
Figura 3.8 – Potência numa carga puramente resistiva	35
Figura 3.9 – Simulação de tensão e corrente com carga puramente indutiva	36
Figura 5.1 Esquema e símbolo de transformadores de intensidade	42
Figura 5.2 – Transformador de intensidade ICES-EBM	43
Figura 5.3 – Relação de transformação do ICES-EBM	43
Figura 5.4 – Esquema de leitura em tensão de um TI	44
Figura 5.5 – Divisor de tensão resistivo	44
Figura 5.6 - Circuito passa-baixo RC	45
Figura 5.7 – Filtros aplicados: a) Divisores Resistivos; b) Transformadores de Intensidade	45
Figura 5.8 – Esquemático de um multiplexer de duas entradas	46
Figura 5.9 – Esquema de um buffer de tensão	47
Figura 5.10 – Circuito somador em tensão não inversor	48
Figura 5.11 – Regulador de tensão	49
Figura 5.12 – Circuito regulador de tensão para tensões inferiores à de referência	49
Figura 5.13 – Referência à massa	50
Figura 5.14 – Esquema geral da solução multiplexada	51
Figura 5.15 – Kit <i>Stellaris LM3S8962</i>	52
Figura 5.16 – Esquemático da placa de aquisição	53
Figura 5.17 – Straight binary transfer characteristic	55
Figura 5.18 – Modo de transferência de informação da ADC <i>AD7928</i>	55
Figura 5.19 – Esquemático da placa da ADC	55
Figura 5.20 – Secção A da placa de aquisição	56
Figura 5.21 – Resposta em frequência do filtro com os divisores resistivos	58
Figura 5.22 – Secção B da placa de aquisição	59
Figura 5.23 – Resposta em frequência do filtro após TI's	60
Figura 5.24 – Secções da placa de aquisição: a) Secção C; b) Secção G	61
Figura 5.25 – Secção D da placa de aquisição - desacoplador	62
Figura 5.26 – Secções da placa de aquisição: a) Secção F; b) Secção E	63
Figura 5.27 – Diagrama da leitura de valores	65
Figura 5.28 – Configuração do SPI pelo LabVIEW	65
Figura 5.29 – Leitura e escrita do SPI no LabVIEW	66
Figura 5.30 – Fecho da configuração do SPI no LabVIEW	66

Figura 5.31 – Desenho 2D das placas de hardware no Ultiboard	67
Figura 5.32 – Vistas 3D das placas de hardware desenvolvidas	67
Figura 5.33 – Conclusão da construção das placas de hardware	68
Figura 5.34 – Diagrama de blocos do programa em C no <i>Stellaris LM3S8962</i>	70
Figura 5.35 Diagrama do segundo protótipo	72
Figura 5.36 – Kit de desenvolvimento <i>dsPICDEM</i>	72
Figura 5.37 – Esquemático da placa de aquisição e acondicionamento de sinal do segundo protótipo	74
Figura 5.38 – Divisores resistivos e filtragem da placa de aquisição e acondicionamento do segundo protótipo	75
Figura 5.39 – Resposta em frequência do filtro com os divisores resistivos do segundo protótipo	77
Figura 5.40 – Transformadores de corrente e aquisição do segundo protótipo	77
Figura 5.41 – Resposta em frequência do filtro após os TI's do segundo protótipo	78
Figura 5.42 – Regulador de tensão do segundo protótipo	79
Figura 5.43 – Desenho 2D da Placa de Hardware do segundo protótipo	79
Figura 5.44 – Vistas 3D da placa de hardware do segundo protótipo	80
Figura 5.45 – Conclusão da construção do segundo protótipo	80
Figura 5.46 – Verificação de diferença de fases provocada pela placa de aquisição	82
Figura 5.47 – Diagrama de aquisição de valores	83
Figura 5.48 – Diagrama de fluxo de sinal do programa	85
Figura 5.49 – Novo diagrama de aquisição de valores	87
Figura 5.50 – Sinal capturado	88
Figura 5.51 – Sinais idênticos capturados	89
Figura 5.52 – Amplitude das harmónicas	91
Figura 5.53 – Captura de corrente e tensão	93

Lista de tabelas

Tabela 2-1 – Tabela comparativa de analisadores	23
Tabela 4-1 – Características da norma EN 50160	40
Tabela 4-2 Tabela das percentagens individuais máximas de cada harmónica de tensão	40
Tabela 5-1 – Tabela de leitura dos valores em tensão DC	68
Tabela 5-2 – Frequências de corte	81
Tabela 5-3 – Verificação dos ciclos completos dos timers	86
Tabela 5-4 – Comparação de valores	89
Tabela 5-5 – Magnitudes das harmónicas pelo cálculo da FFT	90
Tabela 5-6 – Teste ao algoritmo de cálculo da frequência	93
Tabela 5-7 – Resultados obtidos da captura de valores pelo microcontrolador	93

Lista de acrónimos

[ADC](#) – Analogic Digital Convert
[CPU](#) – Central Processing Unit
[DC](#) – Direct Current
[DFT](#) – *Discrete Fourier Transform*
[DMA](#) – *Direct Memory Access*
[FDM](#) – Frequency Division Multiplexing
[FFT](#) – Fast Fourier Transform
[LV](#) – Low Voltage
[MV](#) – Medium Voltage
[Op-amp](#) – Operational Amplifier
[PCB](#) – Printed Circuit Board
[PIC](#) – Peripheral Interface Controller
[QREN](#) – Quadro de Referencia Estratégica Nacional
[RAM](#) – Random Access Memory
[RMS](#) – Root Mean Square
[SF](#) – Série de Fourier
[SI](#) – Sistema Internacional
[SPI](#) – Serial Peripheral Interface
[TDM](#) – Time Division Multiplexing
[THD](#) – *Total Harmonic Distortion*
[TI](#) – Transformadores de Intensidade
[TRMS](#) – True RMS
[VA](#) – Volt-Ampere
[VAr](#) – Volt-Ampère-reactivo
[W](#) – Watt

1 Introdução

A energia elétrica é uma das fontes de energia mais utilizadas na sociedade moderna. O seu uso tornou-se indispensável e é graças a ela que enormes quantidades de equipamentos industriais, domésticos e outros operam facilitando as tarefas quotidianas na sociedade moderna melhorando assim a qualidade de vida das pessoas.

Por ser tão indispensável à sociedade moderna, tornou-se importante garantir a distribuição desta e assegurar uma boa qualidade da energia fornecida aos consumidores. O facto de a energia elétrica fornecida por vezes não ter a qualidade necessária para os equipamentos que a utilizam, resulta em enormes prejuízos podendo mesmo, na pior situação, ocorrer danos permanentes nos equipamentos. Uma má qualidade de energia elétrica pode originar que os equipamentos funcionem deficientemente e deem origem a produtos com defeitos, causando elevados prejuízos principalmente se for a nível industrial.

A norma EN50160 é uma norma europeia que estabelece os parâmetros mínimos de qualidade a que a energia elétrica tem de obedecer. Esta norma estabelece parâmetros relativos à forma de onda da energia elétrica entregue ao fornecedor, à sua frequência e quais os desvios máximos dos vários parâmetros da onda.

A energia elétrica em muitos países é cara, este facto faz com que seja necessário encontrar formas eficientes de reduzir os seus custos. Uma forma de determinar onde estão os maiores desperdícios de energia consiste em recorrer ao uso de analisadores de energia. Estes analisadores podem fornecer informações bastante relevantes para a gestão do consumo de energia e também verificar a qualidade da mesma.

O uso de analisadores é especialmente importante nas indústrias que consomem quantidades massivas de energia e onde os equipamentos necessitam de uma qualidade de energia bastante boa para o seu correto funcionamento. Estes analisadores para além de fornecerem dados acerca das características da energia tendo em vista a sua melhor gestão, servem também para registar possíveis anomalias que ocorram na distribuição da energia e que coloquem esta fora das cláusulas dos contratos estabelecidos entre o fornecedor e o consumidor. Desta forma o consumidor poderá ter provas que podem levar a possíveis reclamações por violação dos contratos e assim poderem ser indemnizados.

1.1 Objetivos

Esta dissertação tem como objetivo a construção de um analisador das características da energia elétrica possuindo baixo custo destinado a uso em pequenas indústrias e serviços e também uso doméstico.

O analisador a desenvolver deve permitir calcular o fator de potência, a distorção harmónica, medir as tensões, as correntes, a frequência, o verdadeiro valor RMS e as potências ativa, reativa e aparente. O aparelho deverá conseguir analisar estes parâmetros tanto para redes monofásicas, bifásicas e trifásicas. Deverá conseguir obter os dados necessários de modo a se conseguir verificar se os parâmetros analisados se encontram dentro do que está estabelecido na norma EN50160.

Uma outra funcionalidade deste equipamento será a de detetar se está a existir consumo ou geração de energia por parte da carga, possuindo disponível uma saída digital que indica se existe ou não geração e que poderá ser utilizada em outros equipamentos.

1.2 Motivação

Tendo sido feita uma análise geral dos objetivos e do conceito na qual este projeto se teria de desenvolver representou uma grande motivação para me integrar neste projeto. Isto porque representava um desafio abrangendo várias áreas de conhecimento científico, tais como, o processamento digital de sinal, a eletrónica digital, a eletrónica analógica, a instrumentação e a programação de sistemas embutidos.

O facto de o projeto se desenvolver nas instalações da empresa SELMATRON, possibilitando uma integração com uma equipa profissional num ambiente real de desenvolvimento e testes de produtos, foi um fator bastante importante para a motivação do autor para com o projeto.

Uma outra motivação para desenvolver este produto foi a possibilidade de poder aprofundar os conhecimentos no que respeita a normas relativas a energia eléctrica.

1.3 Estrutura da dissertação

Esta dissertação divide-se em sete capítulos.

No capítulo 1 apresenta-se o assunto abordado nesta dissertação salientando a importância, o impacto e as razões pela qual existe a necessidade de se monitorizar a energia eléctrica. Também se explica o objetivo desta dissertação (desenvolvimento de um analisador de energia eléctrica) assim como a motivação do autor para o desenvolvimento deste projeto.

No capítulo 2 é apresentado o estado da arte, onde se mostra alguns equipamentos de monitorização de energia eléctrica assim como as suas principais diferenças nas suas características.

No capítulo 3 faz-se uma abordagem teórica dos assuntos relevantes utilizados no desenvolvimento do projeto. Discutem-se os aspetos importantes no que diz respeito à recolha e processamento de sinais para análises das suas características. Também se apresentam os diferentes tipos de potências elétricas e como são medidas.

No capítulo 4 mostra-se aspetos importantes da norma europeia EN50160 que define parâmetros mínimos de qualidade de energia que o equipamento a desenvolver deve conseguir avaliar.

No capítulo 5 inicia-se o desenvolvimento físico do projeto. Começa-se por se fazer uma análise das partes essenciais dos protótipos construídos apresentando os passos da construção de cada um deles assim como as soluções de *software* implementadas.

No capítulo 6 apresenta-se as conclusões relativas a todo o projeto e ao seu desenvolvimento.

No capítulo 7 faz-se uma reflexão acerca do trabalho futuro que poderá ser feito para melhoramento do produto.

2 Estado da arte

Desde as primeiras experiências com energia elétrica que houve a necessidade de se fazer medições relativas a esta. Atualmente existe no mercado vários tipos de equipamentos que fazem leituras da energia elétrica. Esses equipamentos diferem nos seus custos dependendo das suas características físicas e da precisão das grandezas das leituras que conseguem obter.

Existem analisadores de leituras monofásicas, bifásicas e trifásicas. Em geral têm características de medição comuns tais como tensão por fase, corrente por fase, frequência e potência instantânea e total. Dependendo da qualidade e precisão de parâmetro que o analisador consegue monitorizar vai fazer com que o seu custo varie. Quanto mais parâmetros da energia conseguir monitorizar maior vai ser o seu custo. Outra diferença que determina o custo depende dos analisadores serem portáteis ou fixos. Os fixos são normalmente colocados em quadros elétricos sendo a sua maior procura para utilização em instalações fabris embora já se possam encontrar estes dispositivos em algumas redes domésticas. Os analisadores portáteis são mais úteis quando se necessita de fazer-se análises de energia em vários pontos distintos. São fáceis de transportar mas possuem características que os tornam geralmente mais caros.

A título de exemplo são apresentados na Tabela 2-1 três analisadores diferentes, dois portáteis (Fluke 434 Series II e ACD-45PQ da AMPROBE) e um fixo (GE – MTDIN).

	Fluke 434 Series II	GE - MTDIN	AMPROBE ACD-45PQ
Modelo:	Portátil	Montagem	Portátil
Tipo de Entrada:	Trifásico	Trifásico	Monofásico
Tensão Máxima:	1000Vac ($\pm 0,5\%$)	300Vac ($\pm 0,5\%$)	600Vac ($\pm 0,5\%$)
Frequência:	42.50 Hz a 57.50 Hz @ 50Hz nominal 51.00 Hz a 69.00 Hz @ 60Hz nominal	45 Hz a 65Hz	5Hz a 500Hz
Precisão de Potência:	1 %	1 %	$\pm 2\%$
Preço:	4047,97€	361,46€	168,66€

Tabela 2-1 – Tabela comparativa de analisadores

São analisadores de energia elétrica com objetivos diferentes. O MTDIN da empresa GE destina-se principalmente à montagem em quadros elétricos permitindo a medição de grandezas elétricas efetuar o controlo efetivo do sistema [2]. É um analisador barato, comparado com o Fluke, possuindo em relação a este menos funcionalidades e menor precisão em algumas grandezas. O Fluke é um analisador portátil com o objetivo de monitorizar o custo adicional da energia gasta devido a uma possível fraca qualidade da energia, é ideal para caracterização da qualidade de energia, realização de estudos de carga e capturar eventos de tensão difíceis de medir durante um período definido de tempo [3]. O analisador da AMPROBE difere especialmente dos outros por ser monofásico. É um analisador compacto e de fácil utilização. Possui uma pinça amperimétrica que abre de modo a poder-se facilmente contornar os condutores de corrente em sítios difíceis de alcançar [4].

Esta dissertação tem como objetivo construir um analisador de energia elétrica trifásico, de baixo custo, para montagem em quadros elétricos, ou seja, na gama de valores de custo inferior à do analisador MTDIN. A Figura 2.1 ilustra o esquema do analisador a desenvolver.

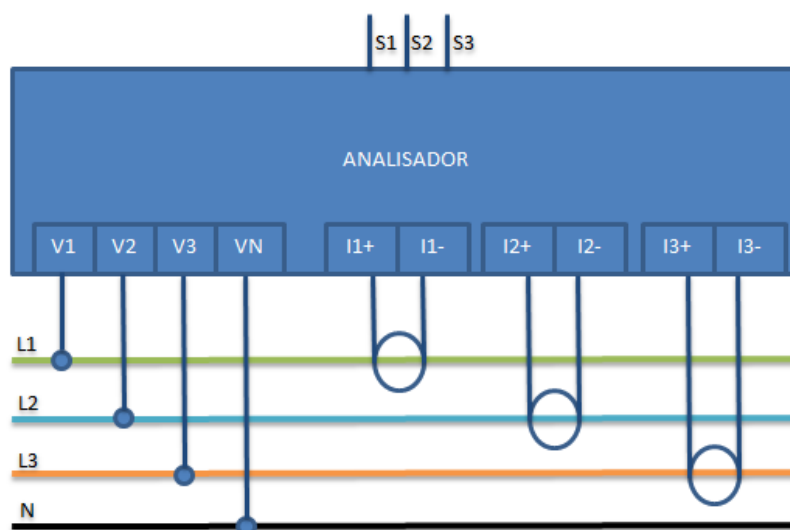


Figura 2.1 – Esquema geral do analisador

L1, L2, L3 e N são as linhas de fases da tensão da rede elétrica, N o neutro, V1, V2, V3 e VN são as entradas de leitura de tensão do analisador, I1+, I1-, I2+, I2-, I3+ e I3- são as entradas de leitura de corrente e S1, S2 e S3 são os sinais de direção de fluxo de energia por fase.

O analisador fará a medição das tensões e correntes de cada fase para calcular os parâmetros da energia elétrica pretendidos.

3 Fundamentação teórica

Os princípios fundamentais que regem a produção de energia elétrica a partir de outras formas de energia foram descobertos durante a década 1820 por Faraday. Hoje em dia a teoria encontra-se bem estabelecida pelo que a seguir apenas se discutem os assuntos relacionados com o projeto que se desenvolveu.

3.1 Processamento digital do sinal

Para se fazer análises e controlo de parâmetros físicos recorre-se muitas vezes a processadores digitais por estes possuírem muitas vantagens devido ao modo como manipulam os dados. Têm capacidade de efetuar cálculos bastante rápidos e assumir um controlo de processos com tempos de resposta também bastante rápidos. Funcionam apenas com dois valores lógicos (0 e 1), ou sejam, são processadores binários. Para isso existe a necessidade de transformar as grandezas físicas, geralmente analógicas em amplitude e contínuas no tempo, em sinais digitais em tempo discreto. Esta tarefa de conversão é geralmente feita recorrendo a um dispositivo designado por ADC (Em inglês, *Analog to Digital Converter*) [5]. A Figura 3.1 representa este processo típico de conversão.

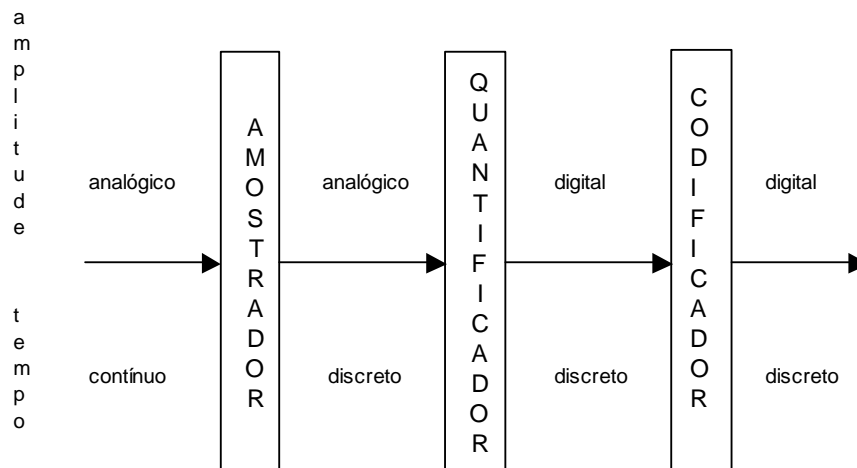


Figura 3.1 – Conversão analógica digital

Inicialmente faz-se a amostragem do sinal onde se transforma o sinal de tempo contínuo em tempo discreto. Posteriormente procede-se à quantização, onde se atribui à amplitude do sinal valores discretos e bem definidos transformando assim num sinal digital. Por fim pode haver um processo de codificação com o objetivo de reduzir a quantidade de informação necessária para representação dos níveis atribuídos pelo processo de quantificação, Figura 3.2 [5].

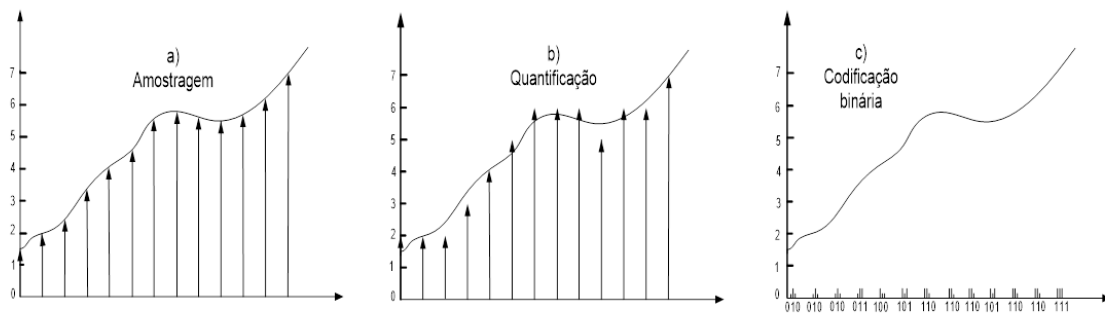


Figura 3.2 – Amostragem, quantificação e codificação.

Neste projeto houve a necessidade de se usar um microcontrolador digital e de fazer-se este processo de conversão. Nas próximas subsecções irá aprofundar-se melhor estes aspetos assim como fundamentos teóricos necessários à realização do projeto.

3.2 Amostragem

Um aspeto muito importante na aquisição de sinal digital passa por garantir que o intervalo entre amostras é suficientemente curto de modo a que mais tarde seja possível recuperar o sinal analógico inicial. Isto é possível por a energia nos sistemas não se poder alterar instantaneamente o que garante que os sinais não se modificam mais rapidamente do que um determinado ritmo que está relacionado com a largura de banda do mesmo sinal. Esta condição é representada matematicamente através do teorema de Nyquist que nos diz qual a frequência de amostragem mínima que se deve impor na aquisição de sinais quando se sabe a frequência máxima do sinal a capturar. Isto garante que mais tarde seja possível recuperar o sinal inicial através de uma operação de filtragem uma vez que não se verificarão fenómenos de Aliasing, isto é, de sobreposição de espectros no sinal amostrado.

Para um sinal na qual apresenta uma frequência máxima B , este teorema diz-nos que a frequência de amostragem deverá ser sempre superior ao dobro de B (frequência de Nyquist), ou seja:

$$f_a > 2B \quad (3-1)$$

Tendo-se como objetivo fazer uma avaliação do sinal até à 25ª harmónica do sinal da rede eléctrica de valor 50 Hz, então significa que

$$f_a > 2 \cdot N_h \cdot f_r = 2 \cdot 25 \cdot 50 = 2500 \text{ Hz} \quad (3-2)$$

Na qual N_h é o número da maior harmónica da qual se pretende analisar e f_r a frequência fundamental da rede eléctrica.

Na prática costuma-se utilizar-se uma frequência de amostragem da ordem de 5 vezes a frequência de Nyquist, no presente caso, devido a limitações físicas dos microcontroladores usados e ao cálculo da FFT (que é explicada no capítulo 3.4) usou-se uma frequência de

amostragem de 6400Hz de modo a que para cada ciclo da frequência da rede elétrica fosse capturado um número de amostras igual a uma potência de 2, ou seja, como o período do sinal da energia elétrica (T_{rede}) é de 20ms, para se ter um número de 128 amostras ($N = 2^7 = 128$) por período (T_a) será dado pela equação:

$$T_a = \frac{T_{rede}}{N} = \frac{20ms}{128} = 156,25us \quad (3-3)$$

O que corresponde à frequência de amostragem:

$$f_a = \frac{1}{T_a} = \frac{1}{156,25 us} = 6400 Hz \quad (3-4)$$

3.3 Quantificação e codificação

Uma vez que os equipamentos de processamento digital de sinal não conseguem manusear sinais com uma distribuição contínua de amplitudes torna-se necessário digitalizar as amostras, isto é, reduzir as amplitudes possíveis a um número finito de valores. Esta operação designa-se por quantificação da amplitude das amostras e introduz um erro de processamento que se designa por erro de quantificação que será tanto menor quanto maior for o número de níveis de amplitude possíveis. A parte que faz a conversão do sinal analógico para digital é a ADC (Analogic Dgital Convert, em Inglês). Como se trata de um equipamento funcionando na base 2 (binário) é necessário converter as amplitudes das amostras em sinais binários, isto é, codificar as amplitudes discretas em sequências de 0's e 1's, que constituem *words*. Assim o número de níveis de quantificação dependerá do comprimento da palavra do equipamento que se utilize. A fórmula que nos dá a resolução da ADC será a seguinte:

$$R = \frac{\Delta V}{2^n} [V/Nível] \quad (3-5)$$

Na qual R é a resolução, ΔV é o intervalo dos valores de entrada suportado pela ADC e n o número de bits da palavra da ADC.

Com isto o erro máximo cometido pela quantização será metade da resolução pois, na pior hipótese, o valor real estará exatamente entre dois níveis.

3.4 Séries de Fourier

A Série de Fourier (SF) é uma ferramenta matemática que decompõe uma função periódica numa soma de funções sinusoidais. É a transformada de Fourier para funções periódicas. Uma série de Fourier correspondente a uma função com um período T é uma soma infinita de funções sinusoidais (seno e cosseno), cada uma com uma frequência que é um múltiplo inteiro de $\frac{1}{T}$ (Frequência da componente fundamental) e pode incluir uma constante que corresponde à componente DC (Componente valor médio do sinal).

A Série de Fourier é bastante importante na área de processamento de sinal. Ela está presente em inúmeras aplicações nos dias de hoje que ajudam especialistas a resolver uma enorme variedade de problemas relacionados com a análise e processamento de sinais. Tem como objetivo separar todas as componentes sinusoidais presentes num sinal podendo mostrar discriminadamente as suas amplitudes e fases ao longo das frequências de que é composto.

No caso do sinal se encontrar amostrado no domínio do tempo apenas se dispõe de um conjunto de amostras igualmente espaçadas que para serem analisadas terá de se recorrer à Transformada de Fourier Discreta (*DFT – Discrete Fourier Transform*) obtendo-se uma lista de coeficientes de sinusoidais complexas [6].

O cálculo, via computador, da *DFT* pode contudo ser simplificado usando a transformada de Fourier rápida (*FFT – Fast Fourier Transform*) que constitui um método eficiente de transformar uma série de amostras (série no tempo) num conjunto de impulsos situados nas frequências das harmónicas e cuja amplitude está relacionada com a amplitude das mesmas. A *FFT* exige contudo que o número de amostras a processar seja uma potência de 2. Não se apresenta aqui o algoritmo que permite o cálculo da *FFT* por este ser muito complicado e atualmente já existem processadores PIC (*Peripheral Interface Controller*) que já possuem bibliotecas onde se inclui instruções para o cálculo da *FFT* (*As dsPIC – Digital Signal Controllers*).

A título de exemplo, na Figura 3.3 é apresentado dois sinais, o sinal 1 de amplitude 5 unidades e frequência 10Hz e o sinal 2 de amplitude 2 unidades e frequência de 15 Hz. Quando somados algebricamente o sinal resultante apresenta-se de forma pouco clara no sentido de não se conseguir distinguir quais as frequências que o compõem.

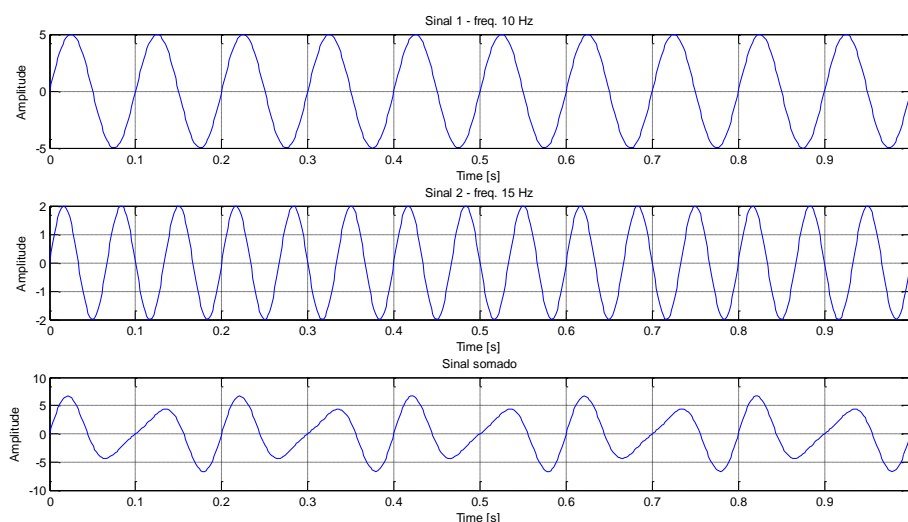


Figura 3.3 – Sinais sinusoidais e sua soma

Ao ser aplicada a *FFT*, ela decompõe o sinal nas frequências e mostra as respetivas amplitudes, tal como mostra a Figura 3.4 (*Ampliação do espetro*) obtida com o programa em Matlab que se segue.

```

%%FFT
clear all, clc, close all
fa = 512; Ta = 1/fa; L = 512; t = (0:L-1)*Ta; f1=10; f2=15;
sinal1 = 5*sin(2*pi*f1*t); sinal2 = 2*sin(2*pi*f2*t);
sinalTotal = sinal1 + sinal2;
figure(); subplot(3,1,1); plot(t, sinal1); title('Sinal 1 - freq. 10 Hz');
xlabel('Time [s]'); ylabel('Amplitude'); grid on; subplot(3,1,2);
plot(t, sinal2); title('Sinal 2 - freq. 15 Hz'); xlabel('Time [s]');
ylabel('Amplitude'); grid on; subplot(3,1,3); plot(t, sinalTotal);
title('Sinal somado'); xlabel('Time [s]'); ylabel('Amplitude'); grid on;
NFFT = 2^nextpow2(L); Y = fft(sinalTotal,NFFT)/L;
f = fa/2*linspace(0,1,NFFT/2+1); figure; stem(f(1:30), 2*abs(Y(1:30)));
grid on; title('Espectro'); xlabel('Frequência (Hz)'); ylabel('|Y(f)|');

```

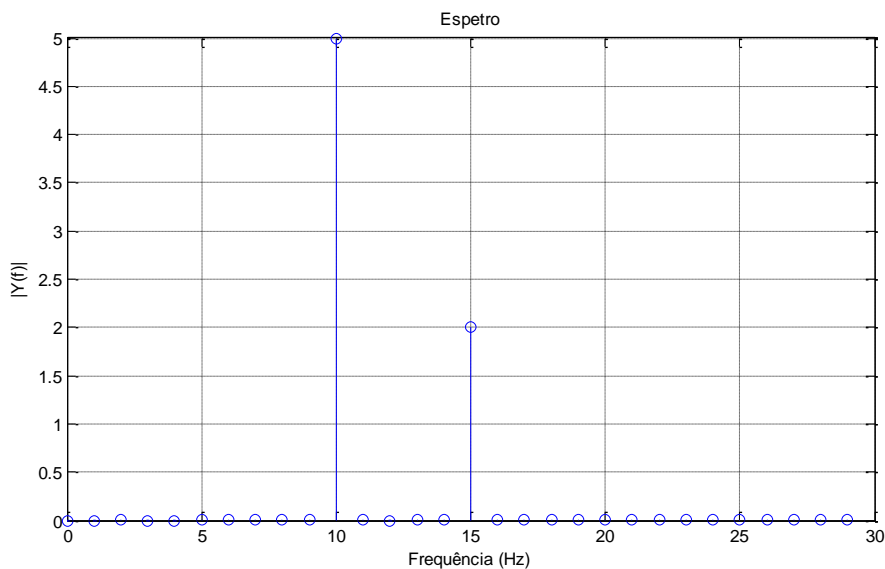


Figura 3.4 – FFT do sinal somado

A resolução em frequência da FFT irá depender da frequência de amostragem do sinal e do número de amostras que foram amostradas. A resolução da FFT é dada pela seguinte fórmula:

$$\Delta f = \frac{f_a}{N} \quad (3-6)$$

Onde Δf é resolução em frequência da FFT, f_a é a frequência a que foi amostrado o sinal e N o número de amostras analisadas pela FFT. No exemplo anterior foi feita a uma frequência de amostragem a 512Hz com um número de amostras de 512. Assim, usando a expressão (3-6), dá-nos uma resolução em frequência de 1Hz e é por essa razão que só são visualizadas, no gráfico anterior, componentes de 1Hz em 1Hz. O número de frequências distintas que a FFT pode calcular é metade do número de amostras analisadas, ou seja, $N/2$. Na Figura 3.4 apenas foram mostradas frequências relevantes para uma maior facilidade de visualização.

3.5 Distorção harmónica total (THD)

A distorção harmónica total, em inglês THD - *Total Harmonic Distortion*, constitui uma forma de avaliar a pureza de um sinal suposto ser sinusoidal mas que apresenta algumas harmónicas na sua constituição. O THD é a relação entre a potência das harmónicas presentes no sinal e a potência da frequência fundamental [7].

O THD é utilizado para caracterizar a linearidade de sistemas de áudio e a qualidade da energia de sistemas de potência eléctricos. A sua expressão é a seguinte:

$$\text{THD} = \frac{\sqrt{V_2^2 + V_3^2 + \dots + V_{inf}^2}}{V_1} \quad (3-7)$$

Da expressão pode-se verificar que num sinal idealmente puro, onde não existisse componentes harmónicas, o THD teria o valor de zero. Quanto maior for o valor de THD menor será a qualidade do sinal se se pretender que este seja puro.

3.6 Comunicação SPI

Atendendo a que no trabalho se utilizou inicialmente um microcontrolador comunicando com uma unidade condicionadora e amostradora dos sinais, tornou-se necessário recorrer a uma interface de comunicação entre as duas unidades. O interface imposto pelo fabricante do Kit é o SPI (Serial Peripheral Interface).

Comunicação SPI é um canal de interface definida inicialmente pela *Motorola*. Tem como objetivo ser usada para comunicar entre vários dispositivos periféricos de microcontroladores, tal como por exemplo, ADC's, memórias, displays, sensores, etc.

É uma arquitetura *Master-Slave* de ligação ponto a ponto onde só pode existir apenas um *Master*. O *Master* é o responsável por gerar o relógio de sincronização na transferência de informação. Apesar de poder estar ligado a vários *Slaves* apenas um só *Slave* é ativo de cada vez para comunicar com o *Master*. A comunicação SPI tem como vantagem de ser uma comunicação *full duplex*, ou seja, é uma comunicação bidirecional com transferência de informação simultânea nos dois sentidos. Além disso, é uma interface simples de se implementar a nível de *hardware*.

O SPI usa apenas quatro linhas para fazer a comunicação. Sendo elas as seguintes:

- MOSI - Master output , Slave Input
- MISO - Master Input, Slave Output
- SCLK - Serial Clock
- CS\ - Chip Select

O MOSI serve para fazer a transferência de informação do *Master* para o *Slave* enquanto que o MISO faz no sentido contrário. O *Master*, como já foi dito, gera o relógio de sincronização e passa-o através do sinal SCLK. O CS\ serve apenas para selecionar o *Slave* pretendido com que vai trocar informação.

No processo de comunicação, o SPI faz uso de dois *shift-registers*, um em cada dispositivo de transferência, ligados em anel. Uma representação gráfica apresenta-se na Figura 3.5.

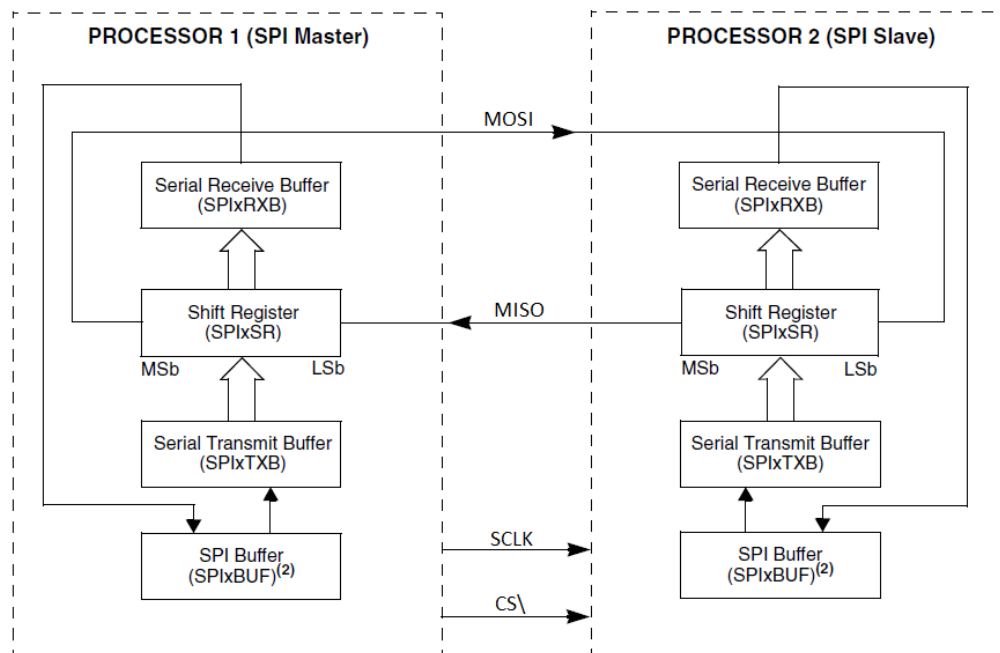


Figura 3.5 - Diagrama de funcionamento do SPI

A cada ciclo de relógio existe uma transferência de um bit entre unidades. No flanco ascendente do relógio faz-se a transferência de um bit para um sentido enquanto que no flanco descendente do relógio faz-se a transferência no sentido contrário. Quando o relógio fizer o mesmo número de ciclos que o comprimento do *shift-register*, toda a informação foi trocada entre eles.

3.7 Valor eficaz (RMS)

Verificando-se que as medidas que caracterizam a qualidade da energia lidam quase exclusivamente com valores quadráticos médios (Em Inglês RMS – Root Mean Square), isto é, valores eficazes da tensão e corrente, torna-se apropriado discutir esta noção matemática.

O valor eficaz de uma tensão ou corrente descreve-se matematicamente pelo valor quadrático médio dessa grandeza.

Para uma função contínua no tempo entre o intervalo de T1 a T2 o seu RMS é expressa da seguinte fórmula:

$$x_{rms} = \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} [f(t)]^2 dt} \quad (3-8)$$

Tendo em conta que se irá trabalhar com sinais amostrados no tempo, isto é discretos, o integral transforma-se num somatório e a expressão que permite o cálculo do RMS é a fórmula seguinte, onde para N amostras, $x_1 \dots x_N$, temos:

$$x_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2} = \sqrt{\frac{x_1^2 + x_2^2 + \dots + x_N^2}{N}} \quad (3-9)$$

É esta fórmula que é usada para se calcular os valores de RMS de tensão e corrente neste projeto, onde os valores de x_i são substituídos por V_i e I_i , respetivamente. Deste modo temos então o seguinte:

$$V_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N V_i^2} ; I_{rms} = \sqrt{\frac{1}{N} \sum_{i=1}^N I_i^2} \quad (3-10)$$

Quando se fala em rede elétrica, frequentemente usa-se o valor de RMS para se designar a tensão e/ou corrente. Apesar de a rede elétrica não ser ideal, ou seja, apresenta alguns desvios em alguns parâmetros do que era suposto, pode-se admitir que as distribuidoras fornecem uma corrente alternada sinusoidal. Se se substituir na fórmula do valor de RMS de função continua no tempo $f(t)$ por uma senoide pura de valor de pico V_p e frequência angular ω , então manipulando matematicamente pode-se chegar ao seguinte resultado [8]:

$$\begin{aligned} V_{rms} &= \sqrt{\frac{1}{T_2 - T_1} \int_{T_1}^{T_2} (V_p \sin(\omega t))^2 dt} = \sqrt{\frac{V_p^2}{T_2 - T_1} \int_{T_1}^{T_2} \frac{1 - \cos(2\omega t)}{2} dt} \\ &= V_p \sqrt{\frac{1}{T_2 - T_1} \left[\frac{t}{2} - \frac{\sin(2\omega t)}{4\omega} \right]_{T_1}^{T_2}} = V_p \sqrt{\frac{1}{T_2 - T_1} \left[\frac{t}{2} \right]_{T_1}^{T_2}} \\ &= V_p \sqrt{\frac{1}{T_2 - T_1} \cdot \frac{T_2 - T_1}{2}} = \frac{V_p}{\sqrt{2}} \end{aligned} \quad (3-11)$$

Note-se que o valor $V_{rms} = \frac{V_p}{\sqrt{2}}$ é apenas válido para o caso de se ter uma senoide pura. Caso o sinal não seja uma senoide pura o valor obtido usando esta expressão não corresponde ao verdadeiro valor RMS (TRMS - True RMS). O equipamento que se está a desenvolver calcula o verdadeiro valor RMS uma vez que não parte do valor de pico do sinal mas do somatório das amostras obtidas no tempo.

Deste modo quando se diz que a tensão da rede elétrica tem o valor de $230V_{rms}$, significa que em termos de amplitude a senoide apresenta-se com uma amplitude de aproximadamente 325V no caso de se tratar de uma senoide pura.

3.8 Potência

Chama-se potência à grandeza que mede a energia transferida por unidade de tempo e a sua unidade é o Watt (W). Pelo Sistema Internacional (SI), Watt é uma grandeza derivada que pode ser descrito também como Joule por segundo, ou seja, J/s. Desta forma, pode-se escrever a fórmula seguinte para o cálculo de potência média, onde ΔE é a energia transferida e Δt o intervalo de tempo que a energia demorou a ser transferida:

$$P_{med} = \frac{\Delta E}{\Delta t} \quad (3-12)$$

No limite, quando o intervalo de tempo tende para zero, ficaríamos com a potência instantânea, ou seja:

$$P_{ins} = \lim_{\Delta t \rightarrow 0} P_{med} = \frac{dE}{dt} \quad (3-13)$$

Que nos indica a potência consumida instantaneamente.

Em sistemas elétricos, a maneira comum de se medir potência é através do produto da corrente elétrica (I) pela tensão (V). Isto porque o trabalho realizado por uma campo elétrico de valor V , para transportar uma corrente $I = \frac{Q}{t}$, onde Q é a carga elétrica é dado por:

$$P(t) = \text{Trabalho por unidade de tempo} = \frac{Q \cdot V}{t} = I \cdot V \quad (3-14)$$

Assim, a fórmula apresentar-se-á da seguinte maneira:

$$P(t) = V(t) \cdot I(t) \quad (3-15)$$

Quando se pretende calcular o valor médio consumido pode-se manipular a formula matematicamente para a seguinte:

$$P_{med} = \frac{1}{T} \int_0^T I(t) \cdot V(t) dt \quad (3-16)$$

E assim nos dará o valor médio da potência consumida num dado intervalo de tempo T .

Contudo, em tensões alternadas sinusoidais são designadas diferentes tipos de potências. Essas potências são potência ativa, potência reativa e potência aparente. A explicação de cada uma delas é apresentada agora de seguida nos próximos subsecções.

3.8.1 Potência aparente e fator de potência

A potência aparente é a soma vetorial dos fasores de potência ativa com o da potência reativa, isto é, a soma da potência dissipada na carga (potência ativa) mais a potência absorvida e devolvida (potência reativa) pela carga [9]. Esta potência mede-se em VA (Volt-Ampere). A sua

fórmula de cálculo pode demonstrar-se ser o produto entre o valor eficaz da tensão e o valor eficaz da corrente, ou seja:

$$P_{apar} = V_{rms} \cdot I_{rms} \quad (3-17)$$

Atendendo a que uma impedância elétrica é matematicamente representável por um número complexo, com parte real que corresponde a uma componente de dissipação e uma parte imaginária que corresponde a uma componente reativa, podemos decompor a potência aparente em duas potências, potência ativa e reativa. A potência ativa corresponde à potência dissipada na parte real da impedância e a potência reativa relaciona-se com a parte imaginária da impedância, sendo absorvida e posteriormente devolvida.

Assim, podemos representar um “Triângulo de Potências” onde se mostra a relação entre todas estas potências (Figura 3.6) [10].

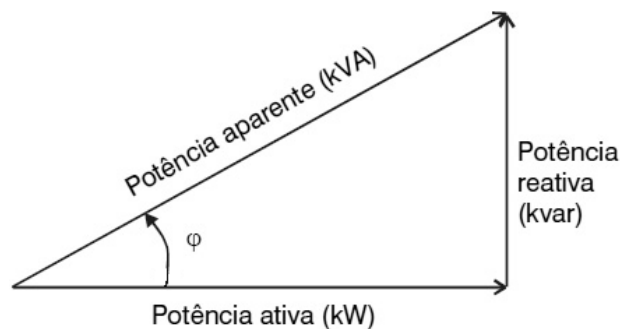


Figura 3.6 - Triângulo de potências

Através deste triângulo de potências pode-se calcular um parâmetro que se designa por fator de potência que é dado pela fórmula:

$$\cos \phi = \frac{P_{ativ}}{P_{apar}} \quad (3-18)$$

O fator de potência constitui assim uma forma de relacionar a potência ativa com a potência aparente. Quanto mais próximo este valor se aproximar do valor de 1, melhor, isto é, obter-se-ia um melhor rendimento se a potencia aparente coincidissem com a potência ativa não havendo assim potência reativa a circular no circuito. Isto porque, como será explicado melhor a seguir, a potência reativa não produz trabalho útil mas causa perdas nas linhas de transmissão de energia. Isto é um problema para a rede elétrica uma vez que limita a sua capacidade de transmitir energia ativa, razão pelo qual os fornecedores de energia a cobram bastante caro nas suas tarifas.

Quando o valor do fator de potência é positivo isto indica que a carga tem características indutivas, ou seja, a corrente está em atraso em relação à tensão. Por outro lado, quando o valor se apresenta negativo indica que a carga tem características capacitivas e a corrente está em avanço em relação à tensão. Uma ilustração deste fenómeno está representada na figura seguinte.

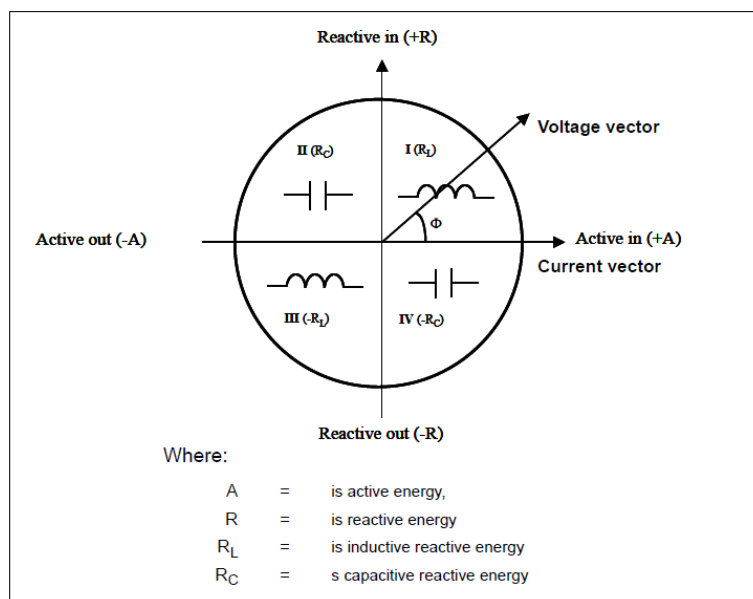


Figura 3.7 – Característica de carga em relação de vetores tensão com o de corrente

3.8.2 Potência ativa

Numa carga puramente resistiva as ondas de tensão e de corrente coincidem em todo o instante, ou seja, a carga não produz nenhum desfasamento entre elas. Deste modo, diz-se que apenas existe potência ativa e toda a energia é transmitida da rede elétrica para a carga. As unidades de potência ativa vêm em Watt. Na Figura 3.8 está simulado esta situação na qual se tem uma tensão alternada a 50 Hz de 230Vrms e uma corrente não desfasada de 5Arms. A potência ativa aparece a uma frequência ao dobro da tensão e sempre com valores positivos.

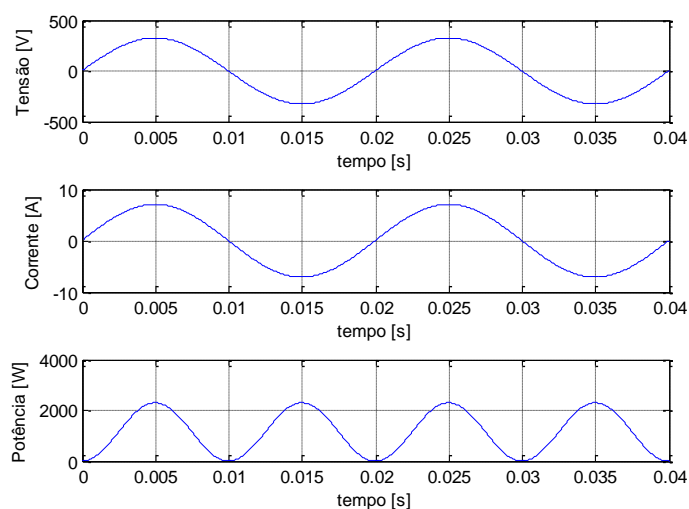


Figura 3.8 – Potência numa carga puramente resistiva

A potência aparente média que circula num circuito durante um determinado intervalo de tempo é dada pela fórmula:

$$P_{ativ} = \frac{1}{T_2 - T_1} \int_{T_1}^{T_2} V(t) \cdot I(t) dt \quad (3-19)$$

Como no presente caso trabalha-se com sinais amostrados, podemos converter o integral num somatório. Para o número de amostras suficientemente grande, N, podemos calcular a potência ativa através do seguinte somatório:

$$P_{ativ} = \frac{1}{N} \sum_{k=1}^N V(k) \cdot I(k) \quad (3-20)$$

3.8.3 Potência reativa

Para os casos em que a carga a ser alimentada pela rede elétrica possui componentes indutivas e/ou capacitivas, esta carga irá originar um consumo de potência reativa que é medida em Volt-Ampère-reactivo (VAR). Esta energia, que é retirada da rede elétrica mas não utilizada para trabalho útil, destina-se essencialmente a manter o campo eletromagnético nos aparelhos para que estes possam operar. Exemplos desses equipamentos são motores elétricos e transformadores que são muito usados nas indústrias.

Uma simulação de uma carga puramente indutiva a operar na rede elétrica encontra-se na figura seguinte. A tensão encontra-se desfasada 90° do sinal de corrente com uma amplitude de 5Arms.

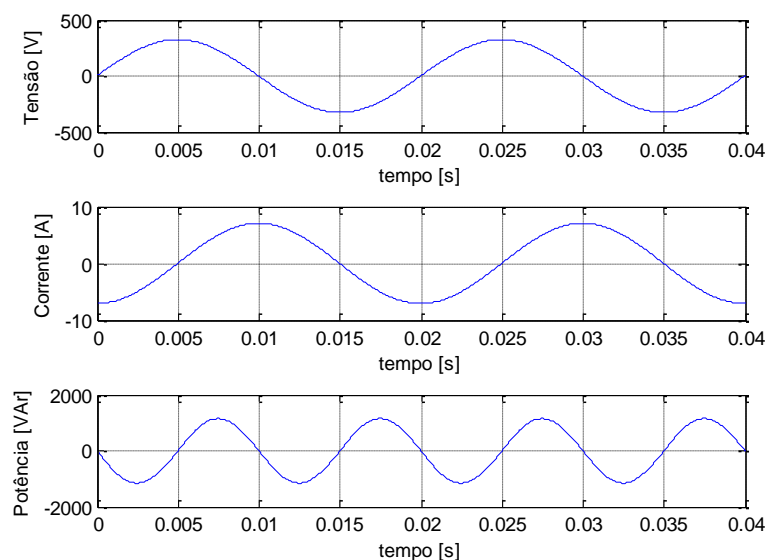


Figura 3.9 – Simulação de tensão e corrente com carga puramente indutiva

Pode-se verificar que a potência tem uma contribuição média nula ao longo do tempo, pois metade do tempo da onda está positiva e na outra metade está negativa.

O cálculo para se obter o valor da potência reativa recorre ao valor de potência aparente e ativa e tem a seguinte fórmula:

$$P_{reat} = \sqrt{P_{apar}^2 - P_{ativ}^2} \quad (3-21)$$

4 Norma EN 50160

Na sociedade moderna, a energia elétrica é um produto muito importante. As pessoas utilizam-na e necessitam-na para a realização de inúmeras tarefas na sua vida cotidiana. Tal como qualquer outro produto, a energia elétrica deve satisfazer certos parâmetros de qualidade. O cumprimento desses parâmetros é essencial para o correto funcionamento dos equipamentos aos quais a rede elétrica está a fornecer energia. Contudo, esses equipamentos juntamente com outros fatores externos causam distorções na qualidade da energia elétrica fornecida comprometendo assim as suas características. Muitas vezes, estas alterações são distorções não lineares o que complicam mais toda a qualidade do sistema. Assim, a qualidade da energia depende de um compromisso entre o fornecedor e o comprador.

Deste modo, métodos de medição e análise de energia elétrica são necessários para que seja possível avaliar a qualidade da energia permitindo aos fornecedores e compradores poderem realizar os seus contratos baseados nessas medições.

A norma EN 50160 é uma das normas europeias que caracteriza parâmetros mínimos de energia elétrica às quais os fornecedores devem obedecer. Embora seja norma europeia, em algumas regiões da europa são acrescentadas ainda mais algumas condições suplementares [1].

As condições impostas pela norma EN 50160 estão representadas na Tabela 4-1 e Tabela 4-2.

Na Tabela 4-1 são apresentadas características em percentagens para baixas tensões (LV – Low Voltage) e tensões médias (MV – Medium Voltage). Estas tensões são de fase para fase em tensões nominais RMS que não devem exceder 1000Vrms para LV e entre 1KVrms e 35KVrms para MV.

As características são apresentadas em médias para certos intervalos de tempo na qual os valores não devem exceder limites especificados em relação à componente fundamental.

Na Tabela 4-2 estão representadas as características das componentes harmónicas que não devem ser excedidas em relação à harmónica fundamental. Nesta tabela pode-se verificar que a análise deve de ter em conta as harmónicas até à 25ª.

Tendo em conta estas características apresentadas pela norma EN 50160 já se pode ter as referências necessárias para a criação de equipamentos de análise de energia elétrica que verifiquem os parâmetros pretendidos. É então baseado nesta norma que este projeto se desenvolveu.

Parâmetro	Especificação
Frequência	Frequência fundamental ao longo de 10 segundos: $\pm 1\%$ (49.5Hz – 50.5Hz), para 95% da semana $-6\% / +4\%$ (47Hz – 52Hz), para 100% da semana
Tensão	Média RMS ao longo de 10 minutos: $\pm 10\%$ para 95% da semana
Variações rápidas de tensão	Baixa tensão: 5% normal 10% esporádico $P_{IT} \leq 1$ para 95% da semana Média tensão: 4% normal 6% esporádico $P_{IT} \leq 1$ para 95% da semana
Cavas	Geral: duração <1s; profundidade <60%; Local devido a variações de carga: Baixa tensão: 10-50%; Média tensão: 10-15%
Interrupções de fornecimento de curta duração	Até 3 minutos Algumas dezenas/centenas de vezes ao ano Duração de 70% das ocorrências <1 segundo
Interrupções de fornecimento de longa duração	Superior a 3 minutos <10 - 50 ao ano
Sobretensão	Baixa tensão: 1500V RMS Média tensão: 170% com ligação à terra 200% sem ligação à terra
Transientes	Baixa tensão: geralmente $\leq 6000V$; Ocasionalmente maior; Tempo de subida: μs - ms
Falha de balanceamento de tensão	Média RMS de 10 minutos: $\leq 2\%$ para 95% da semana
Distorção harmónica	Ver próxima tabela

Tabela 4-1 – Características da norma EN 50160

Ímpares				Pares	
Não múltiplos de 3		Múltiplos de 3		Ordem	%
Ordem	%	Ordem	%		
5	6	3	5	2	2
7	5	9	1.5	4	1
11	3.5	15	0.5	6 ... 24	0.5
13	3	21	0.5		
17	2				
19	1.5				
23	1.5				
25	1.5				

Tabela 4-2 Tabela das percentagens individuais máximas de cada harmónica de tensão

5 Desenvolvimento do projeto

O projeto ao longo da sua execução passou por várias fases e alguns imprevistos técnicos. Tendo sido este projeto submetido a uma candidatura ao QREN (*Quadro de Referência Estratégica Nacional*) desenvolvido em âmbito empresarial ligado à empresa SELMATRON (situada em Esgueira – Aveiro), foi de início imposto que a solução técnica do produto passasse pela utilização de um Kit que a empresa já possuía, o *Stellaris LM3S8962* da *Luminary Micro*. Juntamente com este Kit também dever-se-ia usar o *software LabVIEW* da *National Instruments*. Toda a comunicação e programação entre esse Kit e o computador, para fases de desenvolvimento do projeto, seriam feitas com auxílio do programador *ULINK2* da *Keil*. Após uma árdua tentativa (cerca de 4/5 meses perdidos) concluiu-se que o *software LabVIEW* não permitia concretizar o objetivo a que se pretendia. Optou-se então por implementar no mesmo Kit os algoritmos desenvolvidos em linguagem C. Quando o projeto já se encontrava a progredir ocorreu que a unidade programadora *ULINK2* avariou. Atendendo ao custo desta unidade, optou-se então a SELMATRON por um outro *hardware*, *dsPICDEM 80*, que possui sobre o anterior as vantagens de ser mais económico, mais fácil de programar e dispor de vários conversores ADC internos.

Estes problemas causaram um substancial atraso na realização do projeto.

Nos próximos subcapítulos expõem-se o desenvolvimento do projeto em três fases:

- Construção do primeiro protótipo do projeto - utilizando o *software LabVIEW* com o Kit *Stellaris LM3S8962*
- Alteração do *software* – após o abandono do *LabVIEW* e recurso à linguagem C ainda com o Kit *Stellaris LM3S8962*
- Segundo protótipo – após a avaria do programador e recurso a um novo Kit de desenvolvimento

Inicialmente, no próximo subcapítulo, será apresentado os cálculos gerais de modo a conseguir-se projetar a placa de aquisição. De seguida, serão então apresentadas as três fases do projeto até se chegar à solução do segundo protótipo.

5.1 Aquisição e condicionamento do sinal

Atendendo a que os circuitos eletrónicos não podem funcionar ligados diretamente às redes elétricas por os valores das tensões e correntes serem elevados, torna-se necessário proceder ao condicionamento destes sinais, através de transdutores, de modo a que possam ser analisados pelas unidades eletrónicas. Assim, para se medir as correntes tem-se que se recorrer a transdutores de intensidade que podem ser do tipo transformadores de intensidade (TI) ou transdutores por efeito de Hall. Optou-se pelos TI's por serem mais económicos. Quanto à medida de tensões recorreu-se a divisores resistivos.

5.1.1 Transformadores de intensidade

Transformadores de intensidade (TI's), ou também designados por transformadores de corrente (Em inglês, Current Transformer), são componentes eletrônicos passivos que produzem aos terminais do circuito secundário uma corrente proporcional à que circula no circuito primário. São usados em sistemas com correntes alternadas e têm como principais funcionalidades medir e isolar galvanicamente as elevadas correntes e tensões que circulam no circuito primário.

Os TI's são constituídos por um enrolamento primário, um núcleo magnético e um enrolamento secundário [11]. Normalmente o enrolamento primário reduz-se a uma espira, o fio em que passa a corrente a medir, tal como é mostrado na Figura 5.1.

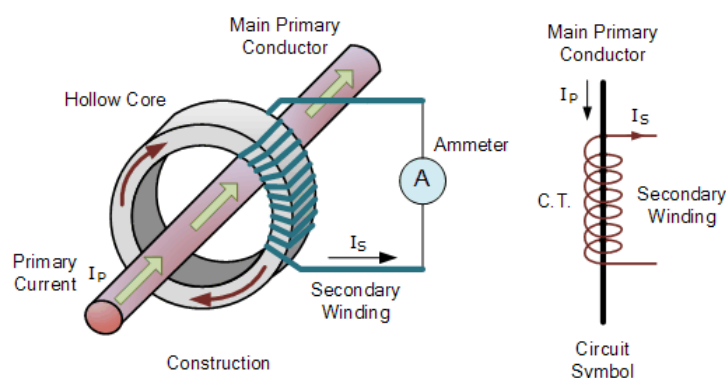


Figura 5.1 Esquema e símbolo de transformadores de intensidade

O funcionamento de um TI pode ser descrito com base nas leis físicas de Faraday-Neumann-Lenz também conhecidas pelas leis da indução eletromagnética. Uma corrente alternada, que circule no enrolamento primário, irá produzir no núcleo um campo magnético variável que por sua vez este irá induzir uma corrente alternada no enrolamento secundário pelo efeito de indução eletromagnética. Em situações ideais, isto é, não existam perdas eletromagnéticas por efeito de joule no circuito, os valores das intensidades das correntes dependerão apenas da relação entre o número de espiras presentes no circuito primário e no circuito secundário. Quantas mais espiras existirem no circuito primário, maior será a intensidade do campo magnético no núcleo o que irá provocar uma maior indução no enrolamento secundário provocando assim uma maior corrente à saída do mesmo. Assim, a relação ideal entre todas as variáveis num transformador pode-se expressar pela relação seguinte:

$$n = \frac{N_P}{N_S} = \frac{I_S}{I_P} = \frac{V_P}{V_S} \quad (5-1)$$

Onde n é a relação de transformação do transformador, N_P é o número de espiras no primário, N_S o número de espiras no secundário, I_P o valor de corrente que atravessa o enrolamento primário, I_S o valor da corrente que atravessa o enrolamento secundário, V_P a tensão presente aos terminais do primário e V_S a tensão aos terminais do secundário.

Manipulando esta expressão matematicamente, pode-se escrever do seguinte modo:

$$I_S = I_P \left(\frac{N_P}{N_S} \right) \quad (5-2)$$

Fórmula que nos indica que num transformador de corrente a transformação seria linear e apenas dependente da relação entre o número de espiras do primário e secundário. Na prática esta transformação não é assim tão linear podendo existir fenómenos de histerese, perdas eletromagnéticas, saturação do núcleo e perdas por efeito de Joule o que faz com que seja necessário proceder a uma escolha do transformador apropriado para cada caso.



Figura 5.2 – Transformador de intensidade ICES-EBM

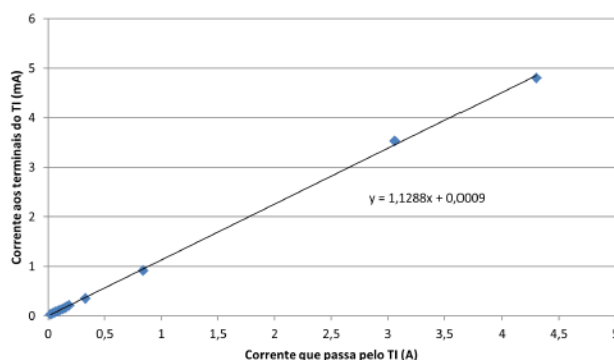


Figura 5.3 – Relação de transformação do ICES-EBM

Neste projeto foram usados transformadores de intensidade fabricados pela ICES-EBM (Figura 5.2) possuindo uma relação de transformação, obtida experimentalmente, de 1:885,9 (Figura 5.3). Estes TI's foram retirados de um antigo equipamento de medida pelo que não se possui as suas características exatas. Deste modo, pode-se usar a expressão seguinte:

$$I_{in} = 885,9 I_{out} \quad (5-3)$$

Onde I_{in} é a corrente que passa no enrolamento primário e I_{out} a corrente que passa no enrolamento secundário.

Como o objetivo é passar para o microcontrolador a informação acerca da corrente que está a ser medida e como a ADC deste apenas permite efetuar leituras em tensão, fez-se passar a corrente gerada no enrolamento secundário por uma resistência de modo a ser gerado uma tensão aos terminais desta. A resistência a utilizar vai depender da corrente máxima que pretendemos medir e dos limites de tensão suportados pelo ADC. Assim, o esquema será o seguinte da Figura 5.4:

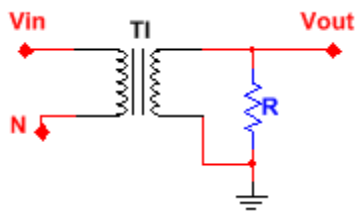


Figura 5.4 – Esquema de leitura em tensão de um TI

Adaptando a expressão anterior (5-3), fica-se com o seguinte:

$$I_{in} = 885,9 I_{out} = \frac{885,9 \cdot V_{out}}{R} \quad (5-4)$$

Podendo assim ser possível a leitura dos valores de corrente pelo microcontrolador através da tensão V_{out} .

5.1.2 Divisor de tensão

As tensões que se pretendem medir com o equipamento que se projeta excedem de longe os limites suportados pelas componentes eletrónicas a utilizar. Para colmatar este problema, teve-se que recorrer a redutores de tensão tendo-se escolhido, por razões económicas, os divisores potenciométricos como mostrados na Figura 5.5.

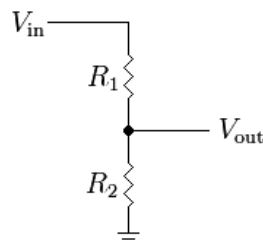


Figura 5.5 – Divisor de tensão resistivo

V_{in} é a tensão à entrada do circuito, R_1 e R_2 são as resistências do divisor resistivo e V_{out} a tensão à saída do circuito.

A expressão que descreve o comportamento do divisor resistivo é dada por:

$$V_{out} = \frac{R_2}{R_1 + R_2} \cdot V_{in} \quad (5-5)$$

Deste modo, o rácio $\frac{V_{out}}{V_{in}}$ será sempre entre o intervalo de 0 e 1, nunca podendo a tensão de saída ser superior à de entrada, isto em módulo.

5.1.3 Filtros

Após a aquisição do sinal é necessário proceder a uma filtragem do mesmo com o objetivo de serem atenuadas as frequências indesejadas e o efeito de *Aliasing* provocado pela amostragem.

Neste projeto, tal como foi calculado no secção 3.2 e 3.4, é necessário que sejam admitidas as frequências até 6400Hz. Um método simples de efetuar esta filtragem consiste em utilizar um filtro passa-baixo de primeira ordem RC, ou seja, uma resistência em série com um condensador, tal como é representado na Figura 5.6 [12].

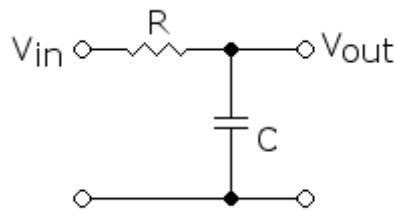


Figura 5.6 - Circuito passa-baixo RC

V_{in} é a tensão de entrada, V_{out} a tensão de saída, R a resistência e C o valor do condensador.

A expressão que equaciona estes componentes com a frequência de corte, f_c , é:

$$f_c = \frac{1}{2\pi RC} \quad (5-6)$$

Tendo como objetivo usar este filtro após os divisores resistivos e transformadores de intensidade, onde já se encontram resistências que podem constituir este R, Figura 5.7, pode-se manipular (5-6) de modo a poder-se calcular o valor do condensador necessário para a aplicação.

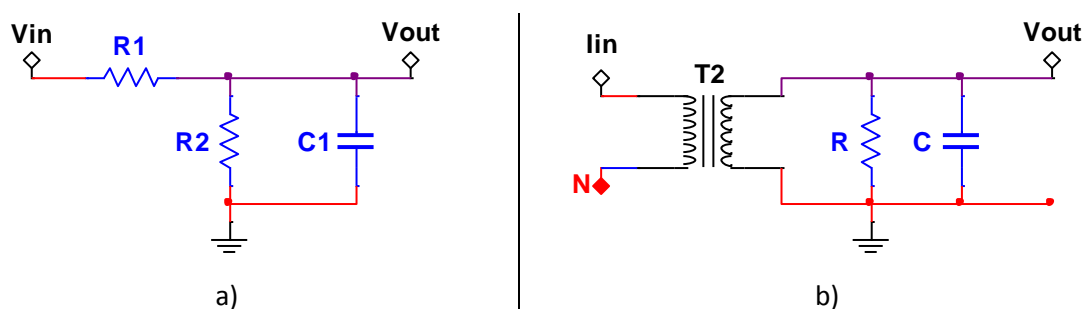


Figura 5.7 – Filtros aplicados: a) Divisores Resistivos; b) Transformadores de Intensidade

Assim, o valor do condensador necessário para que se tenha uma frequência de corte f_c é dado por:

$$C = \frac{1}{2\pi R f_c} \quad (5-7)$$

Para o caso dos transformadores de intensidade, como apresentam só uma resistência visto do condensador (Figura 5.7-b, Resistência R), a expressão será igual a (5-7) onde o R é a resistência colocada aos terminais do transformador de intensidade.

Para o caso dos divisores resistivos, as duas resistências que o compõem, R1 e R2, são vistas em paralelo do lado do condensador. Isto significa que a expressão passa a ser a seguinte (5-8):

$$C = \frac{1}{2\pi(R_1 \parallel R_2)f_c} \quad (5-8)$$

Deste modo é possível projetar os filtros adequados para que sejam filtradas as frequências indesejadas.

5.1.4 Multiplexagem de sinais

Designa-se por multiplexagem o processo de concentrar múltiplos sinais, provenientes de canais distintos, num único canal de transmissão. O dispositivo que efetua este processo de multiplexagem é chamado de multiplexador (Em inglês, multiplexer).

Existem diferentes tipos de multiplexagem de sinais, como por exemplo, multiplexagem na frequência (FDM – Frequency Division Multiplexing) e multiplexagem no tempo (TDM – Time Division Multiplexing). Para este projeto houve a necessidade de fazer-se multiplexagem no tempo e por esta razão apenas é esta que é aqui apresentada.

A multiplexagem no tempo consiste em transmitir diferentes sinais em intervalos de tempo bem definidos. A cada sinal é-lhe atribuído regularmente um tempo específico para ser transmitido. Quando esse tempo acaba, o multiplexer seleciona um outro sinal de uma das suas entradas para este poder ser propagado durante o intervalo de tempo que se segue. A nível de exemplo, na Figura 5.8 [13] é representado um esquemático de um multiplexer de duas entradas.

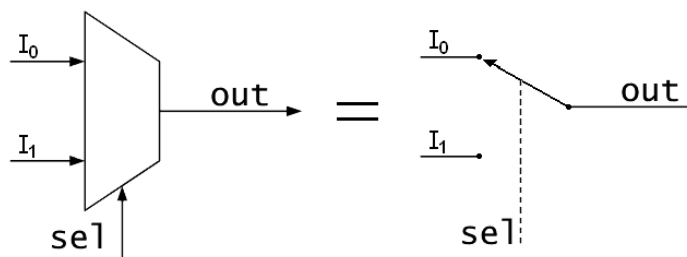


Figura 5.8 – Esquemático de um multiplexer de duas entradas

\$I_0\$ e \$I_1\$ são entradas dos sinais do multiplexer e 'sel' é a entrada de seleção de canal.

Fazendo variar a entrada de seleção, o multiplexer coloca na saída 'out' o sinal correspondente a uma das entradas \$I_0\$ ou \$I_1\$.

5.1.5 Buffer de tensão

Um buffer de tensão, ou também designado por seguidor de tensão, é um amplificador de ganho unitário com uma impedância de entrada elevada e uma impedância de saída baixa. Tem como particularidade colocar à sua saída a mesma tensão que é apresentada à entrada, ou seja, não oferece nenhum ganho em tensão. Por outro lado, tem a vantagem de proporcionar mais corrente o que implica que pode fornecer mais potência à carga que estiver a alimentar. Buffers de tensão são também usados para desacoplamento de circuitos.

O esquema do buffer de tensão está representado na Figura 5.9 [14]:

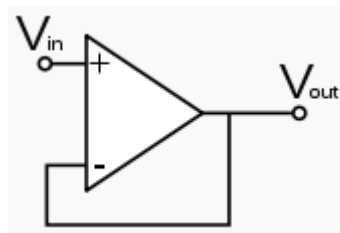


Figura 5.9 – Esquema de um buffer de tensão

O Buffer de tensão é concretizado recorrendo a um Op-amp (Operational Amplifier) ligado em malha fechada com feedback negativo (Figura 5.9).

5.1.6 Circuito somador em tensão não inversor

Devido às diferentes gamas de variação de sinais com que os diferentes componentes eletrónicos podem operar, torna-se necessário fazer alterações das amplitudes máximas de variação permitidas destes sinais de modo que possam ser lidos e tratados por estes componentes.

No presente projeto foi necessário fazer mudanças da amplitude dos sinais provenientes das unidades de aquisição de modo que estes fossem lidos pelas ADC's. Os sinais fornecidos pelas unidades de aquisição variam entre valores negativos a valores positivos. Torna-se necessário que os sinais apenas possuam excursões positivas para que as ADC's consigam ler corretamente. O circuito representado na Figura 5.10 permite resolver o problema. Trata-se de um circuito somador em tensão não inversor onde v_i representa a tensão de entrada, V_1 uma tensão de referência, v_o a tensão de saída e $R_1 \dots R_4$ as resistências que determinam as características de ganho do circuito.

Para o circuito da Figura 5.10 a tensão de saída é dada pela equação (5-9) [15]:

$$v_o = \frac{R_4}{R_3 + R_4} \left(1 + \frac{R_2}{R_1} \right) v_i + \frac{R_3}{R_3 + R_4} \left(1 + \frac{R_2}{R_1} \right) V_1 \quad (5-9)$$

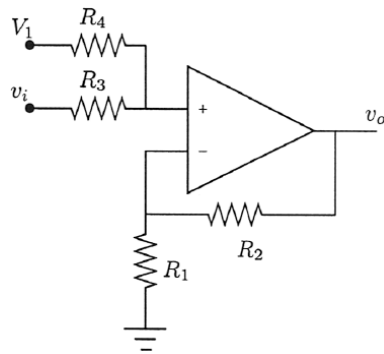


Figura 5.10 – Circuito somador em tensão não inversor

Como no presente caso não se pretende ter ganho em tensão mas sim apenas uma transposição da gama de valores positivos e negativos para apenas positivos, deve impor-se as seguintes relações entre resistências:

$$R_3 = R_4 \quad (5-10)$$

$$R_1 = R_2 \quad (5-11)$$

Uma maneira prática e simples de se obter o mesmo resultado é colocar resistências todas iguais, ou seja:

$$R_1 = R_2 = R_3 = R_4 \quad (5-12)$$

Aplicando à expressão (5-9) obtém-se o seguinte:

$$v_o = v_i + V_1 \quad (5-13)$$

Ou seja, à saída do circuito aparece o sinal de entrada somado de uma tensão positiva V_1 que é a tensão de referencia dada por um regulador e que possui o valor necessário de modo que a excursão do sinal passe a ser sempre em valores positivos.

5.1.7 Regulador de tensão

Um regulador de tensão é um dispositivo que tem como principal característica fornecer uma tensão contínua estabilizada de elevada precisão mesmo quando existem variações nas alimentações do circuito ou na corrente de saída do estabilizador. São dispositivos vulgares e de baixo custo.

Neste projeto foi necessário criar uma tensão estabilizada o mais precisa possível para se fazer a soma de tensão no circuito somador falado na secção 5.1.6. Para isso recorreu-se ao regulador *TL432A* visto que já existia em armazém e cumpria com os requisitos que se pretendia.

Este regulador comporta-se como um diodo Zener onde é necessário existir uma tensão de polarização inversa acima de um certo valor para que passe a ser percorrido por uma corrente no sentido inverso, ou seja no sentido do cátodo para o ânodo mantendo o diodo uma tensão praticamente constante aos seus terminais. O diodo *TL432A* garante entre os seus terminais uma

tensão de referência precisa e estável de 2,5V. Atendendo a que o valor da tensão de referência pretendida, 2,5V, coincidia com o valor da tensão própria do Zener utilizou-se a configuração representada na Figura 5.11 [16].

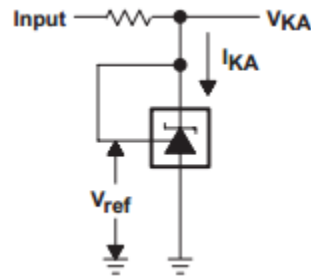


Figura 5.11 – Regulador de tensão

V_{ref} é o valor da tensão de referência definido pelo fabricante e que aparece aos seus terminais quando o regulador é polarizado inversamente. Nesta configuração apenas é necessário calcular o valor da resistência de modo a que exista uma corrente mínima de polarização I_{KA} a passar pelo regulador, pois $V_{KA} = V_{ref}$. Assim, temos:

$$I_{KA} = \frac{Input - V_{KA}}{R} \quad (5-14)$$

No caso da tensão de referência pretendida não coincidir com a definida pelo fabricante e se pretender uma referência de valor mais baixo, o circuito que se utilizou para o efeito está apresentado na Figura 5.12.

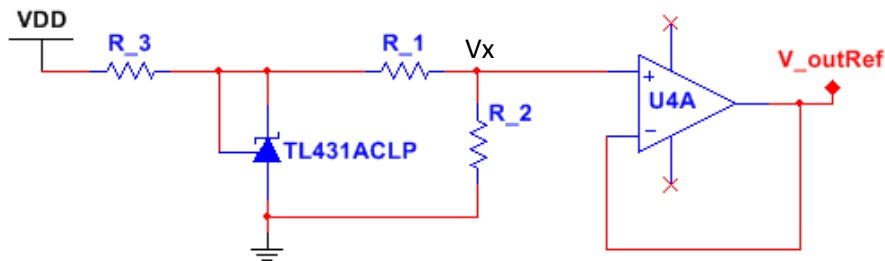


Figura 5.12 – Circuito regulador de tensão para tensões inferiores à de referência

Neste circuito é necessário projetar-se a resistência R_3 de modo a fornecer ao resto do circuito a corrente de polarização do regulador TL431 e a corrente que atravessa o divisor resistivo formado pelas resistências R_1 e R_2 . Na saída do divisor resistivo, V_x , apresentar-se-á a tensão de referência necessária. Esta tensão de referência é transmitida para a saída do Op-Amp, V_{outRef} , pois este está numa configuração seguidor de tensão fazendo o desacoplamento do circuito, tal como visto na secção 5.1.5.

Tendo isto, fica-se então com a tensão de saída do circuito, V_{outRef} , descrita pela equação:

$$V_{outRef} = V_x = \frac{R_2}{R_1 + R_2} \cdot V_{ref} \quad (5-15)$$

Sendo a corrente consumida pelo divisor resistivo dada pela equação:

$$I_{DR} = \frac{V_{ref}}{R_1 + R_2} \quad (5-16)$$

Então a corrente total que este circuito irá necessitar é dada pela equação:

$$I_T = I_{DR} + I_{KA} \quad (5-17)$$

Deste modo, já se pode calcular o valor da resistência R_3 . Este valor é dado pela seguinte expressão:

$$R_3 = \frac{V_{DD} - V_{ref}}{I_T} \quad (5-18)$$

Assim, projetando todos estes valores ter-se-á à saída do circuito uma tensão de referência fixa.

5.1.8 Referência à massa

A referência à massa constitui um aspeto importante para o desenvolvimento de equipamentos elétricos pois é por esta referência que todo o circuito se vai referenciar. Tendo o equipamento vários níveis de tensão para diferentes componentes eletrónicos, teve-se de projetar o seguinte circuito:

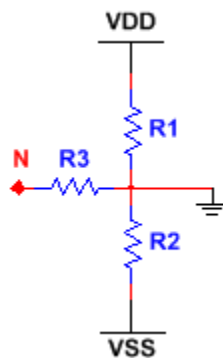


Figura 5.13 – Referência à massa

Desta maneira, tanto as alimentações VDD, VSS e Neutro (N) da rede elétrica ficam estabilizadas em relação à massa.

É necessário dimensionar as resistências de modo a que a potência nelas consumida não exceda os valores permitidos pelas características do seu fabricante. Neste caso, a potência máxima das resistências que se pretende usar é de $\frac{1}{3}$ Watt. Assim, pela fórmula da potência temos:

$$P = V \cdot I = R \cdot I^2 = \frac{V^2}{R} \quad (5-19)$$

Para VDD e VSS, de 5V e -5V, respetivamente, usando resistências de 2.2KΩ a potência dissipada será então de:

$$P = \frac{V^2}{R} = \frac{5^2}{2200} \approx 0,011 W < \frac{1}{3} W \quad (5-20)$$

O que é um valor aceitável para o efeito pretendido.

5.2 Construção do primeiro protótipo do projeto

Nesta secção apresenta-se a tentativa inicial de solução deste projeto. Por imposição da gerência da firma SELMATRON teve-se que utilizar como unidade de processamento um Kit *Stellaris LM3S8962 da Luminary Micro* e como pacote de *software* o LabVIEW da *National Instruments*, isto conjuntamente com hardware adicional destinado a fazer a aquisição do sinal. Primeiramente, descreve-se os aspetos mais importantes do Kit de desenvolvimento, de seguida o *hardware* criado para o efeito de aquisição dos vários sinais que se pretende analisar, depois o *software* implementado no *LabVIEW* e por fim os resultados desta primeira fase.

Na Figura 5.14 pode-se observar o diagrama de blocos geral.

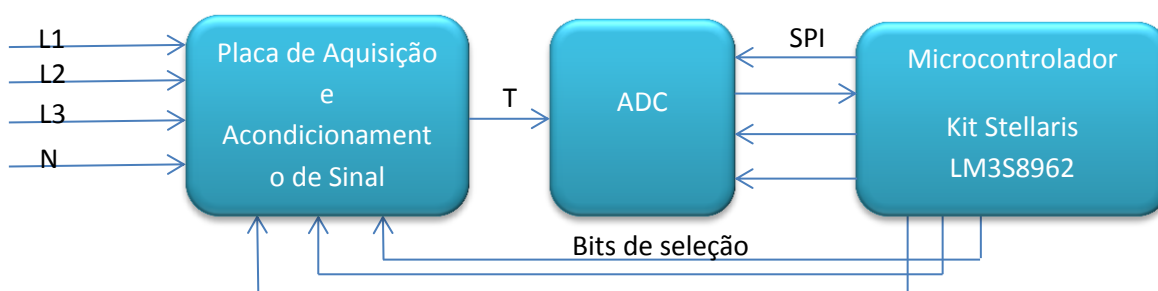


Figura 5.14 – Esquema geral da solução multiplexada

L1, L2, L3 representam as 3 fases da tensão e corrente da rede elétrica, N o neutro, T o sinal do canal selecionado pelos bits de seleção e SPI a comunicação entre a ADC e o Microcontrolador.

5.2.1 Kit de desenvolvimento *Stellaris LM3S8962*

O Kit de desenvolvimento *Stellaris LM3S8962* é apresentado na Figura 5.15. Este kit de desenvolvimento é baseado no microcontrolador *LM3S8962 Cortex-M3* da *Texas Instruments* possuindo alguns periféricos que ajudam a sua interação com outros dispositivos e o utilizador.

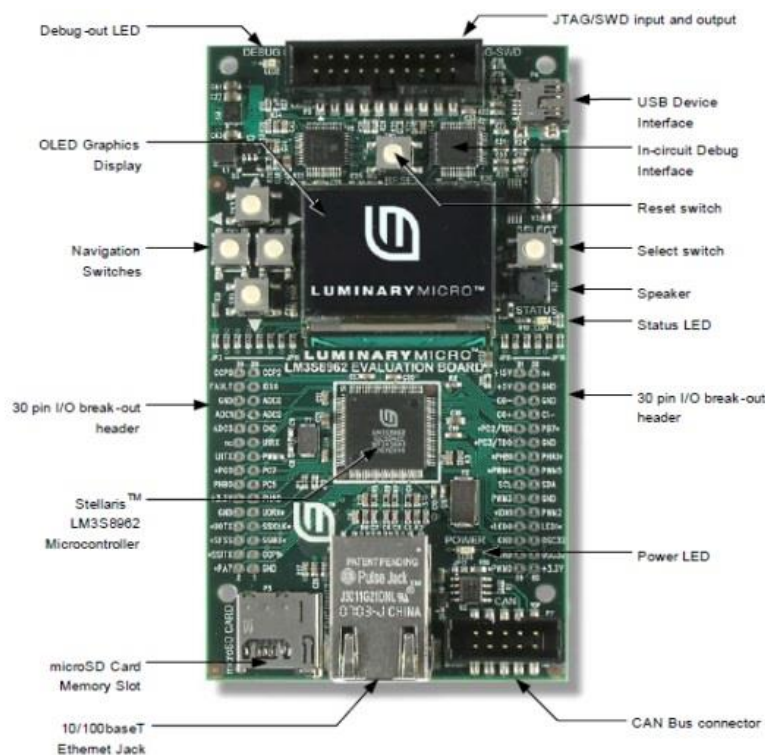


Figura 5.15 – Kit Stellaris LM3S8962

Os seus periféricos principais são os seguintes:

- Microcontrolador LM3S8962 Cortex M3 - core
- Ecrã OLED gráfico (128 x 96 pixels)
- Controlador Ethernet 10/100;
- Slot cartões MicroSD;
- Interface USB para alimentação de energia da placa e para debug
- 5 Botões de interface e um de Reset
- Standard ARM 20-pin JTAG debug
- Todos os pinos do microcontrolador encontram-se disponíveis externamente para ligação com hardware adicional externo

Quanto às especificações do microcontrolador, verifica-se que tem características necessárias para o cumprimento dos objetivos deste projeto. A sua velocidade de processamento e capacidade de memória RAM são características importantes para o correto desempenho na análise de energia que se pretende avaliar. Salientando as características mais importantes deste microcontrolador temos então as seguintes:

- Funcionamento a 50MHz
- 64KB de memória SRAM
- 256KB de memória Flash
- 10/100 Ethernet Controller
- Síncrona interface série (SSI)

- 4 canais de 10 bit de ADC de 500KSPS
- Até 42 GPIOs
- 5V máximo em alguns pinos de entrada

5.2.2 Placa de aquisição (primeiro protótipo)

Baseado no que foi apresentado na secção 5.1, elaborou-se a placa de aquisição do primeiro protótipo. O esquemático da placa, tendo em vista a sua construção em circuito impresso, está representado na Figura 5.16 tendo sido desenhado com recurso ao software *MultiSim*. Está dividido em várias secções, numeradas de A a I, as quais serão em seguida explicadas em subcapítulos.

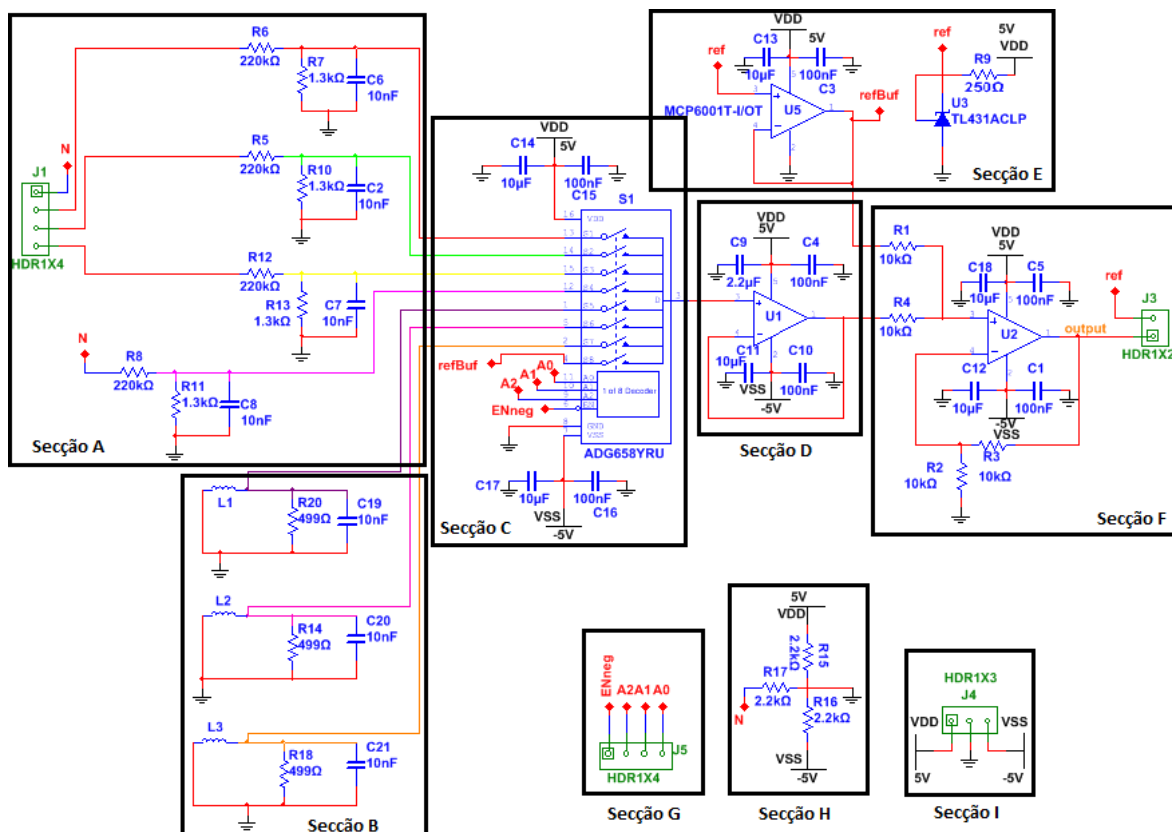


Figura 5.16 – Esquemático da placa de aquisição

Tendo em conta que os sinais provenientes dos sete canais de entrada da placa passariam pelos mesmos processos até chegarem à entrada da ADC, optou-se por abordar uma solução de multiplexagem dos mesmos utilizando um multiplexador de 8:1 para se reduzir os custos de *hardware*.

Como o objetivo desta placa de aquisição é acondicionar os sinais de entrada de modo a poderem ser lidos pela ADC, apresenta-se em seguida na secção 5.2.2.1 as características da ADC utilizada

assim como o esquemático da placa em que a ADC foi montada. A razão de colocar a ADC numa placa separada deve-se ao facto de possibilitar uma maior flexibilidade para os testes do equipamento projetado.

As tensões de alimentação da placa de aquisição são introduzidas através de um ligador (secção I, Figura 5.16) onde VDD e VSS correspondem a 5V e -5V, respetivamente. Junto aos pinos de entrada de alimentação de todos os componentes não passivos foram colocados condensadores de $10\mu F$ e $100nF$ para atenuar efeitos oscilatórios de altas frequências e baixas frequências, respetivamente, que possam ocorrer.

Todas as tensões estão interligadas às tensões de referência do modo como se indica na secção H, Figura 5.16 como explicado anteriormente na secção 5.1.8.

5.2.2.1 ADC (primeiro protótipo)

Atendendo a que o Kit Stellaris LM3S8962 apenas possui quatro unidades ADC e o projeto implica o uso de 7 unidades (3 para corrente por fase, 3 para tensões por fase e uma para tensão de neutro) e ainda a que os 10 bits das ADC's do Kit apenas permitiam uma precisão de 0,83V ($2 \cdot 425 / 1024 = 0,83V$) optou-se por recorrer a uma unidade ADC de 12 bits que permite uma maior precisão de 0,21V ($2 \cdot 425 / 4096 = 0,21V$).

A ADC escolhida para a leitura dos canais de entrada foi a *AD7928* da *Analog Devices*. Esta ADC possui 12 bits de quantização o que garante 4096 níveis de quantização ($2^{12} = 4096$ Níveis). Uma vez que a ADC possui a capacidade de suportar valores entre 0V e 5V nos seus pinos de entrada, então aplicando a equação (3-5), verifica-se que os sinais à sua saída apresentam uma resolução de quantização de:

$$R = \frac{\Delta V}{2^n} = \frac{5 - 0}{2^{12}} \approx 1,22 \cdot 10^{-3} \left[\frac{V}{\text{Nível}} \right] \quad (5-21)$$

Logo, o erro máximo cometido pela quantização será metade da resolução pois na pior hipótese o valor aplicado na entrada estará exatamente entre dois níveis, ou seja, um valor de erro de $0,61 \cdot 10^{-3}$ Volt. Em termos de projeto verifica-se que o erro pode ser desprezável pois corresponde apenas a menos de 0,0122%.

Em termos de codificação do sinal lido a ADC foi programada no modo "*Straight Binary Transfer Characteristic*" tal como mostra a Figura 5.17 retirada do próprio *datasheet* [17].

Toda a comunicação entre a ADC e o microcontrolador é feita por interface SPI já estudada na secção 3.6.

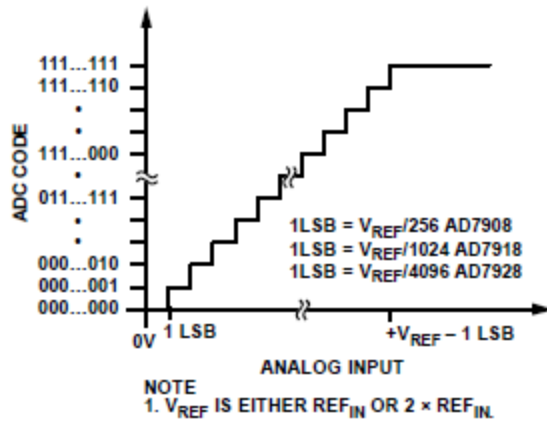


Figura 5.17 – Straight binary transfer characteristic

O modo de transferência de informação da ADC mostra-se na Figura 5.18.

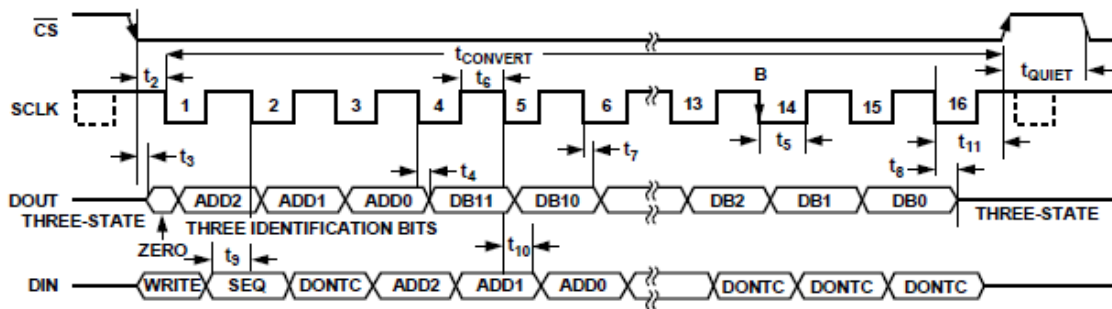


Figura 5.18 – Modo de transferência de informação da ADC AD7928

Onde CS\ é o Chip Select, SCLK o Serial Clock, DOUT os dados de saída da ADC e DIN os dados de entrada da ADC.

Na Figura 5.19 é mostrado o esquemático da placa da ADC criado recorrendo ao *software Multisim* tendo em vista a sua construção em circuito impresso.

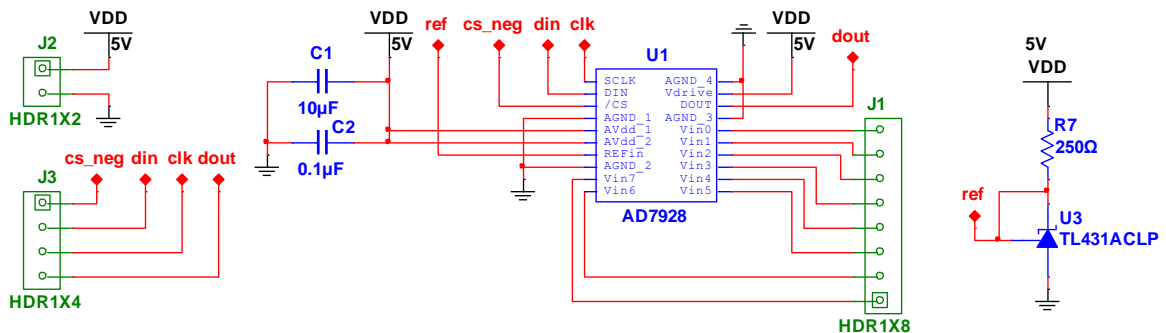


Figura 5.19 – Esquemático da placa da ADC

As ligações que fazem a comunicação entre a [ADC](#) e o microcontrolador são feitas através do Header J3 (ver Figura 5.19) onde se encontram todos os sinais necessários para que ocorra a

comunicação via SPI. Através do Header J2 da mesma figura é feita a alimentação de toda a placa da ADC. As entradas analógicas da ADC são feitas através do Header J1.

Para efeitos de configurações da ADC foi necessário fornecer uma tensão de referência de 2,5V, aqui descrito por “ref” e obtida por polarização inversa do regulador *TL432A* já falado na secção 5.1.7.

Fazendo a polarização do regulador com uma corrente de 10mA para se ter a tensão de referência de 2,5V em “ref”, pode-se manipular a expressão (5-14) e calcular o valor da resistência necessária para polarizar o regulador tendo-se obtido:

$$R_7 = \frac{Input - V_{KA}}{I_{KA}} = \frac{5 - 2,5}{10mA} = 250\Omega \quad (5-22)$$

Onde *Input* representa a tensão VDD, V_{KA} a tensão “ref” e I_{KA} a corrente de polarização.

Assim, esta placa irá receber o sinal proveniente da placa de aquisição da Figura 5.16, fazer a leitura digital do sinal pela ADC e enviar para o microcontrolador a informação lida.

5.2.2.2 Divisores resistivos com filtragem (Secção A)

A Figura 5.20 representa o diagrama da secção A da placa de aquisição e tem como objetivo efetuar o decaimento dos valores lidos em tensão da rede elétrica para valores com uma amplitude de excursão que a [ADC](#) pode ler.

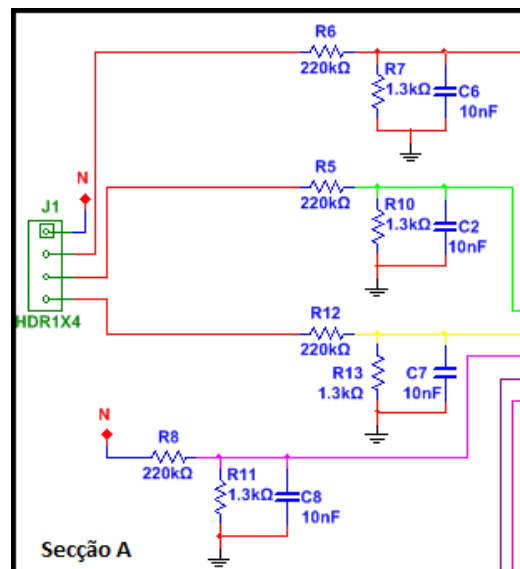


Figura 5.20 – Secção A da placa de aquisição

Adicionalmente, tal como foi explicado na secção 5.1.3, efetua-se também a filtragem necessária para serem atenuadas as frequências indesejadas e também atenuar o efeito de *Aliasing*.

A ADC aqui usada tem como valores de entrada possíveis compreendidos entre 0V a 5V. Deste modo, houve necessidade de projetar uma unidade capaz de baixar os valores de tensão que existem na rede para valores compreendidos nesta diferença de amplitudes. A rede elétrica em média possui uma tensão alternada de $230V_{rms}$, ou seja, usando a expressão (3-11), uma tensão de pico:

$$V_p = V_{rms} \cdot \sqrt{2} = 230 \cdot \sqrt{2} \approx 325V \quad (5-23)$$

Atendendo às normas EN 50160 que dizem que os valores de RMS não devem ter variações de $\pm 10\%$ durante 95% da semana, optou-se por dar uma margem aproximadamente de 30% do valor de pico da onda, ou seja:

$$V_{max} = V_p \cdot 1,3 = 422,5V \approx 425V \quad (5-24)$$

Deste modo, esta secção A da placa de aquisição deverá converter tensões que não excedam o intervalo de $[-425V ; 425V]$ para o intervalo de $[-2,5V ; 2,5V]$.

Usando a fórmula do divisor resistivo (5-5) podemos verificar que o rácio entra as tensões de entrada e de saída é dado pela equação:

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} \quad (5-25)$$

Usando os valores de $V_{in} = 425V$ e $V_{out} = 2,5V$, obtém-se o seguinte valor:

$$\frac{V_{out}}{V_{in}} = \frac{2,5}{425} \approx 0,00588 \text{ [V/V]} \quad (5-26)$$

Fazendo a atribuição mais próxima para rácio de $\frac{V_{out}}{V_{in}}$ das resistências da série E24 [18], pode-se chegar à conclusão que o valor das resistências que melhor representam esta relação são as seguintes:

$$R_1 = 220K\Omega \quad (5-27)$$

$$R_2 = 1,3K\Omega \quad (5-28)$$

Utilizando estas resistências a relação $\frac{V_{out}}{V_{in}}$ será:

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} = \frac{1,3K}{1,3K + 220K} \approx 0,00587 \text{ [V/V]} \quad (5-29)$$

O que é um valor muito próximo do pretendido.

Em termos de potência dissipada nas resistências, podemos observar que a potência máxima consumida nas resistências é dada por:

$$P_{R1} = \frac{V^2}{R_1} = \frac{(425 - 2,5)^2}{220K} \approx 0.811 \text{ W} \quad (5-30)$$

$$P_{R2} = \frac{V^2}{R_2} = \frac{2,5^2}{1,3K} \approx 0.005 \text{ W} \quad (5-31)$$

Daqui conclui-se que a resistência R_1 excede os valores de potência normais suportados pelas resistências comuns (1/3 W) sendo necessário o uso de resistências de potência de pelo menos 1W.

Tendo já os valores das resistências calculados, pode agora fazer-se o cálculo do valor do condensador a ser utilizado para a filtragem. Como estas duas resistências do divisor resistivo estão em paralelo quando vistas do condensador, então pode-se usar a expressão (5-8) para o cálculo do valor do condensador, tendo como a frequência de corte o valor de 6400Hz como visto anteriormente no capítulo 3.2.

$$C = \frac{1}{2\pi(R_1 \parallel R_2)f_c} = \frac{1}{2\pi \cdot (220K \parallel 1,3K) \cdot 6400} = 19,1 \text{ nF} \quad (5-32)$$

Não havendo este valor em armazém, optou-se por se colocar o condensador de valor existente mais próximo inferior, ou seja, um condensador com o valor de 10nF . Fazendo novamente as contas este condensador irá provocar uma frequência de corte de:

$$f_c = \frac{1}{2\pi(R_1 \parallel R_2)C} = \frac{1}{2\pi \cdot (220K\Omega \parallel 1,3K\Omega) \cdot (10\text{nF})} = 12,2 \text{ KHz} \quad (5-33)$$

O que é admissível.

Fazendo a simulação desta parte do circuito através do *software MultiSim*, verificou-se que os cálculos efetuados correspondiam aos simulados, tal como mostra a Figura 5.21.

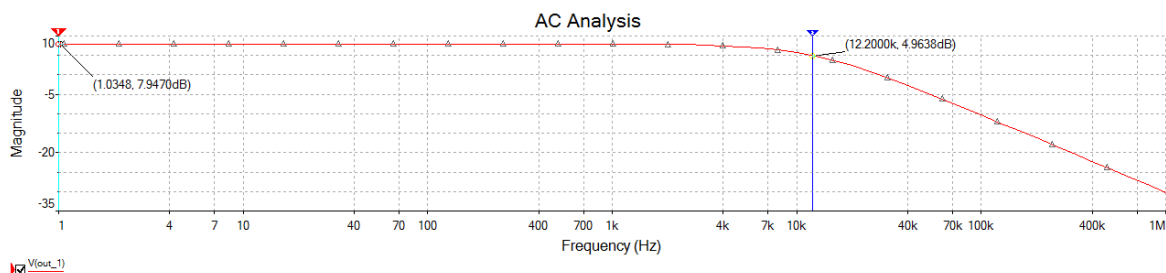


Figura 5.21 – Resposta em frequência do filtro com os divisores resistivos

No eixo vertical a magnitude do sinal é representada em decibel.

Neste gráfico, verifica-se que para a frequência de 12,2KHz a magnitude cai 3dB, o que representa a frequências de corte do filtro.

5.2.2.3 Transformadores de intensidade com filtragem (Secção B)

Na Figura 5.22, que corresponde à secção B da placa de aquisição, encontra-se o esquema do circuito que tem como o objetivo ler valores de correntes obtidas pelos transformadores de

intensidade e simultaneamente realizar também a filtragem necessária para serem atenuadas as frequências indesejadas e atenuar o efeito de Aliasing.

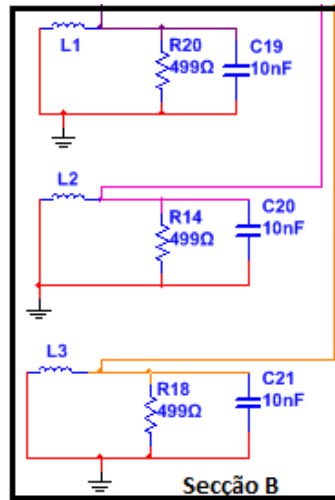


Figura 5.22 – Seção B da placa de aquisição

Visto que este esquema já foi desenvolvido com o objetivo de ser construído em circuito impresso, os terminais da L1, L2 e L3 são os pontos em que ligam os terminais do enrolamento secundário dos transformadores de Intensidade. Deste modo será mais fácil o processo de montagem dos transformadores de intensidade.

Tal como foi dito no capítulo 5.1.1, no projeto são usados transformadores de intensidade com uma relação de transformação de 1:885,9. Não se conhecendo exatamente as características dos transformadores admitiu-se apenas para efeitos de teste uma corrente máxima de 5A. Assim, usando a fórmula (5-4) pode-se verificar qual o valor ideal para a resistência de modo a ter $V_{out} = 2,5V$ quando se faz passar o máximo de corrente na entrada, ou seja, $I_{in} = 5A$. Então:

$$R = \frac{885,9 \cdot V_{out}}{I_{in}} = \frac{885,9 \cdot 2,5}{5} = 442,95 \Omega \quad (5-34)$$

Sendo as resistências de 499Ω aquelas que mais se aproximavam deste valor em laboratório, optou-se por usar estas para efeitos de teste. Deste modo, a corrente máxima que o equipamento poderia ler será então:

$$I_{in} = \frac{885,9 \cdot V_{out}}{R} = \frac{885,9 \cdot 2,5}{499} = 4,438 A \quad (5-35)$$

Usando a expressão (5-3), corresponde a uma corrente à no secundário do transformador de:

$$I_{out} = \frac{I_{in}}{885,9} = \frac{4,438}{885,9} \approx 5 mA \quad (5-36)$$

Fazendo o cálculo da potência máxima consumida pela resistência temos:

$$P_{R1} = \frac{V^2}{R} = \frac{2,5^2}{499} \approx 0.0125 W \quad (5-37)$$

O que é uma potência que resistências SMD suportam (1/8 W). Por razões práticas de criação da placa optou-se por usar-se este tipo de resistências para esta parte do projeto.

Fazendo agora os cálculos para o filtro passa baixo, tem-se apenas a resistência de 499Ω quando vista pelo condensador. Assim, pode-se usar a equação (5-7) para se calcular o valor do condensador.

$$C = \frac{1}{2\pi R f_c} = \frac{1}{2\pi \cdot 499 \cdot 6400} = 49,8nF \quad (5-38)$$

O valor do condensador existente em armazém mais adequado foi de $10nF$ obtendo-se assim uma frequência de corte de:

$$f_c = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 499 \cdot 10nF} = 31,8 KHz \quad (5-39)$$

Valor considerado admissível.

Fazendo a simulação desta parte do circuito através do *software MultiSim*, verificou-se que os cálculos efetuados correspondiam aos simulados, tal como mostra a Figura 5.23.

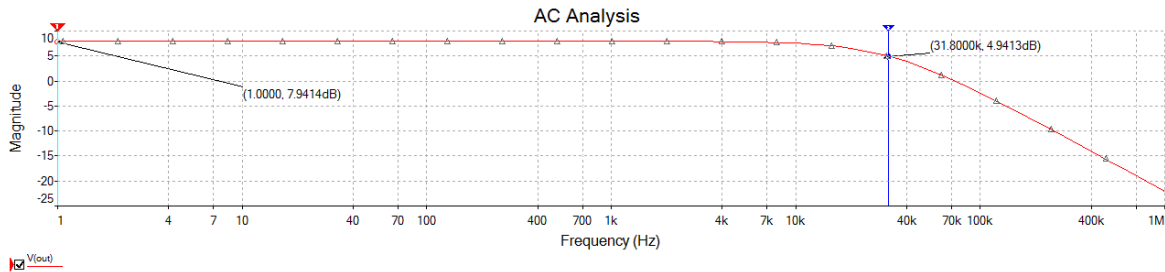


Figura 5.23 – Resposta em frequência do filtro após TI's

No eixo vertical a magnitude do sinal é representada em decibel.

Neste gráfico, verifica-se que para a frequência de $31.8KHz$ a magnitude cai $3dB$, o que representa a frequências de corte.

5.2.2.4 Multiplexação dos canais (Secção C e G)

O objetivo das secções C e G, representadas na Figura 5.24 a) e b) da placa de aquisição, é fazer a multiplexação no tempo dos sinais provenientes dos diferentes canais de entrada (sete – três correntes e quatro tensões) após a sua filtragem.

O motivo por que se optou pela introdução de multiplexação foi para reduzir as componentes de hardware uma vez que os sinais de entrada iriam todos ter o mesmo acondicionamento passando pelo mesmo tipo de hardware até à entrada da ADC.

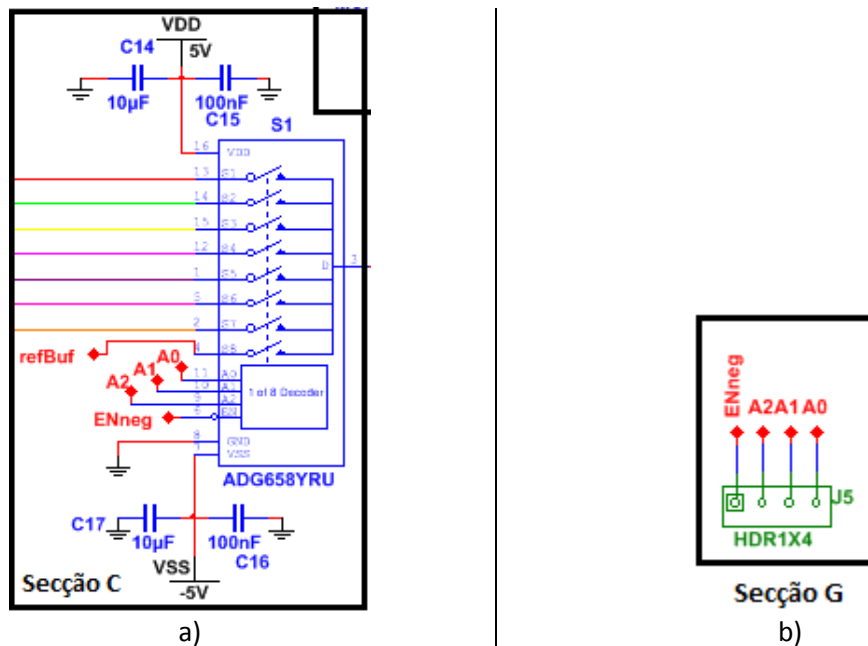


Figura 5.24 – Secções da placa de aquisição: a) Secção C; b) Secção G

Neste projeto usou-se o *multiplexer* ADG658 da empresa *Analog Devices*. Este *multiplexer* faz a multiplexação no tempo de 8 sinais de entrada para uma única saída (8:1). Tem como características principais:

- Ser um multiplexer Rail-to-Rail (permite colocar à sua saída toda a gama de valores possíveis das entradas analógicas sem que ocorra saturação)
- As tensões suportadas pelas entradas analógicas abrangem as tensões máximas e mínimas presentes à saída dos filtros [-2,5V; 2,5V]
- Tem resistência de saída baixa (45Ω)
- Ativação do canal pretendido faz-se através de 4 pinos de entrada digitais (3 de seleção e 1 de ativação do multiplexer)
- Velocidade de transição de canais típica de 80ns, muito superior à necessária

Quando o multiplexer está ativo, através do pino de ativação (ENneg), Figura 5.24-a), a seleção do canal pretendido é feita por 3 pinos de entrada digital. Estas entradas serão provenientes do microcontrolador que selecionará o canal que pretende ativar através do conector da secção G, Figura 5.24-b). Este ligador é o responsável pela ligação física entre o microcontrolador e o multiplexer.

5.2.2.5 Buffer de desacoplamento (Secção D)

A secção D, representada na Figura 5.25, foi projetada apenas para fazer o desacoplamento entre os sinais passados pelo multiplexer e o circuito somador que se trata a seguir. Trata-se de um

circuito com alta impedância de entrada e baixa de saída possuindo os sinais de entrada e saída a mesma amplitude em Volt.

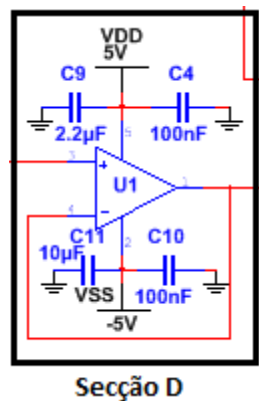


Figura 5.25 – Secção D da placa de aquisição - desacoplador

O Op-amp escolhido para o efeito foi o *MCP6001* da empresa *Microchip* que tem como características as seguintes:

- Ganho constante para largura de banda até 1MHz
- Rail-to-Rail
- Tensões de entrada e de saída de V_{SS} a V_{DD}

Deste modo, este buffer garante o desacoplamento e a passagem de sinal por completo sem que haja atenuações nas componentes harmónicas que se quer analisar.

5.2.2.6 Circuito somador de tensão e regulador de tensão com buffer (secção F e E)

Pretendendo-se transferir o sinal fornecido pelo multiplexer (variando entre [-2,5V a 2,5V]) para os valores que a ADC pode ler [0V a 5V], torna-se necessário adicionar uma tensão de 2,5V ao sinal.

Para tal usou-se o circuito somador de tensão não inversor, apresentado na secção F da Figura 5.26-a) e secção 5.1.6, com um regulador de tensão com buffer, apresentado na secção E Figura 5.26-b) e secção 5.1.7.

Usando o regulador *TL432A*, na configuração mostrada na Figura 5.11, pretende-se fazer a polarização inversa para que este apresente aos seus terminais de saída a tensão de referência de 2,5V.

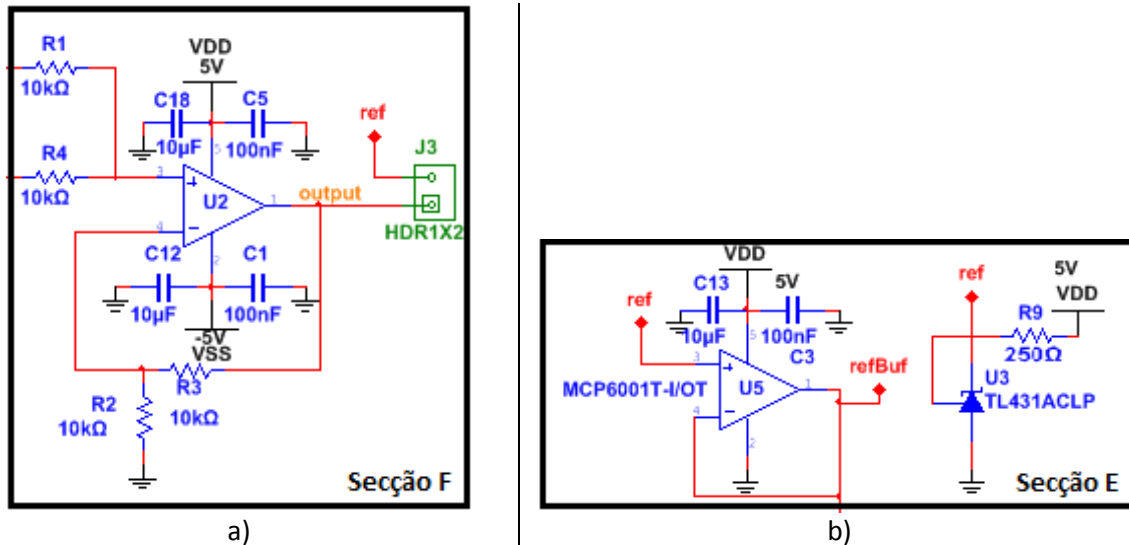


Figura 5.26 – Secções da placa de aquisição: a) Secção F; b) Secção E

Fazendo a polarização do regulador com uma corrente de 10mA, recomendada pelo datasheet, tendo o valor de VDD de 5V e uma tensão de saída de 2,5V, pode-se manipular a expressão (5-14) e calcular o valor da resistência necessária para polarizar o regulador. Assim, o valor da resistência necessária a colocar para a sua polarização será:

$$R = \frac{Input - V_{KA}}{I_{KA}} = \frac{5 - 2,5}{10mA} = 250\Omega \quad (5-40)$$

Para desacoplar o regulador de tensão do circuito somador usou-se um buffer de tensão à saída do regulador. Deste modo não haverá perturbações na tensão de saída do regulador e esta tensão será passada através do buffer para o circuito somador.

Tal como foi visto na secção 5.1.6, quando os valores das resistências do circuito somador são do mesmo valor, equação (5-12), este circuito não apresenta nenhum ganho em tensão apenas apresentando à sua saída a soma em tensão das suas entradas. Foram escolhidas resistências de 10KΩ por simplicidade.

Deste modo, à saída do circuito somador os valores da tensão apenas poderão variar entre os valores pretendidos de tensão, isto é, de [0V a 5V].

A ligação física do sinal de saída desta placa para a ADC externa é feita por uso do Header HDR1X2 da Figura 5.26-a).

5.2.3 Software LabVIEW

Por imposição da gerência da empresa SELMATRON proprietária do projeto uma vez que foi quem o submeteu ao QREN, teve que se realizar toda a programação do Kit de desenvolvimento *Stellaris LM3S8962* apresentado na secção 5.2.1 recorrendo ao *software* LabVIEW.

O LabVIEW é uma linguagem de programação gráfica criada pela empresa *National Instruments* que tem como objetivo principal desenvolver rapidamente aplicações de medição e controle [19].

O seu estudo é complexo tendo demorado cerca de mês e meio para se obter alguma familiarização com o programa de modo a conseguir-se desenvolver o software necessário para se iniciar a leitura das variáveis a processar pelo protótipo desenvolvido.

Os programas em LabVIEW são chamados de VI's (Virtual Instruments) podendo ser chamados por outros VI's podendo possuir entradas e saídas. Existe uma interface que utiliza um painel frontal e toda a programação gráfica é feita num diagrama de blocos. Os blocos são ligados por linhas condutoras de sinal que definem o fluxo do programa. Um VI só é executado quando todas as suas entradas estiverem disponíveis e só são colocadas nas saídas as informações quando todo o VI tiver sido executado. Por esta razão a ordem pela qual o fluxo de dados é realizado depende das funções existentes, isto é, vai depender da rapidez de execução de cada bloco na obtenção dos dados. Isto implica que processos possam ser criados em paralelo ficando à responsabilidade do gestor de tarefas do LabVIEW a sequência de processamento dos cálculos. Deste modo, o LabVIEW é uma linguagem concorrente. Apesar de ser uma linguagem intuitiva e fácil de implementar, após se dominar, não permite uma gestão de recursos com grande eficiência. Possui no entanto uma variedade enorme de VI's de rápida utilização para o programador poder utilizar e fazer medições, cálculos e controlos.

Alguns VI's básicos do LabVIEW que tiveram interesse para a realização do projeto são os blocos que implementam ciclos (While's e For's), blocos de decisões (Case's), blocos que definem sequências de fluxos de dados (Flat Sequence), blocos de análises de sinais (FFT's, valores máximos) etc.

5.2.3.1 Programação LabVIEW

Tendo como objetivo inicial verificar a correta leitura dos valores recolhidos pela ADC e enviados para o microcontrolador, criou-se um programa em LabVIEW para este propósito. O diagrama geral deste programa está apresentado na Figura 5.27.

Para isso, o programa terá de passar pelas seguintes fases:

- Inicialmente fazer-se a configuração do SPI do microcontrolador
- De seguida fazer um pedido de leitura de um valor à ADC através do envio de uma palavra de configuração pela interface SPI
- Salvar num array pré-definido o valor recebido da ADC
- Caso o array não esteja todo preenchido volta a fazer leituras, caso contrário mostrar no painel de controlo os valores que constituem o array preenchido

- Depois de mostrar os valores no display se o utilizador quiser parar o programa este irá terminar fechando a comunicação SPI, caso contrário volta para a fase de leitura da ADC e preenchimento do array

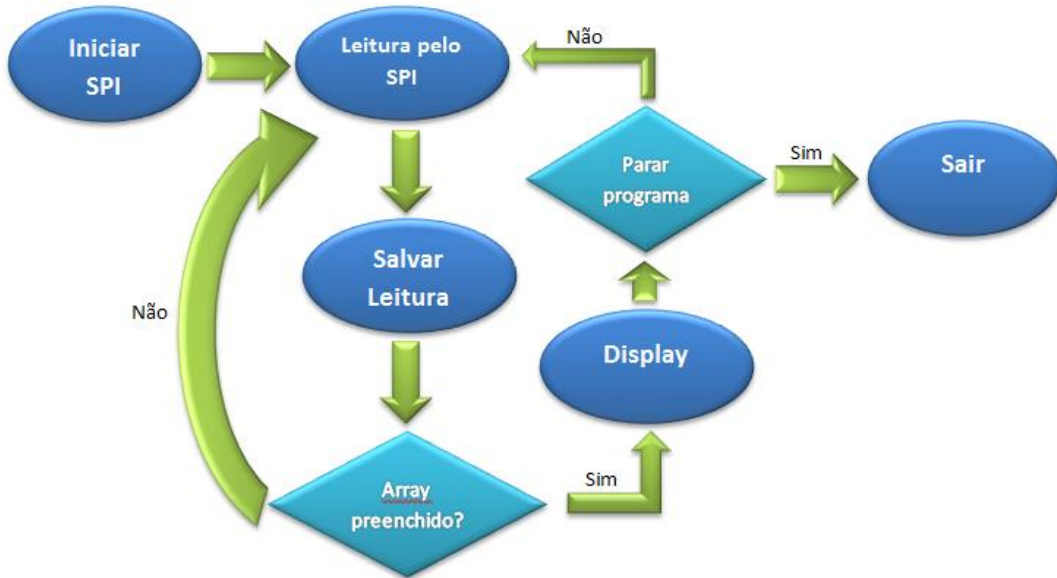


Figura 5.27 – Diagrama da leitura de valores

Começando pela configuração do SPI, já existe no LabVIEW um VI dedicado à sua configuração chamado “ARM SPI Create Configuration Reference.vi”. Foi assim usado esse bloco, tal como se mostra na Figura 5.28.

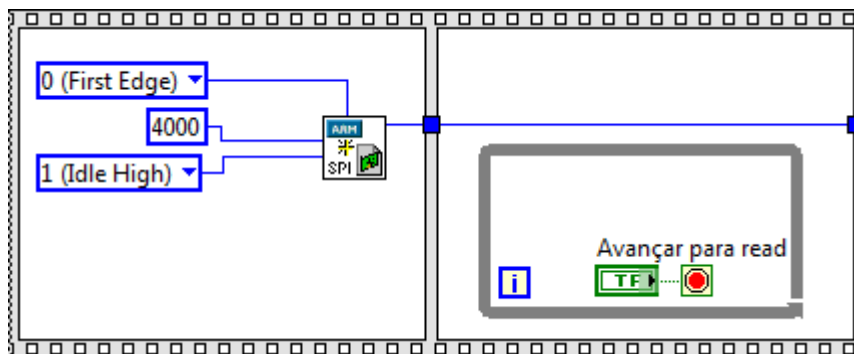


Figura 5.28 – Configuração do SPI pelo LabVIEW

O bloco de configuração do SPI foi programado a validar a informação no primeiro flanco de relógio (First Edge), o estado inativo com valor ativo (Idle High) e com uma frequência de relógio de 4000Khz.

Esta configuração só é necessária ser executada uma única vez em todo o programa podendo-se posteriormente proceder-se ao seu uso para leitura e escrita de dados passando o sinal de saída deste bloco para a entrada apropriada de outro bloco de leitura e escrita chamado de “ARM SPI Write Read.vi”. A utilização deste bloco está mostrada na Figura 5.29.

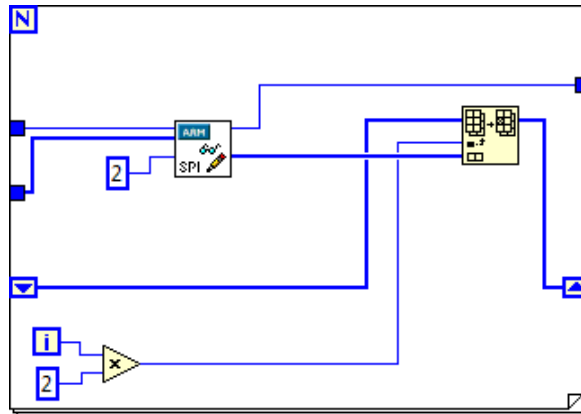


Figura 5.29 – Leitura e escrita do SPI no LabVIEW

O bloco “ARM SPI Write Read.vi” está inserido num ciclo “for” que tem, para este caso, parâmetros de entrada a configuração do SPI, um array onde serão guardados os valores lidos da [ADC](#) e o número de iterações que o ciclo “for” deve efetuar (N).

Quanto ao bloco de SPI de escrita e leitura este recebe três parâmetros, eles são:

- A configuração do SPI
- Um array com os valores de escrita que o microcontrolador deve enviar para a ADC de modo a receber a leitura na configuração desejada
- O número de bytes a ler (2 Bytes pois são enviados 16bits de informação pela ADC tal como foi visto na secção 5.2.2.1).

Como parâmetros de saída este bloco coloca à sua saída a configuração do SPI que recebeu e os dados obtidos pelo processo de leitura. Estes dados são entregues a um bloco (Replace Array Subset Function.vi) que os colocará de uma forma ordenada e sequencial os dados num array para depois poderem ser analisados.

Por fim, é necessário fechar todas as referências relativas à configuração do SPI. O bloco que faz essa tarefa é apresentado na Figura 5.30.

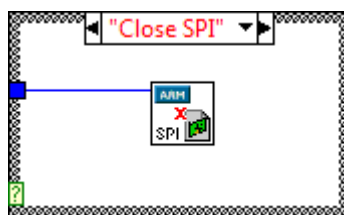
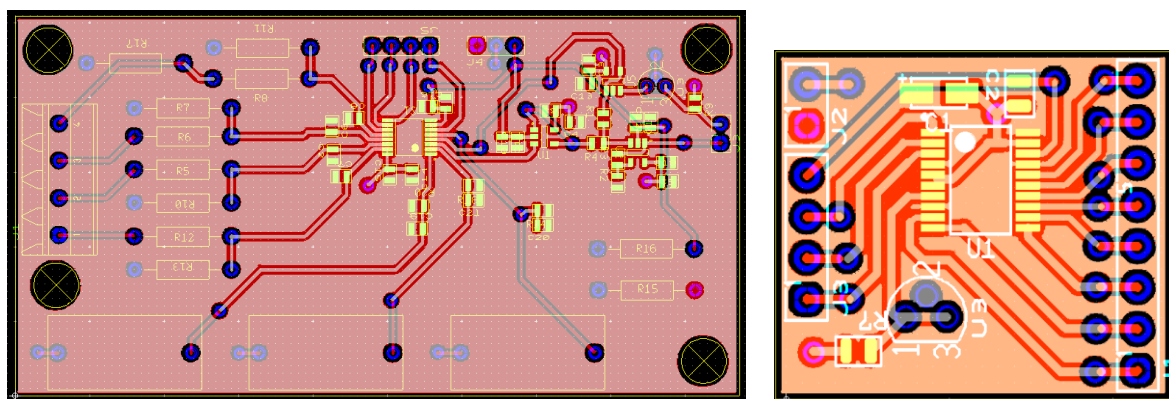


Figura 5.30 – Fecho da configuração do SPI no LabVIEW

5.2.4 Ensaio do protótipo

Na parte que diz respeito à construção do hardware projetado, ou seja, à placa de aquisição e acondicionamento de sinal e à placa onde se encontra a ADC (ambas apresentadas na secção 5.2.2), recorreu-se ao software Ultiboard da empresa *National Instruments* para o desenho das placas circuito impresso. Foram transferidos do software Multisim para o Ultiboard os ficheiros que representam os esquemáticos dos circuitos e assim foram desenhados e projetados todos os caminhos necessários ao fluxo dos vários sinais que interligam os componentes. O desenho das placas em 2D mostra-se na Figura 5.31.

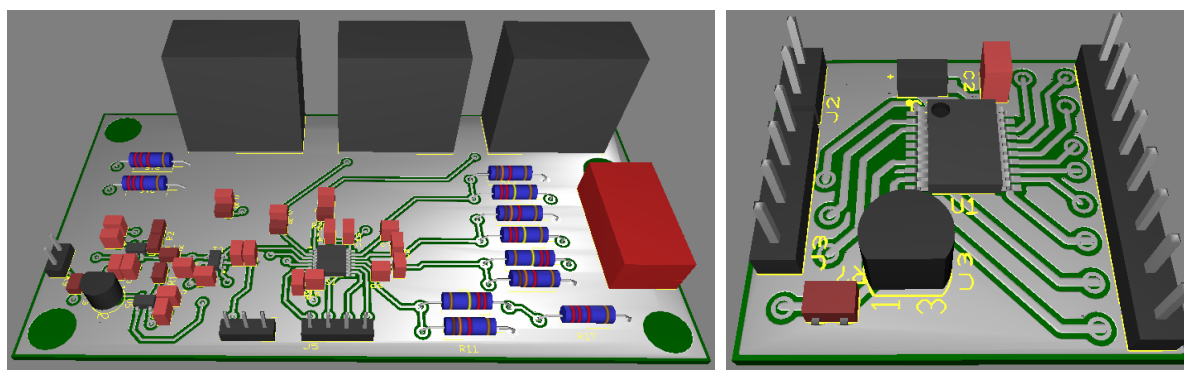


a) Placa de aquisição

b) Placa da ADC

Figura 5.31 – Desenho 2D das placas de hardware no Ultiboard

O *Ultiboard* também permite que se tenha uma perceção em 3D das placas que estão a ser desenhadas. A Figura 5.32 mostra as imagens 3D das placas desenvolvidas.



a) Placa de aquisição

b) Placa da ADC

Figura 5.32 – Vistas 3D das placas de hardware desenvolvidas

Tendo-se finalizado a construção das placas procedeu-se à implantação dos componentes eletrónicos obtendo-se o resultado apresentado na Figura 5.33.

Fazendo correr o programa LabVIEW apresentado na secção 5.2.3, utilizando apenas uma só entrada e o multiplexer fixo nesta, verificou-se que não era possível cumprir a velocidade de amostragem necessária (44800Hz) e assim não se conseguia obter na saída da ADC amostras que permitissem reproduzir vários ciclos da tensão de 50Hz de onde fosse possível extrair informação acerca das componentes harmónicas nomeadamente da 25ª. Para além deste problema se se quisesse alterar a entrada do multiplexer, amostrando outro sinal, verificava-se que a comutação era muito demorada (36us) o que não iria permitir estudar desfasagem entre os sinais.

Outro problema que se verificou foi o facto de o programa demorar demasiado tempo (180us) a voltar novamente a preencher um novo array de dados de amostras após ter completado o anterior.

Analisando os resultados anteriores desconfiou-se que o LabVIEW era um programa demasiado lento para permitir atingir os objetivos propostos. Tentou-se fazer uma análise mais profunda da sua estrutura mas debateu-se com a dificuldade de não existirem publicações que descrevessem em pormenor o seu funcionamento intrínseco. Através de uma análise mais profunda pode concluir-se que o LabVIEW não permite que um dos seus blocos de programa possa operar continuamente e concluir o seu trabalho sem que a estrutura principal do LabVIEW o interrompa para executar outros blocos, isto é, são atribuídos janelas no tempo a cada programa. Mesmo no caso de se ter apenas um bloco o sistema central do LabVIEW interrompe o processamento deste não o deixando operar continuamente. Tendo em vista resolver o problema tentou-se fazer-se uma limpeza ao código C gerado após a compilação do programa em LabVIEW tendo-se conseguido reduzir o tempo de amostragem mas não foi o suficiente para satisfazer a exigência dos requisitos que se pretendia.

Deste modo após ter-se despendido cerca de cinco meses do tempo de projeto conclui-se que seria necessário abandonar a ideia de desenvolver o software baseado no LabVIEW. Optou-se então por recorrer à construção do código baseado apenas na linguagem C.

5.3 Alteração do software

Tendo-se então concluído que a programação do Kit de desenvolvimento deveria ser em linguagem de programação C para assim existir rapidez suficiente para a execução das tarefas necessárias, procedeu-se à sua elaboração.

O diagrama de blocos do programa em linguagem C mostra-se na Figura 5.34. Inicialmente o programa começa por fazer as ativações e configurações necessárias para a sua execução, ou seja, faz a configuração do *Clock* do Kit de desenvolvimento, da interface SPI e dos portos dos pinos que irão fornecer a informação necessária à comutação dos canais do *multiplexer*.



Figura 5.34 – Diagrama de blocos do programa em C no Stellaris LM3S8962

Seguidamente procede-se à captura dos valores provenientes à saída da placa de aquisição e acondicionamento de sinal para *array's* previamente criados para o efeito. Nesta fase é decidido o canal que o *multiplexer* deve mostrar à saída e, seguidamente, enviada uma *word* à unidade ADC, através da interface SPI, para assim fazer a leitura deste canal e depois ser colocada a informação no *array* correspondente a esse canal.

Tendo sido a informação guardada nos arrays correspondentes a cada canal, procede-se à conversão dos valores lidos pela ADC (de 0V a 5V) para os valores reais, isto é, para valores de excursão de sinal entre $[-425V$ a $425V]$.

Terminado o processo de conversão, inicia-se a análise dos dados e calcula-se as características elétricas da rede. Efetuam-se os seguintes cálculos:

- Valores máximos, mínimos e RMS de cada canal
- Potência aparente, ativa e reativa
- Fator de potência

O código desenvolvido encontra-se no Anexo A – Código C do primeiro protótipo, desta dissertação.

5.3.1 Resultado via programação em C

Durante a elaboração e teste do código escrito, verificou-se que com a implementação do código em linguagem C o período de amostragem do sinal já cumpria com as características necessárias à concretização do projeto, ou seja, já possuía rapidez suficiente na aquisição de amostras e não existia algum tipo de interrupção no processo de captura dessas amostras.

Após verificada a correta conversão de valores pela ADC em todos os canais, verificou-se que os primeiros cálculos das características elétricas da rede (Valores máximos, mínimos e RMS, potências aparente, ativa e reativa e fator de potência) eram efetuados corretamente.

Sucedeu contudo que após realizadas as primeiras verificações a unidade programadora *ULINK2 da Keil* que permitia transferir do computador para o Kit *Stellaris LM3S8962* o programa em C compilado avariou impossibilitando a continuação dos testes e do projeto com o Kit de desenvolvimento *Stellaris LM3S8962*.

Devido ao elevado custo da unidade programadora, a gerência da empresa SELMATRON impôs que a nova solução devesse passar por um Kit de desenvolvimento mais económico. Após algumas análises dos Kits de desenvolvimento existentes no mercado, optou-se por continuar o desenvolvimento do projeto utilizando o Kit *dsPICDEM™ 80 – Pin Starter Development Board* da empresa *Microchip*.

Esta nova abordagem à solução do problema proposto para o projeto, foi necessária a construção de uma nova placa de aquisição e acondicionamento de sinal assim como a escrita de um novo código em linguagem de programação C que configurasse corretamente o novo Kit de desenvolvimento. Todos estes passos são explicados na secção 5.4 onde se fará uma análise completa a todas as partes intervenientes da nova solução.

5.4 Segundo protótipo

Nesta secção apresenta-se a solução do projeto desenvolvido utilizando o Kit de desenvolvimento *dsPICDEM* da *Microchip* e a linguagem de programação C. Tal como é apresentado na secção 5.4.1, este Kit possui características substancialmente diferentes em relação ao que se tinha utilizado anteriormente (Kit *Stellaris LM3S8962*). De modo a se aproveitar melhor as vantagens deste novo Kit procedeu-se também à construção de um novo *hardware* de aquisição de dados em vez de se utilizar o protótipo desenvolvido na secção 5.2.2. Este apresenta algumas diferenças que serão explicadas melhor na secção 5.4.2.

Tendo este novo Kit a possibilidade de fazer as leituras de tensão simultaneamente de todos os canais de entrada necessários (7 canais) através de ADC's internas, não se torna necessário recorrer a uma ADC externa para o efeito. Isto implica que também já não existe a necessidade de haver sinais de seleção de canais. A tentativa de solução via multiplexagem foi então abandonada fazendo-se para cada canal de entrada a aquisição e acondicionamento de sinal individualmente.

Deste modo, o diagrama de blocos desta nova solução do projeto é apresentado na Figura 5.35 onde L1, L2, L3 representam as 3 fases da tensão e corrente da rede elétrica, N o neutro, C1, C2 e C3 os 3 sinais das 3 fases da tensão e corrente da rede elétrica já acondicionados e CN o neutro também já acondicionado. A saída de informação da unidade desenvolvida é feita via unidade pickit3 devendo no futuro ser feita pelo canal de comunicação RS-232 do kit ou via SPI (ver secção 5.4.1).

A unidade pickit3 foi desenvolvida pela Microchip tendo em vista programar e testar os processadores da mesmo fabricante. Os programas em C podem ser compilados e testados utilizando esta unidade que liga ao PC via USB e ao kit de desenvolvimento via RJ-11

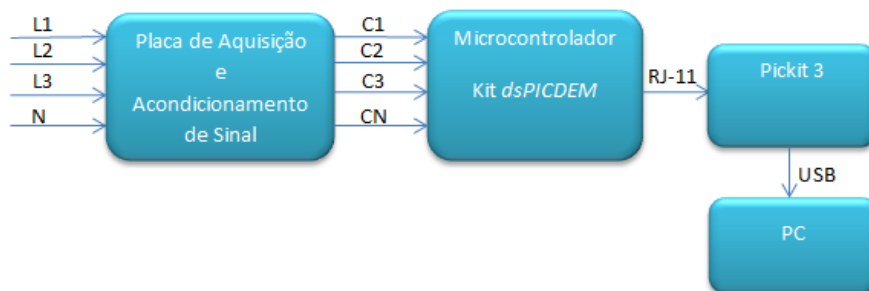


Figura 5.35 Diagrama do segundo protótipo

5.4.1 Kit de desenvolvimento dsPICDEM-80

Como já se disse, para este projeto foi usado um kit de desenvolvimento da Microchip designado por *dsPICDEM™ 80 – Pin Starter Development Board*, representado na figura Figura 5.36.

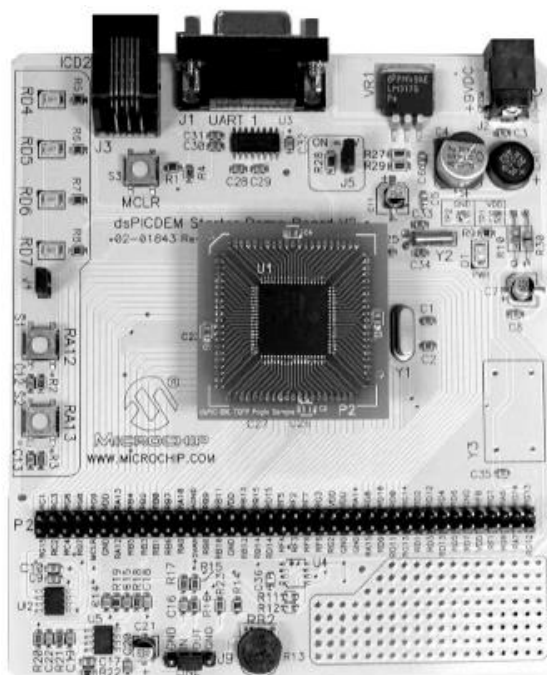


Figura 5.36 – Kit de desenvolvimento dsPICDEM

Este Kit de desenvolvimento é simples de programar e integrar com *hardware* externo apresentando disponíveis todos os pinos dos módulos CPU que suporta. Os módulos que suporta são o dsPIC30F e dsPIC33F que são módulos dedicados especialmente ao processamento de sinal digitais onde a rapidez de cálculo é necessária. A dsPIC33F é um microcontrolador mais sofisticado do que a dsPIC30F possui maior rapidez e consome menos energia para a realização das suas tarefas. Por esta razão foi este o módulo escolhido para o projeto. Deste modo, as características principais do Kit de desenvolvimento são as seguintes:

- Microcontrolador *dsPIC33FJ256GP710A*
- 9VDC de alimentação
- LED indicador de energia
- Conector programador MPLAB ICD 2
- Canal de comunicação RS-232
- Potenciômetro de 10 K Ω
- Potenciômetro digital Microchip MCP41010
- Amplificador Operacional Microchip MCP6022
- Cristal de 7.37 MHz
- Botão switch reset
- Quatro LEDs vermelhos
- Dois botões de switch
- 80-pin dual-row header com pinos identificados
- Área protótipo para uso de hardware

O microcontrolador utilizado (*dsPIC33FJ256GP710A*) é um microcontrolador de 16 bits. Suporta DSP (em inglês, *Digital Signal Processor*) o que é uma vantagem comparado com outras famílias de gama inferior. Os microcontroladores com DSP são especializados para tratamento digital de sinal a grande velocidade mesmo em tempo real. Conseguem fazer cálculos de maior complexidade de maneira eficaz num curto espaço de tempo, como por exemplo as transformadas de Fourier que são cálculos pesados para microcontroladores.

Uma característica bastante importante deste microcontrolador para este projeto é o facto de possuir duas ADC's internas com possibilidade de ler simultaneamente, cada uma, quatro canais diferentes, o que irá permitir a leitura em simultâneo dos 7 canais que o projeto necessita.

Algumas características técnicas do microcontrolador são as seguintes:

- Até 40 MIPS
- 8 canais de hardware de DMA
- 2 Kbytes de DMA buffer
- Até 5 interrupções externas
- SRAM de 30 Kbytes
- Memória Flash de 256 KBytes
- 2 ADC's internas
- As ADC podem operar com 10-bit a 1.1 MSPS ou 12-bit a 500 KSPS
- Capacidade das ADC capturar 2, 4, ou 8 valores simultaneamente.
- Duas interfaces de comunicação de SPI, I²C, UART e CAN.
- 9 Timers de 16-bit

As duas ADC's existentes no microcontrolador *dsPIC33F* possuem, cada uma, quatro canais que possuem quatro opções de operação (formato inteiros, com ou sem sinal, e formato fracionário, com ou sem sinal). Em qualquer um destes formatos podem operar com 9-bits e sinal ou 10-bits sem sinal.

Analisando estas características do microcontrolador verifica-se que possui condições suficientes para a concretização do projeto em causa.

5.4.2 Placa de aquisição (segundo protótipo)

À semelhança do que foi feito para a primeira placa de aquisição e acondicionamento de sinal utilizada no primeiro protótipo (secção 5.2.2), nesta secção apresentam-se os cálculos e o desenvolvimento do segundo protótipo. Representa-se na Figura 5.37 o novo esquemático para esta nova solução do problema tendo sido desenhado com recurso ao *software Multisim* e tendo em vista a sua construção em circuito impresso. O esquemático está dividido em secções de I a IX que estão detalhadas nas próximas subsecções.

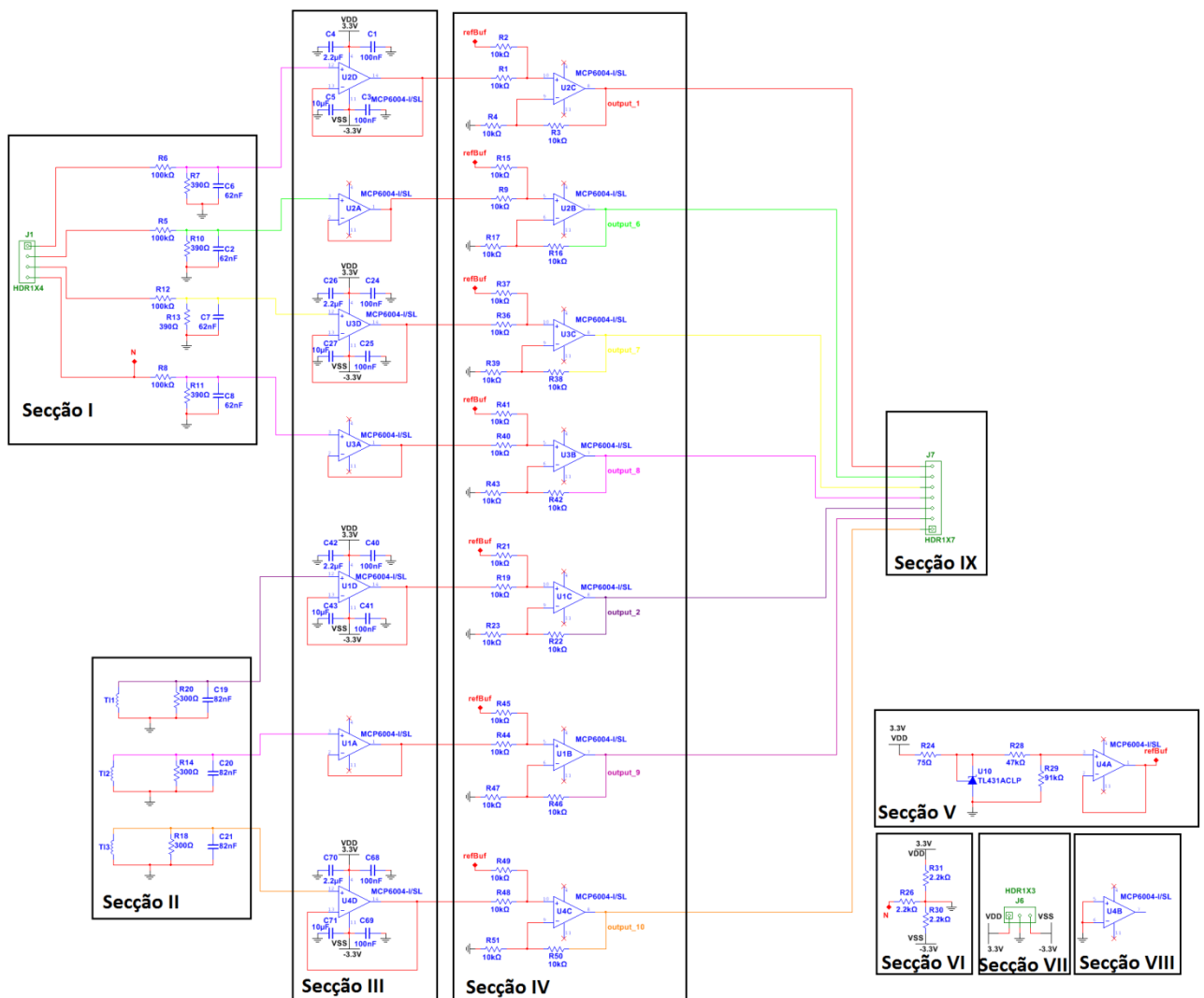


Figura 5.37 – Esquemático da placa de aquisição e acondicionamento de sinal do segundo protótipo

Nesta abordagem optou-se por fazer o acondicionamento de sinal individualmente para cada canal de entrada para evitar os problemas que surgiram no protótipo anterior face às dificuldades que a multiplexagem apresentava a nível de rapidez de aquisição. Uma vez que o Kit de desenvolvimento apresenta a possibilidade de serem lidos em simultâneo os 7 sinais necessários à elaboração do projeto, estes serão passados, após serem desacoplados e acondicionados, diretamente ao microcontrolador pela Secção IX.

As tensões de alimentação da placa de aquisição são introduzidas através de um ligador (secção VI, Figura 5.37) onde VDD e VSS correspondem a 3,3 e -3,3V, respetivamente. Junto aos pinos de entrada de alimentação de todos os componentes não passivos foram colocados condensadores de $10\mu F$ e $100nF$ para atenuar efeitos oscilatórios de altas frequências e baixas frequências, respetivamente, que possam ocorrer.

Todas as tensões estão interligadas às tensões de referência do modo como se indica na secção VI, Figura 5.37, como explicado anteriormente na secção 5.1.8.

Na secção VIII da Figura 5.37 mostra-se um Op-Amp com as suas entradas ligadas à massa pois foi o único não usado uma vez que os circuitos integrados possuem quatro Op-Amp's cada um (*OPA4348* da *Texas Instruments*) e este não foi necessário para o projeto. Desta maneira este Op-amp não irá provocar interferências internas significativas no circuito.

5.4.2.1 Divisores resistivos com filtragem (Secção I)

Na Figura 5.38 apresenta-se o diagrama da secção I da placa de aquisição.

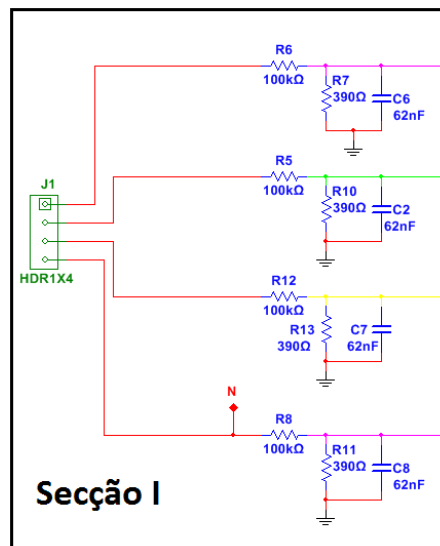


Figura 5.38 – Divisores resistivos e filtragem da placa de aquisição e acondicionamento do segundo protótipo

Esta secção é idêntica à que foi projetada na secção 5.2.2.2 tendo a mesma funcionalidade. Apenas é agora necessário recalculer os valores da resistências e condensadores de modo as tensões à saída estarem compreendidas entre 0V a 3.3V.

Tendo igualmente a mesma excursão de sinal à entrada de $[-425V ; 425V]$ e pretendendo neste caso colocar à saída tensões no intervalo de $[-1,65V ; 1,65V]$ temos que entre resistências o rácio deverá ser:

$$\frac{V_{out}}{V_{in}} = \frac{1,65}{425} \approx 0,003882 \text{ [V/V]} \quad (5-41)$$

Pode-se chegar à conclusão que o valor das resistências da série E24 que melhor descrevem esta relação são as seguintes:

$$R_1 = 100K\Omega \quad (5-42)$$

$$R_2 = 390\Omega \quad (5-43)$$

Utilizando estas resistências a relação $\frac{V_{out}}{V_{in}}$ será:

$$\frac{V_{out}}{V_{in}} = \frac{R_2}{R_1 + R_2} = \frac{390\Omega}{100K\Omega + 390\Omega} \approx 0,003884 \text{ [V/V]} \quad (5-44)$$

O que é um valor muito próximo do pretendido.

Verificando a potência máxima dissipada nas resistências é:

$$P_{R1} = \frac{V^2}{R_1} = \frac{(425 - 1,65)^2}{100K} \approx 1,792 \text{ W} \quad (5-45)$$

$$P_{R2} = \frac{V^2}{R_2} = \frac{1,65^2}{390} \approx 0,007 \text{ W} \quad (5-46)$$

Mais uma vez a resistência R_1 excede os valores de potência normais suportados pelas resistências comuns (1/3 W) sendo necessário o uso de resistências de potência de pelo menos 2W.

Recalculando o valor do condensador para a mesma frequência de corte (6.4KHz) obtém-se:

$$C = \frac{1}{2\pi(R_1 \parallel R_2)f_c} = \frac{1}{2\pi \cdot (100K\Omega \parallel 390\Omega) \cdot 6400} = 64 \text{ nF} \quad (5-47)$$

Não existindo este valor optou-se pelo condensador de valor 56nF que apresenta uma frequência de corte de:

$$f_c = \frac{1}{2\pi(R_1 \parallel R_2)C} = \frac{1}{2\pi \cdot (100K\Omega \parallel 390\Omega) \cdot (56nF)} \approx 7,3 \text{ KHz} \quad (5-48)$$

O que é admissível.

Fazendo a confirmação dos cálculos através de simulação pelo *software MultiSim*, verificou-se que os cálculos efetuados correspondiam aos simulados, tal como mostra a Figura 5.39.

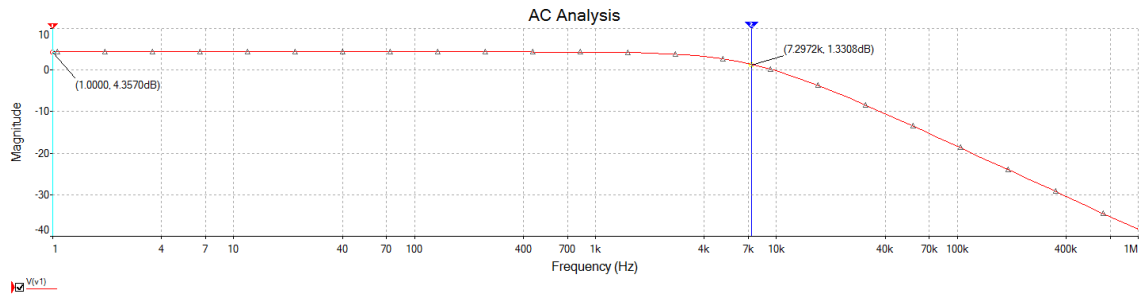


Figura 5.39 – Resposta em frequência do filtro com os divisores resistivos do segundo protótipo

No eixo vertical a magnitude do sinal é representada em decibel.

Neste gráfico, verifica-se que para à frequência de 7.3KHz a magnitude cai 3dB, o que representa a frequências de corte do filtro.

5.4.2.2 Transformadores de intensidade com filtragem (Secção II)

Na Figura 5.22, apresenta-se o diagrama da secção II da placa de aquisição.

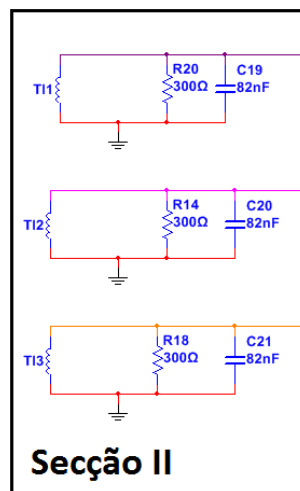


Figura 5.40 – Transformadores de corrente e aquisição do segundo protótipo

Esta secção é idêntica à que foi projetada na secção 5.2.2.3 tendo a mesma funcionalidade.

Assim, usando os mesmos transformadores de intensidade, fazendo para efeitos de testes uma corrente máxima de 5A a passar pelo primário e querendo-se ter uma tensão máxima de $V_{out} = 1,65V$ então o valor da resistência será:

$$R = \frac{885,9 \cdot V_{out}}{I_{in}} = \frac{885,9 \cdot 1,65}{5} = 292,35 \Omega \quad (5-49)$$

Utilizando resistências de 300Ω a corrente máxima que o equipamento poderia ler será então:

$$I_{in} = \frac{885,9 \cdot V_{out}}{R} = \frac{885,9 \cdot 1,65}{300} = 4,872 \text{ A} \quad (5-50)$$

O que corresponde a uma corrente à no secundário do transformador de:

$$I_{out} = \frac{I_{in}}{885,9} = \frac{4,872}{885,9} \approx 5,5 \text{ mA} \quad (5-51)$$

Fazendo o cálculo da potência máxima consumida pela resistência temos:

$$P_{R1} = \frac{V^2}{R} = \frac{1,65^2}{300} \approx 10 \text{ mW} \quad (5-52)$$

O que é uma potência que resistências SMD suportam (1/8 W) e na qual foi utilizado.

Recalculando o valor dos condensadores de modo a se ter a frequência de corte de 6.4KHz tem-se:

$$C = \frac{1}{2\pi R f_c} = \frac{1}{2\pi \cdot 300 \cdot 6400} = 83\text{nF} \quad (5-53)$$

O valor do condensador existente em armazém mais adequado foi de 82nF obtendo-se assim uma frequência de corte de:

$$f_c = \frac{1}{2\pi \cdot R \cdot C} = \frac{1}{2\pi \cdot 300 \cdot 82\text{nF}} = 6469\text{Hz} \quad (5-54)$$

Valor considerado admissível.

Fazendo a simulação do circuito através do *software MultiSim*, verificou-se mais uma vez que os cálculos efetuados correspondiam aos simulados, tal como mostra a Figura 5.41.

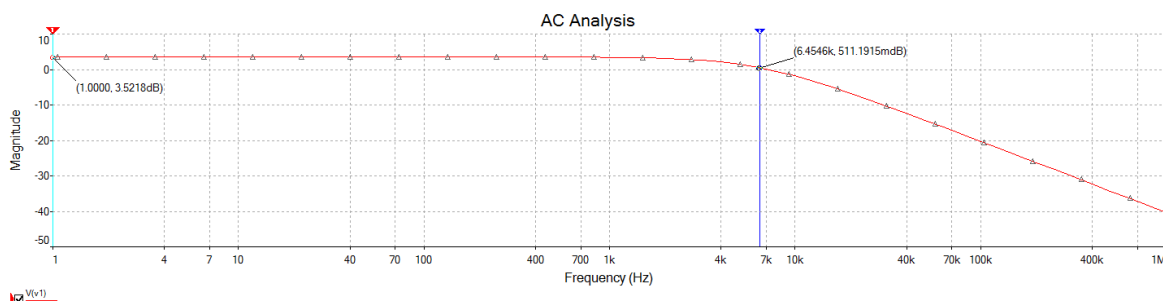


Figura 5.41 – Resposta em frequência do filtro após os TI's do segundo protótipo

No eixo vertical a magnitude do sinal é representada em decibel.

Neste gráfico, verifica-se que para a frequência de 6454Hz a magnitude cai 3dB, o que representa a frequências de corte.

5.4.2.3 Regulador de tensão do segundo protótipo (Secção V)

Na Figura 5.42 está representada a secção V do segundo protótipo de *hardware* do projeto.

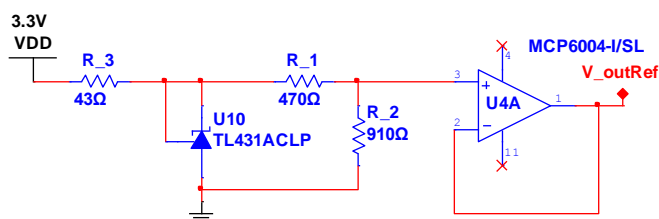


Figura 5.42 – Regulador de tensão do segundo protótipo

Esta parte do circuito tem como objetivo fornecer um valor de tensão de referência preciso e estável às entradas dos circuitos somadores, que se apresentam na secção IV da Figura 5.37, na qual foram projetados do mesmo modo que na secção 5.2.2.6 do primeiro protótipo.

Assim, usando novamente o regulador TL431, que tem como tensão de referência 2,5V quando polarizado inversamente, pretende-se ter na saída V_outRef uma tensão de referência de valor 1,65V que corresponde a metade do intervalo de excursão de sinal pretendido à saída da placa de acondicionamento de sinal. Esta tensão irá ser passada aos circuitos somadores e fazer que à saída da placa de acondicionamento de sinal estejam os valores compreendidos na gama desejada [0V a 3,3V].

5.4.2.4 Ensaio da placa de aquisição (segundo protótipo)

Para o segundo protótipo, recorreu-se novamente ao software Ultiboard da empresa *National Instruments* para o desenho das placas circuito impresso. Foram desenhados e projetados todos os caminhos necessários ao fluxo dos vários sinais que interligam os componentes. O desenho das placas em 2D mostra-se na Figura 5.43.

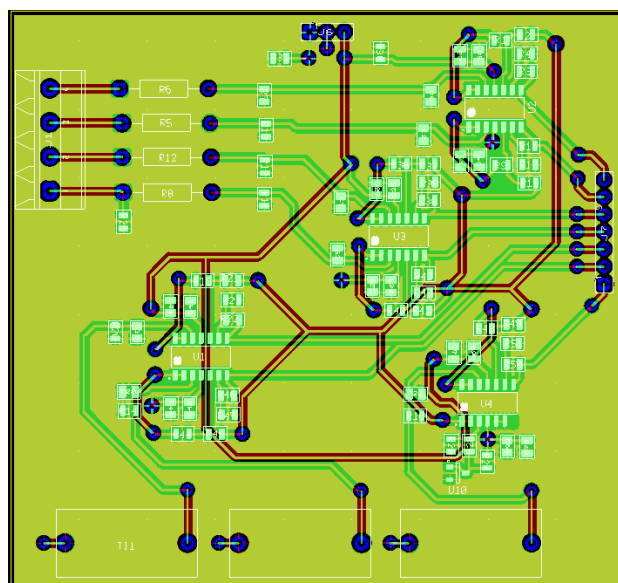


Figura 5.43 – Desenho 2D da Placa de Hardware do segundo protótipo

Tendo uma perceção 3D do segundo protótipo, a Figura 5.44 mostra as imagens 3D das placas desenvolvidas.

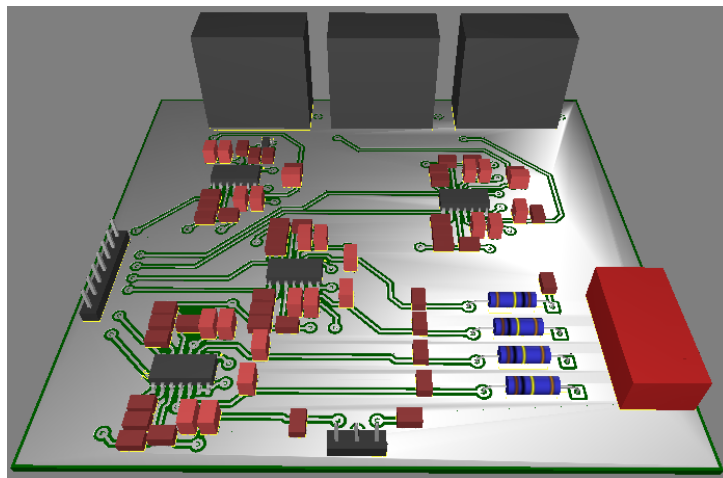


Figura 5.44 – Vistas 3D da placa de hardware do segundo protótipo

Tendo-se finalizado a construção das placas procedeu-se à implantação dos componentes eletrónicos obtendo-se o resultado apresentado na Figura 5.45.

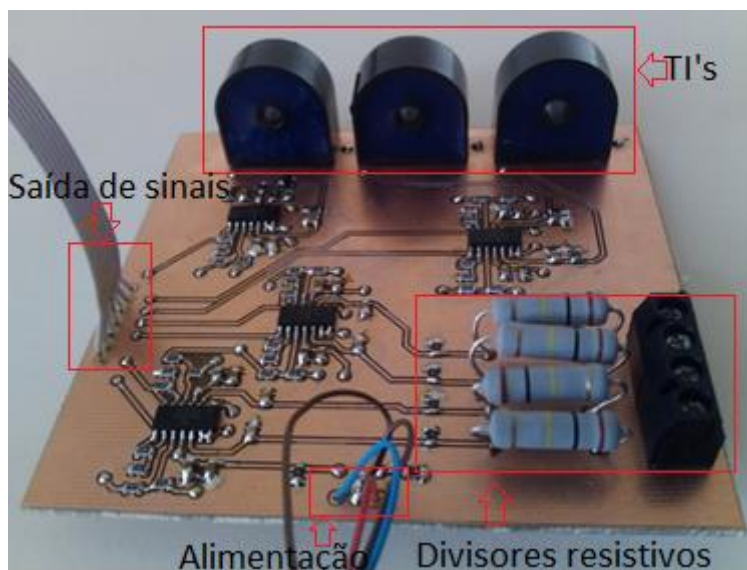


Figura 5.45 – Conclusão da construção do segundo protótipo

Após o processo de construção estar concluído procedeu-se aos testes de fluxo de sinais.

Fazer-se a alimentação da placa com as tensões de 3,3V e -3,3V, verificou-se primeiramente a tensão gerada à saída do circuito do regulador de tensão (V_{outRef}), apresentado na Figura 5.42. Este apresenta o valor de 1.68V que é um valor dentro dos valores esperados.

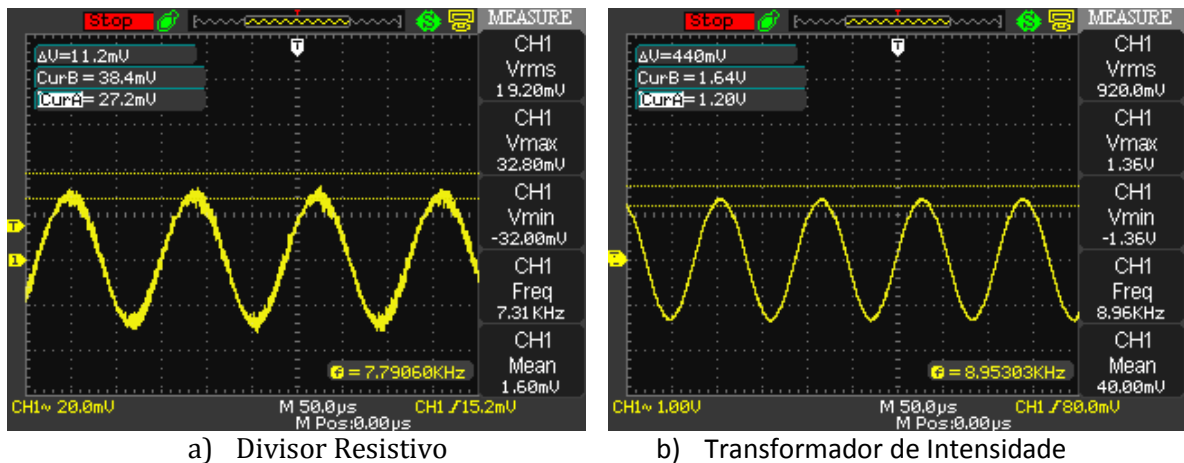
Para efeitos de testes foi aplicado, nas estradas dos divisores resistivos (secção 5.4.2.1), um sinal sinusoidal de aproximadamente 10V de amplitude. Este sinal serviu para verificar a queda de tensão provocado pelos divisores resistivos e assim testar o circuito com parâmetros bem

definidos sem ser necessário ligar-se os 230Vrms da rede elétrica nas entradas de tensão do circuito.

Verificou-se assim que a tensão após os divisores resistivos tinha a relação projetada com a equação (5-41), tendo uma tensão de pico de 38,4mV.

Em seguida, verificou-se para cada um das entradas de tensão os sinais gerados à saída da placa. Estes apresentaram um aumento em tensão de aproximadamente 1,65V em relação à entrada do circuito somador de acordo com o esperado, ou seja, tiveram um aumento em tensão de 1,65V provocada pelos circuitos somador e regulador mostrado na Figura 5.26 a) e Figura 5.42 respetivamente.

Para verificar o funcionamento dos filtros (Da Figura 5.38) aumentou-se a frequência da onda gerada até que se verificasse um decaimento de 3dB no sinal de saída, ou seja, o sinal de saída apresentava-se com uma amplitude aproximadamente de 70,7% do sinal de entrada. Deste modo, construiu-se a Tabela 5-2 que mostra as imagens capturadas à saída do osciloscópio assim como as frequências previstas de corte e medidas.



a) Divisor Resistivo

b) Transformador de Intensidade

	Frequência de Corte prevista (KHz)	Frequência de Corte Medida
Divisores Resistivos	7,3KHz	7,31KHz
Transformadores de Corrente	6,5KHz	8,96KHz

Tabela 5-2 – Frequências de corte

Pode concluir-se que as frequências de cortes não se desviam muito do que foi calculado.

Posteriormente procedeu-se aos testes dos transformadores de intensidade. Fez-se passar os 230Vrms da rede elétrica por uma carga resistiva de 672Ω o que faz gerar uma corrente do seguinte valor:

$$I_{RMS,1} = \frac{V_{RMS}}{R} = \frac{230}{672} \approx 0,342 A_{rms} \quad (5-55)$$

Fazendo passar esta corrente pelo circuito primário dos transformadores de intensidade, estes geraram uma tensão aos terminais da resistência de 300Ω colocada no circuito secundário dos

transformadores de aproximadamente 0,115Vrms. Isto implica que a corrente que atravessa esta resistência após o circuito secundário seja dada por:

$$I_{RMS,2} = \frac{V_{RMS}}{R} = \frac{0,115}{300} \approx 0,383 \text{ mA}_{rms} \quad (5-56)$$

Calculando a relação de transformação n do transformador, verifica-se que é:

$$n = \frac{I_{RMS,1}}{I_{RMS,2}} = \frac{0,342 \text{ A}_{rms}}{0,383 \text{ mA}_{rms}} \approx 893 \quad (5-57)$$

Este valor encontra-se bastante próximo do esperado (885,9) e dado pela equação (5-3).

Aplicando a tensão da rede elétrica numa carga resistiva, onde idealmente a tensão coincidia com a onda de corrente, pode verificar-se se existia algum desvio de fase provocada pelo circuito de aquisição (Figura 5.37) na saída. A Figura 5.46 mostra-se o resultado obtido.

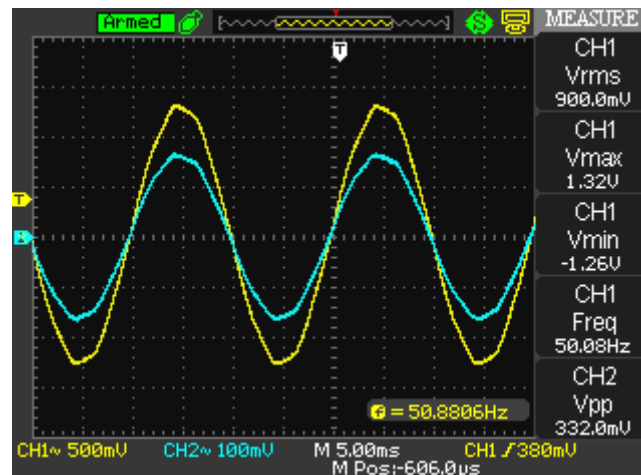


Figura 5.46 – Verificação de diferença de fases provocada pela placa de aquisição

Daqui pode-se concluir que não existe nenhum desvio de fase significativo pois a onda de corrente (CH2 – Azul) coincide temporalmente com a onda de tensão da rede (CH1 – Amarelo).

5.4.3 Software

No desenvolvimento do *software* para este segundo protótipo existiram duas abordagens. Na primeira abordagem mostra-se uma tentativa de solução que inicialmente se considerou ideal para o problema. A segunda abordagem é uma solução simplificada da primeira uma vez que nem todos os aspetos foram possíveis alcançar.

5.4.3.1 Primeira abordagem

Como já foi referido anteriormente, a programação do novo kit de desenvolvimento foi escrita em linguagem C.

Tirando proveito das capacidades do kit de desenvolvimento, nomeadamente o permitir que as saídas das ADC's sejam enviadas via DMA (*Direct Memory Access*) para a RAM (O espaço total da RAM dedicado à DMA é de apenas 2Kbytes), elaborou-se uma ideia de um programa que fizesse a leitura dos valores necessários à concretização do projeto. A Figura 5.47 ilustra esse diagrama geral da captura de valores desde a ADC ao seu armazenamento na memória RAM para posteriormente se proceder aos cálculos.

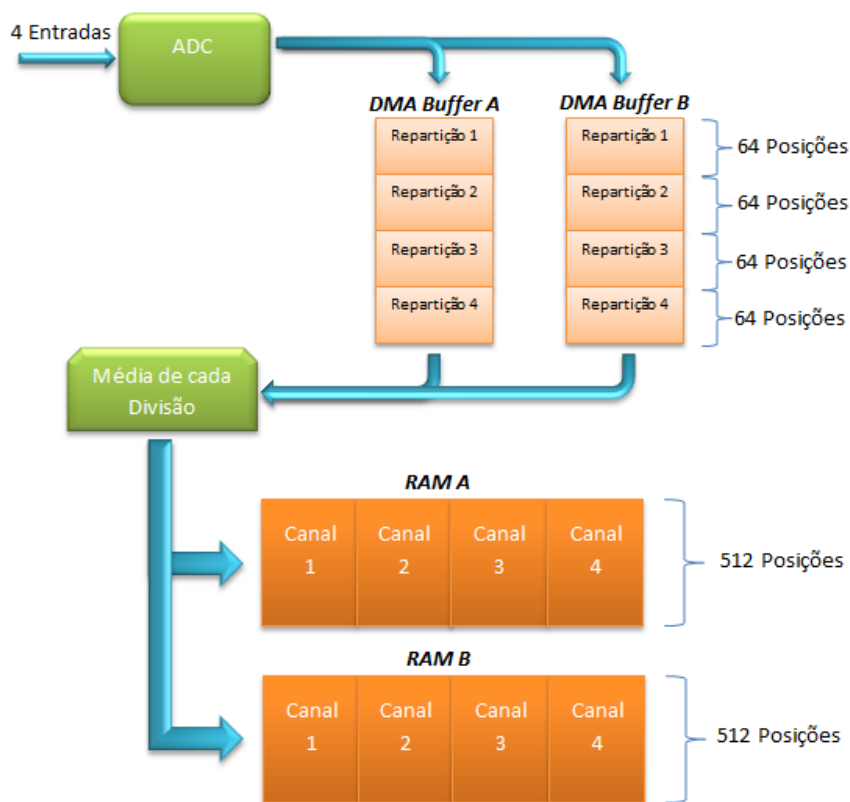


Figura 5.47 – Diagrama de aquisição de valores

Por razões de simplicidade apresenta-se apenas o funcionamento de uma única ADC com os respetivos espaços de armazenamento dos valores capturados. A leitura das restantes três entradas (A quarta entrada não é utilizada por não ser necessária) é feita pela outra ADC do microcontrolador tendo o mesmo diagrama de funcionamento.

Neste diagrama de blocos, a ADC faz leituras simultâneas das quatro entradas que possui (optou-se por um funcionamento da ADC em formato fracional com 9-bits e sinal por esta opção permitir obter diretamente sinais positivos e negativos tomando como média o valor de tensão entre 3,3V

e 0V). Como cada leitura corresponde a 10-bits, o seu armazenamento em memória RAM requer dois bytes por leitura.

Seguidamente, via DMA (Direct Memory Access), essas amostras são guardadas separadamente em quatro repartições diferentes que integram os buffers de memória dedicado ao DMA. Cada buffer de memória DMA pode possuir até 64 posições, o que dá um total de 64 posições x 8 repartições x 2 ADC's x 2 Bytes = 2048 Bytes de memória RAM DMA para oito entradas.

Os buffers de DMA são espaços da memória RAM que vão funcionar alternadamente, ou seja, os quatro valores lidos simultaneamente por cada a ADC são guardados, cada um deles, nas suas repartições dedicadas (Repartição 1...4) de um dos buffers de DMA. Quando estas repartições estiverem totalmente preenchidas, os próximos valores lidos serão guardados nas repartições do outro buffer de DMA e assim sucessivamente.

Isto tem a vantagem de permitir efetuar o cálculo dos valores médios armazenados em cada um dos buffers durante o tempo em que as próximas leituras da ADC são guardadas no outro buffer, fazendo com que não se perca informações acerca dos dados a medir. Deste modo, quando as repartições de um dos DMA Buffers estiverem totalmente preenchidas, é lançada uma interrupção ao CPU que irá calcular a média de cada repartição do buffer de DMA que desencadeou essa interrupção e colocar os valores médios em posições de memória da RAM destinada ao seu futuro processamento e análise. Esta determinação dos valores médios permitirá a filtragem dos possíveis ruídos que possam ocorrer durante a captura de valores pela unidade ADC.

Assim, a captura é efetuada a uma frequência mais elevada, ou seja, haverá uma sobre amostragem de valores, e ao fazer-se a filtragem (média) serão removidas as variações significativas da amplitude das ondas da energia elétrica que ocorreram nesse intervalo de tempo.

Com a ideia da sobre amostragem, pretende-se tomar partido da alta velocidade do funcionamento em DMA e aproveitar todo o espaço de memória de 2Kbytes dedicado ao DMA.

Como se tem 8 canais de entrada para as duas ADC's, 2 buffers de DMA e cada amostra fornecida pela ADC ocupa dois bytes, então o número de posições de armazenamento de valores de cada repartição dos buffers de DMA é dado por:

$$\text{Número de posições} = \frac{2KB}{8\text{Canais} \times 2\text{buffers} \times 2\text{bytes}} = 64 \text{ posições} \quad (5-58)$$

Criaram-se também dois buffers na memória RAM, o *Buffer RAM A* e *Buffer RAM B*, que têm como objetivo funcionar alternadamente de modo a permitir a análise contínua dos valores obtidos sem que exista uma perda de informação durante o tempo necessário para efetuar os cálculos.

Cada Buffer RAM está também dividido em quatro canais que destinam-se ao armazenamento dos valores fornecidos pelo algoritmo que calcula a média e que se inicia pela rotina de serviço à interrupção do DMA.

Após o armazenamento completo de um dos Buffer RAM, será iniciado um programa que efetua a análise dos valores seguintes (se a velocidade do CPU assim o permitir):

- Valor máximo, Valor mínimo e Valor TRMS
- Potencias Ativa, Reativa e Aparente
- Detecção do sentido do fluxo de energia
- Fator de Potência
- FFT
- Amplitude das harmónicas e suas percentagens relativas
- THD
- Frequência

Após estes cálculos o CPU ficará à espera de novos valores do outro buffer RAM para uma nova análise.

O esquema do programa a implementar está representado na Figura 5.48:

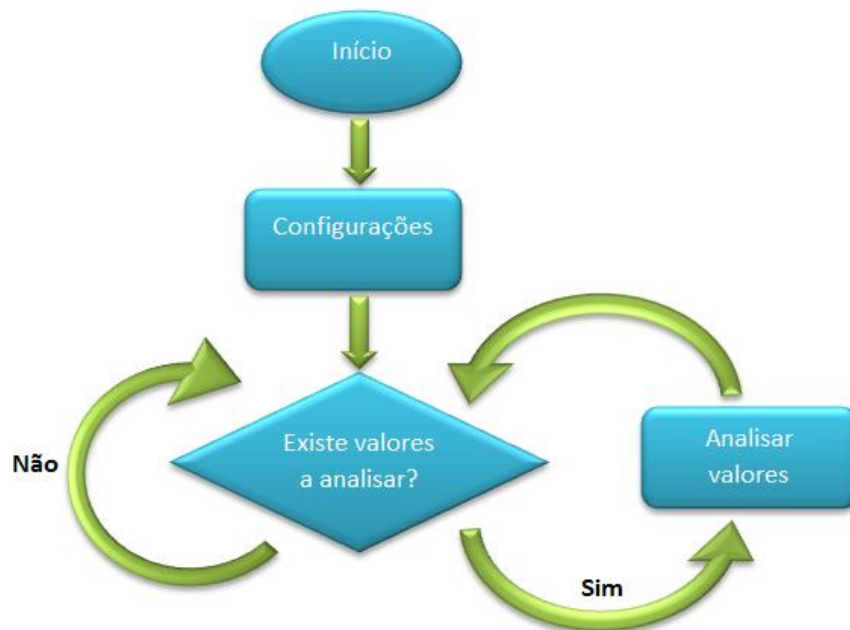


Figura 5.48 – Diagrama de fluxo de sinal do programa

5.4.3.1.1 Resultados obtidos

Ao fazer-se o desenvolvimento do programa de modo a corresponder ao diagrama apresentado na Figura 5.47, verificou-se que existiam dois problemas.

Um dos problemas estava relacionado com o preenchimento dos buffers de DMA. Verificou-se que o hardware apenas permite nas configurações do DMA a escrita de valores em blocos de

memória cuja dimensão são potências de 2. Uma vez que para o preenchimento sequencial dos buffers é necessário que os timers, que comandam a captura de valores pela ADC, possuam um período bem definido. Isto é, sendo o período de amostragem dos sinais de entrada pretendido de 156,25us (tal como foi visto na secção 3.2), para se fazer a sobre amostragem e aproveitar-se todo o espaço de memória dedicada a cada buffer de DMA, ter-se-ia de dividir este período pelo número de posições existentes em cada repartição do buffer. Deste modo, verifica-se qual o período necessário à captura de cada amostra por efeito de sobre amostragem ($T_{s.a.}$) que é dado por:

$$T_{s.a.} = \frac{T_a}{\text{Número de posições por repartição}} \quad (5-59)$$

Assim, $T_{s.a.}$ teria de ser um número inteiro do período de funcionamento do CPU (40MHz, isto é, $T_{CPU} = 25ns$) para que os timers desencadeassem no tempo correto a captura de valores pela ADC, ou seja:

$$\text{Número de ciclos dos timers} = \frac{T_{s.a.}}{T_{CPU}} \quad (5-60)$$

A Tabela 5-3 mostra os cálculos obtidos.

#Posições por repartição	$T_{s.a.}$ (us)	#Ciclos dos timers	Ciclos completos dos timers
64	2,44140625	97,65625	X
32	4,8828125	195,3125	X
16	9,765625	390,625	X
8	19,53125	781,25	X
4	39,0625	1562,5	X
2	78,125	3125	✓

Tabela 5-3 – Verificação dos ciclos completos dos timers

Desta tabela pode-se verificar que apenas blocos de DMA de 2 posições permitem ciclos completos dos timers.

Outro problema observado foi relativo ao tempo necessário para os cálculos dos parâmetros residentes nos buffers RAM. Este tempo de processamento das 512 posições por canal de cada um dos buffers (RAM A ou RAM B), mesmo para duas posições de memória DMA por repartição, excede o tempo de preenchimento do outro buffer RAM, perdendo-se assim dados e a objetividade do diagrama proposto, ou seja, fazer uma análise contínua de todos os sinais de entrada.

Perante os dois problemas acima citados teve-se que abandonar o esquema proposto na Figura 5.47 tendo-se optado por esquema mais simples que a seguir se descreve na secção 5.4.3.2.

5.4.3.2 Segunda abordagem

Visto que o microcontrolador não possui velocidade suficiente para executar, de uma forma contínua, todos os cálculos especificados na secção 5.4.3.1, optou-se por se simplificar o algoritmo de programação do modo representado na Figura 5.49.

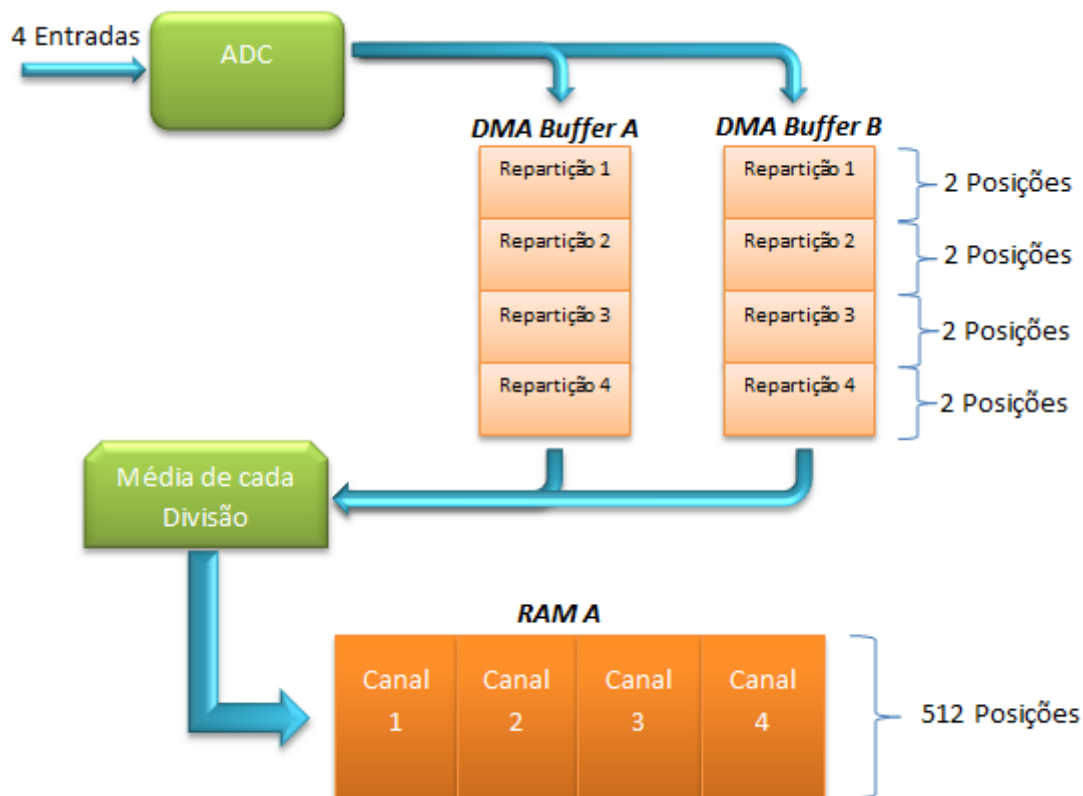


Figura 5.49 – Novo diagrama de aquisição de valores

O novo diagrama é similar ao apresentado na Figura 5.47, utilizando Buffers de DMA de 2 posições por repartição e diferindo por utilizar apenas um buffer de RAM para armazenar os valores recebidos pela chamada da rotina de serviço à interrupção do DMA que faz a filtragem da sobre amostragem.

Neste modo de funcionamento serão apenas analisados valores da energia à velocidade que o microcontrolador consegue processar.

5.4.3.2.1 Resultados obtidos pelo novo algoritmo

Começou-se por fazer-se a verificação do período de amostragem do sinal pela unidade ADC. Sempre que esta efetuava duas leituras a rotina de serviço à interrupção do DMA era chamada.

Para tal, associado à rotina de serviço à interrupção, foi colocado no kit um pino a comutar de valor lógico para indicar, através da leitura pelo osciloscópio, que esta tinha sido chamada. Esta comutação era efetuada à frequência de 6,4KHz.

De seguida verificou-se a codificação das amostras capturadas. Nas entradas da ADC foi aplicado um sinal com a frequência de 50 Hz, com amplitude de 1,65V e com offset de 1,65V tendo em vista verificar toda a gama de valores que a ADC lia. O resultado da captura das primeiras 128 amostras pode ser visualizado na Figura 5.50.

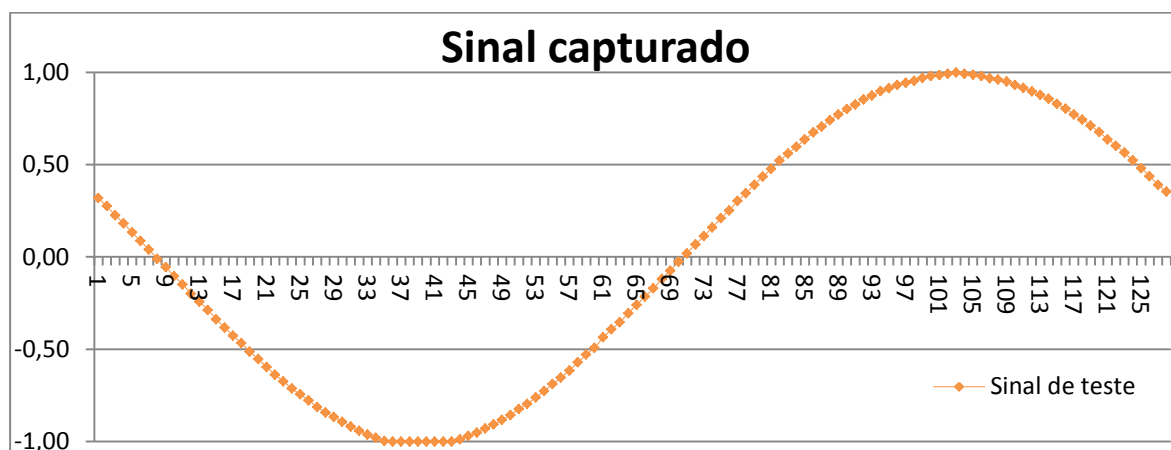


Figura 5.50 – Sinal capturado

No eixo vertical é representada a codificação no formato “*fractional Q1.15*” onde os valores da captura são normalizados a [-1; 1]. No eixo horizontal é apresentado o número da amostra capturada nesse período de aquisição.

Pode verificar-se que a codificação é efetuada corretamente e que para o número 128 amostras corresponde a um ciclo inteiro de onda, tal como tinha sido projetado na secção 3.2.

Feitas as primeiras verificações das capturas dos sinais, procedeu-se às verificações de cálculos dos parâmetros da energia elétrica a analisar. Como ponto de partida decidiu-se fazer a captura de quatro períodos do sinal da rede elétrica para cada canal, ou seja, de 512 valores (4x128 amostras) de acordo com a frequência de amostragem anteriormente definida.

Programaram-se as fórmulas apresentadas nas subsecções da fundamentação teórica (secção 3) para o cálculo dos diversos parâmetros dos sinais de entrada. Para efeitos de teste foi novamente aplicado um sinal de 50Hz, 1,65V de amplitude e 1,65V de *offset* em duas entradas da ADC (em que uma entrada representa os valores para a tensão e a outra uma entrada para os valores para a corrente). Estes sinais simulam uma carga puramente resistiva com a voltagem e corrente máxima permitida pelo analisador. O sinal capturado pode ser observado na Figura 5.51.

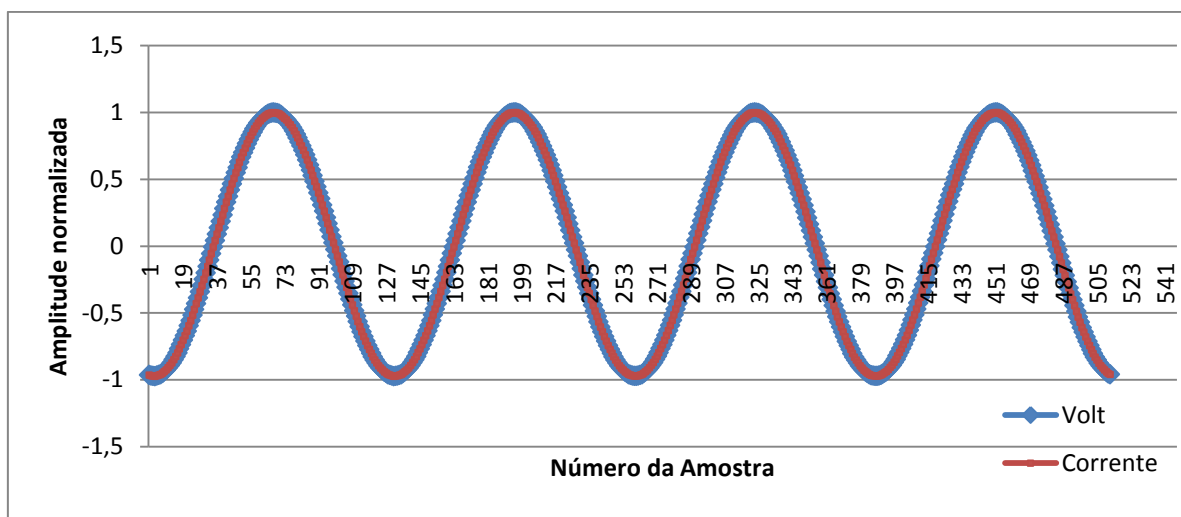


Figura 5.51 – Sinais idênticos capturados

Tendo em vista verificar se o cálculo dos valores efetuados estavam corretos e também se existiam erros significativos gerados por arredondamento devido ao facto do microcontrolador ter uma arquitetura de 16bits, paralelamente efetuou-se o mesmo cálculo utilizando o software Excel numa máquina de 32bits. A Tabela 5-4 mostra os valores obtidos pelo microprocessador assim como os valores calculados pelo *software* Excel. Também mostra os valores de erro absoluto e erro relativo entre os valores obtidos pelos dois processos.

Parâmetro	Valor obtido pelo Microcontrolador	Valor calculado no <i>software</i> Excel	Erro absoluto	Erro Relativo (%)
Irms (A)	3,4025037	3,402503888	1,87905E-07	5,52256E-06
Vrms (V)	296,77667	296,8111971	0,034527118	0,011632687
Imax (A)	4,8624845	4,862484497	3,2E-09	6,581E-08
Vmax (V)	424,16992	424,1699325	1,25E-05	2,94693E-06
Imin (A)	-4,7435393	-4,743539002	2,984E-07	6,29066E-06
Vmin (V)	-413,79395	-413,79394	1E-05	2,41666E-06
P. Aparente (VA)	1009,78375	1009,901252	0,117502168	0,011635016
P. Ativa (W)	1009,78345	1009,901252	0,117802168	0,011664721
P. Reativa (VAr)	0,7905694	0,778376618	0,012192782	1,566437282
Fator de Potência	0,9999997	1	3E-07	3E-05

Tabela 5-4 – Comparação de valores

Desta tabela pode concluir-se que os valores calculados através do kit estavam de acordo com o esperado e que os erros cometidos por arredondamento não são muito elevados, tendo sido registado o pior caso para o cálculo da potência reativa um erro relativo de aproximadamente 1,5% em relação ao calculado via Excel.

No que diz respeito ao cálculo da FFT e observação da magnitude das harmónicas que constituem o sinal, também se fez a comparação dos valores calculados pelo microcontrolador com os valores calculados pelo software Matlab. A Tabela 5-5 mostra esses resultados.

Harmónica	Frequência (Hz)	Magnitude calculado pelo microcontrolador (V)	Magnitude calculado pelo Matlab (V)
-----	0	0,20751953	7,6847
1	50	414,4165	419,7301
2	100	0	1,7009
3	150	0	0,4937
4	200	0	0,373
5	250	0	0,4784
6	300	0	0,2569
7	350	0	0,0342
8	400	0	0,1184
9	450	0	0,0256
10	500	0	0,0671
11	550	0	0,0519
12	600	0	0,0219
13	650	0	0,0837
14	700	0	0,0427
15	750	0	0,0534
16	800	0	0,0302
17	850	0	0,0352
18	900	0	0,017
19	950	0	0,0553
20	1000	0	0,0546
21	1050	0	0,0393
22	1100	0	0,0075
23	1150	0	0,0149
24	1200	0	0,023
25	1250	0	0,004

Tabela 5-5 – Magnitudes das harmónicas pelo cálculo da FFT

Esta tabela mostra que o cálculo da FFT foi efetuado corretamente pelo microcontrolador tendo sido discriminada a componente fundamental de 50Hz com a amplitude próxima do esperado embora a precisão não seja tão grande como à do cálculo pelo *software Matlab*. De salientar que a função que calcula a FFT para a família dsPIC está limitada a um máximo de 256 amostras.

Em outros testes a esta função da FFT, foi aumentada em múltiplos de 50Hz a frequência do sinal de entrada até à 25ª harmónica (1250Hz) tendo sido sempre observado o correto funcionamento desta função uma vez que mostrou sempre as magnitudes correspondentes para essas frequências, o que prova que o algoritmo de cálculo da FFT possui largura de banda suficiente para detetar até à 25ª harmónica.

Para este sinal, o cálculo da distorção harmónica (THD) e o cálculo para as percentagens relativas das harmónicas em relação à fundamental foi de zero pois não apresentava magnitude nas componentes das suas harmónicas visto ser uma senoide pura.

De modo a verificar o cálculo da distorção harmónica, aplicou-se numa entrada da ADC uma onda quadrada de 50Hz, de modo a obter-se um valor médio nulo nas leituras do sinal, para depois se efetuar o cálculo THD. Quando é aplicado a série de Fourier a este sinal e por este possuir “simetria ímpar em torno da origem e par em relação ao eixo vertical que passa pela abcissa $\pi/2$ é uma série de senos só com termos ímpares” [5]. O resultado deste ensaio pode ser observado na Figura 5.52, com recurso ao software Excel, onde se visualiza que existe apenas amplitude nas componentes harmónicas ímpares.

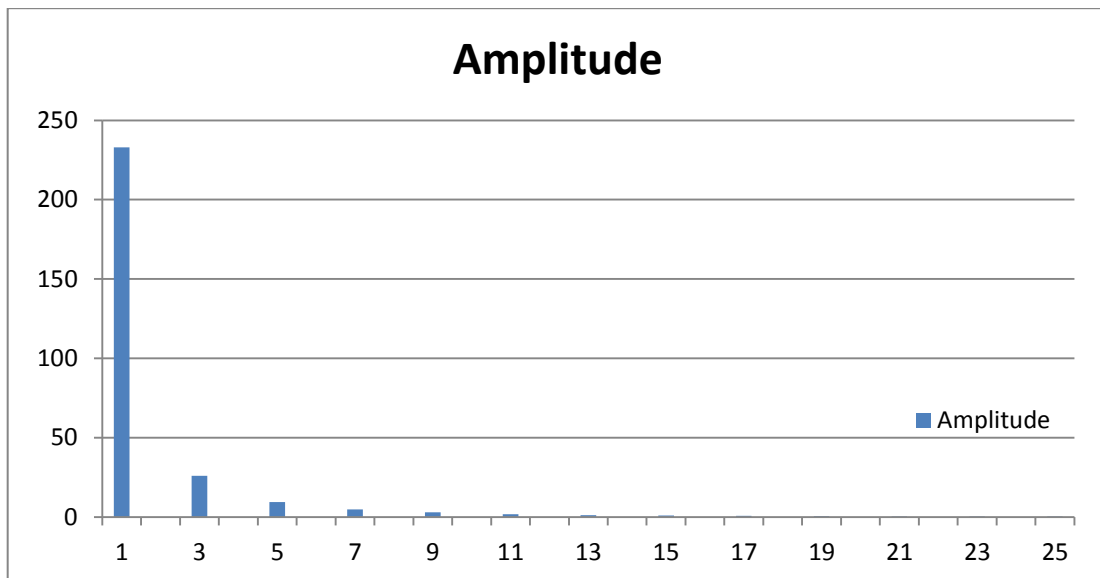


Figura 5.52 – Amplitude das harmónicas

Verificou-se assim que o cálculo da THD se apresenta correto, tendo para este ensaio o valor de $THD = 0,12128084$.

Para o cálculo da frequência do sinal de entrada, foi usado um algoritmo que detetava a passagem por zero. Sempre que se verifique uma mudança de sinal entre amostras consecutivas indica que existe uma passagem por zero. Essa troca de sinal é detetada pela multiplicação algébrica dos valores entre amostras consecutivas. Sempre que o valor for negativo ou igual a zero indica que está a ocorrer uma passagem por zero.

Primeiramente o algoritmo procura a primeira ocorrência de passagem por zero. De seguida, até chegar ao final das amostras, vai procurar a próxima passagem por zero e calcula a diferença temporal entre essa passagem por zero e a passagem que ocorreu anteriormente a essa. Tendo sido detetadas várias diferenças temporais que existiram entre passagens por zero, são somados todos esses tempos e divididos pelo número de intervalos temporais que ocorreram, ou seja, faz-se uma média dos tempos entre passagens por zero (T_{med}).

Assim, como este algoritmo tem o objetivo de calcular a frequência de uma onda sinusoidal, estes tempos correspondem a metade do período de onda da senoide. Desta maneira, para se ter o valor da frequência do sinal (f_{sinal}) será necessário fazer o inverso do dobro do tempo calculado, ou seja:

$$f_{sinal} = \frac{1}{2 * T_{med}} \quad (5-61)$$

De modo a calcular com maior precisão o tempo em que ocorre a passagem por zero entre amostras, recorreu-se ao uso de interpolação de primeira ordem entre nas amostras que detetam a passagem por zero. A linearização de primeira ordem é dada pela equação:

$$y = m \cdot x + b \quad (5-62)$$

Onde y é o valor do eixo das ordenadas que representa a amplitude, m declive da reta, x o eixo das abcissas que representa o tempo e b a ordenada na origem.

Como se pretende calcular o valor de x para o qual a amplitude em y é zero, pode-se manipular matematicamente a expressão do seguinte modo:

$$x = \frac{y - b}{m} = \frac{0 - b}{m} = -\frac{b}{m} \quad (5-63)$$

Uma vez que todas as amostras estão separadas igualmente pelo mesmo intervalo de tempo, fazendo corresponder a primeira amostra da deteção de passagem por zero à origem dos eixos das abcissas e a próxima amostra a uma unidade temporal de distância, então b e m serão dados por:

$$b = y(i) \quad (5-64)$$

$$m = \frac{y(i+1) - y(i)}{x(i+1) - x(i)} = \frac{y(i+1) - y(i)}{1 - 0} = y(i+1) - y(i) \quad (5-65)$$

Deste modo, o instante em que a amplitude passou por zero (t_{inst}) será dado pelo instante da primeira amostra somada da distância calculada em x , ou seja, é dado por:

$$t_{inst} = x(i) - \frac{y(i)}{y(i+1) - y(i)} * T_a \quad (5-66)$$

Sendo T_a é a período de amostragem do sinal.

Para se testar a funcionalidade deste algoritmo, foi aplicado a uma das entradas da ADC um sinal de 50Hz e elaborou-se a Tabela 5-6 que mostra as várias leituras obtidas pelas capturas assim como a média, valor máximo e mínimo obtido.

É possível observar que as leituras de frequência através do kit se apresentam com valores bastante próximos dos 50Hz aplicados.

Nº da Amostra	1	2	3	4	5	6	7	8	9	10
Frequência (Hz)	49,9 1581 7	50,1 7921 4	49,91 655	49,91 391	50,18 651	50,18 4036	49,92 4023	49,92 333	50,19 1704	49,92 272
Média (Hz)	50,0257814									
Valor Máximo (Hz)	50,191704									
Valor Mínimo (Hz)	49,91391									

Tabela 5-6 – Teste ao algoritmo de cálculo da frequência

Feitas as verificações ao software, procedeu-se à captura de valores usando a placa de aquisição. Fez-se passar corrente, provocada pelos 230Vrms da rede elétrica, na mesma resistência de 672Ω que foi utilizada para fazer os testes à placa de aquisição mostrada na secção 5.4.2.4. Fazendo a captura dos valores de corrente e tensão obteve-se os resultados gráficos, através do uso do software Excel, mostrados na Figura 5.53.

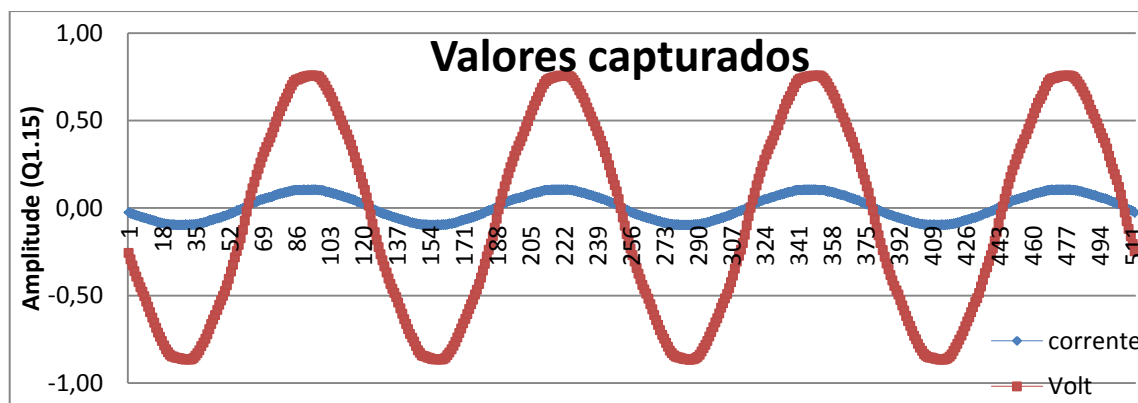


Figura 5.53 – Captura de corrente e tensão

Dos sinais capturados, elaborou-se a Tabela 5-7 com os resultados obtidos através do microcontrolador.

Parâmetro	Valor obtido pelo Microcontrolador
Irms (A)	0.3498108
Vrms (V)	237,06508
Imax (A)	0,4868432
Vmax (V)	321,2402
Imin (A)	-0,4872
Vmin (V)	-369,385
P. Aparente (VA)	82,927925
P. Ativa (W)	85,36194
P. Reativa (VAr)	NaN
Fator de Potência	1,0159937

Tabela 5-7 – Resultados obtidos da captura de valores pelo microcontrolador

Analisando os dados obtidos, verificou-se que é necessário proceder-se a calibrações do equipamento, pois, apesar de ter valores não muito distantes dos reais, não apresenta os valores corretamente. Valores como a potência reativa ter um valor superior à da potência aparente ou o fator de potência ter um valor superior a uma unidade, são valores incorretos.

6 Conclusões finais

Neste projeto existiram alguns entraves que originaram enormes perdas de tempo essenciais ao desenvolvimento do produto e conseqüentemente a uma finalização mais avançada do mesmo.

Inicialmente, como foi imposto pela empresa, o produto teria de ser desenvolvido através do kit de desenvolvimento Stellaris LM3S8962 (Apresentado na secção 5.2.1) correndo programação em LabVIEW. Com isto, foi necessário despender-se um tempo significativo com a aprendizagem da programação LabVIEW tendo em vista a sua implementação. Após várias tentativas concluiu-se que esta programação não servia para o desenvolvimento do projeto por esta ser demasiado lenta na aquisição dos dados necessários e assim se perderem informações importantes.

A insistência da gerência da SELMATRON em que se usasse a programação LabVIEW, levou a que se fizessem algumas tentativas de limpeza de código gerado pelo compilador do LabVIEW. Concluiu-se que esta limpeza de código era uma tarefa bastante complicada e pouco prática para futuros desenvolvimentos devido à complexidade do sistema além de que não resultaria uma vez que o sistemas operativo do LabVIEW atribui 'time-slots' a cada VI do LabVIEW o que implica que os programas de aquisição de dados sejam interrompidos e a informação perdida.

Tendo sido gastos cerca de cinco meses com a tentativa de programação LabVIEW e a respetiva solução em hardware (Solução multiplexada, secção 5.2.2), foi decidido abandonar este software e implementar os algoritmos em código em linguagem de programação C. Passado cerca de um mês conseguiram-se obter alguns dos resultados pretendidos ao nível de rapidez do sistema. Contudo, passado pouco tempo a unidade programadora *ULINK2* da *Keil* avariou. Esta avaria implicou algumas perdas de dados relativos ao desempenho da solução em causa. Devido ao elevado custo desta unidade foi decidido mudar-se para um kit de desenvolvimento mais barato.

Com o novo kit de desenvolvimento (*dsPICDEM 80*, secção 5.4.1) houve a necessidade de se criar e testar uma nova placa de aquisição e acondicionamento de sinal para se tirar melhor proveito das capacidades do kit, nomeadamente deste possuir duas ADC's internas que permitem a leitura de 8 canais de entrada simultaneamente.

Esta nova placa de aquisição, tal como a primeira, corresponderam perfeitamente às características esperadas de acondicionamento de sinal. Na primeira placa (secção 5.2.2) os sinais de entrada eram corretamente multiplexados quando eram geradas as trocas de informação nas entradas do *multiplexer*. No percurso dos sinais de ambas as placas também se verificou que as gamas de tensões nas diferentes secções das placas cumpriam com o que tinha sido projetado e simulado assim como as frequências de corte que estavam dentro dos valores previstos.

No que diz respeito ao *software* do segundo protótipo, conseguiu-se garantir a frequência de aquisição mínima desejada, ou seja, os 6,4KHz de todos os canais à entrada das ADC's. Houve a tentativa de se fazer uma sobre amostragem com 64 pontos por amostra, via recurso ao DMA, de modo a filtrar-se possíveis ruídos existentes nas entradas das ADC's. Contudo, devido a características do microcontrolador e estando este a operar à frequência máxima (40MHz),

apenas foi possível fazer a sobre amostragem ao dobro da frequência mínima sem ser possível tirar proveito máximo da memória dedicada ao DMA. Como o tempo dedicado à dissertação deste projeto já não era muito, decidiu-se poder-se fazer possíveis melhorias na sobre amostragem em trabalho futuro.

No desenvolvimento do software também se tentou fazer uma análise contínua da energia elétrica, ou seja, sem que o microcontrolador parasse a leitura das entradas das ADC's para efetuar os cálculos dos parâmetros. Para isso, usou-se dois buffers na memória RAM para cada ADC, de modo que fossem feitos os cálculos dos parâmetros necessários a analisar num dos buffers enquanto o outro estivesse a ser preenchido com novos valores. Contudo verificou-se que esta estratégia não era adequada uma vez que o CPU demorava mais tempo a fazer os cálculos num dos buffers do que o tempo necessário ao preenchimento do outro buffer. Visto isto, mudou-se para uma nova solução mais simplificada. Essa solução passou por fazer a análise da energia da rede elétrica à rapidez do microcontrolador, ou seja, eram armazenados valores apenas para um buffer na memória RAM e seguidamente interrompia-se as leituras das ADC's para se fazer a análise dos parâmetros dos valores desse buffer. Terminados esses cálculos da análise de parâmetros, eram efetuadas novas leituras para uma outra análise e assim sucessivamente.

Foram elaborados todos os algoritmos necessários para o cumprimento dos objetivos propostos de análises das características da energia da rede elétrica, ou seja, foram elaborados algoritmos de deteção de valores de tensão e corrente máximos e mínimos, valores do verdadeiro valor RMS (TRMS), valores de potências aparente, ativa e reativa, valor de fator de potência, valor de distorção harmónica, percentagens relativas das componentes harmónicas, frequência do sinal da rede elétrica e deteção do sentido de fluxo de energia colocando à saída um pino com essa informação digital.

Com estes resultados é possível fazer grande parte das avaliações acerca do cumprimento ou não da qualidade mínima de energia regulamentada pela norma EN 50160 apresentada na Secção 4.

Após se proceder à montagem do kit de desenvolvimento com a placa de aquisição e acondicionamento de sinal, verificou-se que o equipamento necessita de ser calibrado de modo a avaliar corretamente os parâmetros da rede elétrica.

Finalizando, pode-se concluir que com este trabalho foram adquiridos conhecimentos em diversas áreas.

O estudo da energia possibilitou também a melhor compreensão e as implicações dos diversos tipos de potência (Potência aparente, ativa e reativa) assim como outros parâmetros essenciais à sua caracterização tal como fator de potências e distorção harmónica.

A aprendizagem de uma nova e diferente tipo de linguagem de programação (LabVIEW, linguagem de programação gráfica e concorrente) constituiu uma mais-valia em possíveis trabalhos futuros que se possam vir a realizar.

O facto de esta dissertação ter sido realizada no âmbito empresarial foi bastante vantajoso uma vez que deu para começar a perceber muitos aspetos da dinâmica interna da empresa. O acesso a algum apoio técnico por parte de excelentes profissionais e o acesso às ferramentas disponíveis na empresa, tal como por exemplo máquinas de construção de protótipos de PCB's, foram aspetos bastante positivos no desenvolvimento do projeto.

A construção dos protótipos apresentados nesta dissertação nestas máquinas, assim como o seu desenho e simulação nas ferramentas de *software MultiSim* e *Ultiboard*, também foram uma novidade e uma parte bastante entusiasmante. Implicou adquirir alguns conhecimentos acerca dos encapsulamentos dos componentes assim como dar a entender a enorme variedade das características dos diversos componentes.

O facto de se ter começado este projeto pela raiz, proporcionou uma outra visão, sensibilidade e experiência acerca de aspetos importantes a se considerar no começo do projeto e nas decisões a serem tomadas ao longo do seu desenvolvimento.

7 Trabalho futuro

Não tendo sido possível a conclusão total do projeto no tempo que se dispôs, torna-se necessário a continuação do desenvolvimento em vários aspetos, nomeadamente os seguintes:

- Replicação de código para a segunda ADC
- Otimização dos cálculos e resultados efetuados por *software*
- Calibração
- Introdução de uma interface gráfica com o utilizador
- Encapsulamento de todo o equipamento
- Continuação do desenvolvimento de software de modo a registar e alertar quando algum parâmetro de qualidade de energia não cumprir com as normas EN 50160
- Certificação do produto

Deste modo ter-se-ia um produto pronto para a sua introdução e comercialização no mercado de analisadores de energia elétrica.

8 Bibliografia

- [1] "EN 50160 - Voltage Characteristics of Public Distribution Systems," 2004.
- [2] "GE Energi Management," [Online]. Available: <http://www.geindustrial.com.br/produtos/medidores/mtplus/02.asp>.
- [3] "FLUKE," [Online]. Available: <http://en-us.fluke.com/products/power-quality-tools/fluke-434-series-ii-power-quality.html#overview>.
- [4] "TestEquity," [Online]. Available: <http://www.testequity.com/products/3728/>.
- [5] A. P. d. Melo, teoria dos sistemas de controlo lineares, Aveiro, 2010.
- [6] K. Duda, "Interpolation Algorithms of DFT for Parameters Estimation of Sinusoidal and Damped Sinusoidal Signals".
- [7] A. M. Rodrigues, "INFLUÊNCIA DA DISTORÇÃO HARMÔNICA E DE DESEQUILÍBRIOS EM INFLUÊNCIA DA DISTORÇÃO HARMÔNICA E DE DESEQUILÍBRIOS EM," Universidade Federal do Paraná.
- [8] A. M. HULAK e M. F. ALVES, "ON THE ANALYSIS OF (UN)TRUE ROOT MEAN SQUARE MEASUREMENT," Instituto de Engenharia do Porto.
- [9] "oakland," [Online]. Available: http://www.secs.oakland.edu/~frick/EE4220-EM_Dynamics/TW_Chapter_7.pdf.
- [10] "Materia do Curso," [Online]. Available: <http://materiadocurso.blogspot.pt/2012/12/o-que-e-fator-potencia-fp-potencia.html>.
- [11] ElectronicsTutorials. [Online]. Available: <http://www.electronicstutorials.ws/transformer/current-transformer.html>.
- [12] "Wikipédia," [Online]. Available: http://pt.wikipedia.org/wiki/Filtro_passa-baixo.
- [13] "Wikipedia," [Online]. Available: <http://en.wikipedia.org/wiki/Multiplexer>.
- [14] Wikipédia. [Online]. Available: [http://pt.wikipedia.org/wiki/Buffer_\(eletr%C3%B4nica\)](http://pt.wikipedia.org/wiki/Buffer_(eletr%C3%B4nica)).
- [15] P. Fonseca, Sistemas de Instrumentação Electrónica, Aveiro, 2011/2012.
- [16] "Datasheet," [Online]. Available: <http://datasheet.octopart.com/TL431ACLP.-Texas-Instruments-datasheet-7273310.pdf>.

[17] "Analog Devices," [Online]. Available: http://www.analog.com/static/imported-files/data_sheets/AD7908_7918_7928.pdf.

[18] "Analog Devices," [Online]. Available: http://www.analog.com/static/imported-files/rarely_asked_questions/moreInfo_raq_resistors.html.

[19] N. Instruments. [Online]. Available: <http://www.ni.com/labview/why/pt/>.

Anexos:

Anexo A – Código C do primeiro protótipo

```
#include "inc/hw_memmap.h"
#include "inc/hw_ssi.h"
#include "inc/hw_types.h"
#include "driverlib/ssi.h"
#include "driverlib/gpio.h"
#include "driverlib/sysctl.h"
#include "driverlib/debug.h"
#include "driverlib/sysctl.h"
#include "utils/uartstdio.c"
#include "boards/ek-lm3s8962/drivers/rit128x96x4.h"
#include "stdio.h"
#include "inc/lm3s8962.h"
#include <stdint.h>
#include <stdlib.h>
#include "math.h"
#include "arm_math.h"
#define NUM_SSI_DATA 512 // tamanho do array de entrada de cada canal (bytes/2)
#define NUM_CANAIS 7 // numero de canais
#define NUM_OP (NUM_CANAIS * NUM_SSI_DATA) // numero de operaçoes a fazer na funcao
"captura"
#define MUX_PIN_MASK 0x43 // PORTA_BASE 0100 0011 (pin 0, 1 e 6 activos)
#ifdef DEBUG
void
__error__(char *pcFilename, unsigned long ulLine){
}
#endif
void calibrar(float32_t ulDataTx, float32_t *Vp){
    unsigned int i; unsigned long ulDataRx;
    GPIOPinWrite(GPIO_PORTA_BASE, MUX_PIN_MASK, 0x43); // seleciona o ultimo canal do
mux (referencia)
    for(i=0; i<2; i++){
        {
            SSIDataPut(SSIO_BASE, ulDataTx);
            while(!SSIBusy(SSIO_BASE)) {
            }
            SSIDataGet(SSIO_BASE, &ulDataRx);
        }
        *Vp = (ulDataRx & 0x0FFF); // dados
    }
}
void captura(unsigned long ulDataTx, float32_t *V1, float32_t *V2, float32_t *V3, float32_t *N, float32_t
*I1, float32_t *I2, float32_t *I3 ){
    volatile int i; volatile int iMux; // variavel contadora da atribuicao do canal
    volatile unsigned long sel; // variavel de selecao de canal
    unsigned long ulDataRx; unsigned long data;
    volatile unsigned int iV1=0, iV2=0, iV3=0, iN=0, iI1=0, iI2=0, iI3=0; // inicializacao das variaveis de posicao
cada vector
    unsigned char muxTab[] = {0x00, 0x01, 0x02, 0x03, 0x40, 0x41, 0x42};
    for(i = 0, iMux = 0; i < NUM_OP; i++, iMux++) {
```

```

// fazer a seleção do canal do multiplexer
if(iMux > 6) // caso chegue ao ultimo canal de leitura
    iMux = 0; // reinicia a contagem para o primeiro canal
    GPIOPinWrite(GPIO_PORTA_BASE, MUX_PIN_MASK, muxTab[iMux]); //
muxTab[iMux] escreve o canal que se pretende ler

SSIDataPut(SSI0_BASE, ulDataTx);
// Wait until SSI0 is done transferring all the data in the transmit FIFO.
while(!SSIBusy(SSI0_BASE)) {
}
SSIDataGet(SSI0_BASE, &ulDataRx);
data = (ulDataRx & 0x0FFF); // dados
if(iMux == 1){
    V1[iV1] = (float32_t) data;
    iV1++;
}
else if( iMux == 2) {
    V2[iV2] = (float32_t) data;
    iV2++;
}
else if( iMux == 3) {
    V3[iV3] = (float32_t) data;
    iV3++;
}
else if( iMux == 4) {
    N[iN] = (float32_t) data;
    iN++;
}
else if( iMux == 5) {
    I1[iI1] = (float32_t) data;
    iI1++;
}
else if( iMux == 6) {
    I2[iI2] = (float32_t) data;
    iI2++;
}
else if( iMux == 0) {
    I3[iI3] = (float32_t) data;
    iI3++;
}
}
}
int main(void){
    int i; // variavel contadora
    int j, k; // variavel de teste simular tempo (apagar depois)
    unsigned long ulDataTx;
    float32_t parc; // variavel auxiliar para o calculo de PotAct
// inicialização do array dos canais a ler
float32_t V1[NUM_SSI_DATA]; // tensao da fase 1
float32_t V2[NUM_SSI_DATA]; // tensao da fase 2
float32_t V3[NUM_SSI_DATA]; // tensao da fase 3
float32_t N[NUM_SSI_DATA]; // tensao do neutro
float32_t I1[NUM_SSI_DATA]; // corrente da fase 1
float32_t I2[NUM_SSI_DATA]; // corrente da fase 2
float32_t I3[NUM_SSI_DATA]; // corrente da fase 3
float32_t testOutput[NUM_SSI_DATA];

```



```

float32_t V1max, V2max, V3max, Nmax, I1max, I2max, I3max; // maximo valor no array de
amostra
float32_t V1min, V2min, V3min, Nmin, I1min, I2min, I3min; // minimo valor no array de amostra
uint32_t V1maxInd, V2maxInd, V3maxInd, NmaxInd, I1maxInd, I2maxInd, I3maxInd; // index do
maximo valor no array de amostra
uint32_t V1minInd, V2minInd, V3minInd, NminInd, I1minInd, I2minInd, I3minInd; // index do minimo
valor no array de amostra
float32_t V1rms, V2rms, V3rms, Nrms, I1rms, I2rms, I3rms; // RMS (Root mean square) para cada
canal
float32_t V1rmsReg[300], NrmsReg[300]; // Registo de RMS
float32_t V1mediaRms, NmediaRms;
float32_t PotAparC1, PotAparC2, PotAparC3; // Potencia aparente para canal Cx
float32_t PotActC1, PotActC2, PotActC3; // Potencia activa para cada canal Cx
float32_t PotReactC1, PotReactC2, PotReactC3; // Potencia Reactiva para cada canal Cx
float32_t FPontC1, FPontC2, FPontC3; // factor de potencia do canal Cx
// -- ADC e Converções --
float32_t Vp = 230*sqrt(2)*1.3;
float32_t delta = 2*Vp/(4096); // 2^NBits = 2^12 = 4096
unsigned long * pUIDataRx;
pUIDataRx = (unsigned long*) calloc (NUM_SSI_DATA, sizeof(unsigned long)); // reserva
espaço inicializado com 0
SysCtlClockSet(SYSCTL_SYSDIV_1 | SYSCTL_USE_OSC | SYSCTL_OSC_MAIN |
SYSCTL_XTAL_8MHZ);
// The SSI0 peripheral must be enabled for use.
SysCtlPeripheralEnable(SYSCTL_PERIPH_SSI0);
SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
/* Configure Status pin A0, A1, A6 as output */
GPIOPADConfigSet(GPIO_PORTA_BASE, MUX_PIN_MASK, GPIO_STRENGTH_2MA,
GPIO_PIN_TYPE_STD);
GPIODirModeSet (GPIO_PORTA_BASE, MUX_PIN_MASK, GPIO_DIR_MODE_OUT);
GPIOPinConfigure(GPIO_PA2_SSI0CLK);
GPIOPinConfigure(GPIO_PA3_SSI0FSS);
GPIOPinConfigure(GPIO_PA4_SSI0RX);
GPIOPinConfigure(GPIO_PA5_SSI0TX);

GPIOPinTypeSSI(GPIO_PORTA_BASE, GPIO_PIN_5 | GPIO_PIN_4 | GPIO_PIN_3 |
GPIO_PIN_2);

SSIConfigSetExpClk(SSI0_BASE, SysCtlClockGet(), SSI_FRF_MOTO_MODE_0,
SSI_MODE_MASTER, 3000000, 16);

SSIEnable(SSI0_BASE);

while(SSIDataGetNonBlocking(SSI0_BASE, pUIDataRx) // &ulDataRx[0]){
}
free(pUIDataRx); // liberta o espaço de memoria
ulDataTx = 0x8310; // 0x8310 - modo canal 0 da ADC(0V a 5V); 0xFFFF0 - modo 8 canais(NAO
TESTADO)
j=0;
while(1){
//----- captura da ADC-----
captura(ulDataTx, V1, V2, V3, N, I1, I2, I3);
// converção para valores reais
for(i=0; i<NUM_SSI_DATA; i++){
V1[i] = -Vp + delta*V1[i];
V2[i] = -Vp + delta*V2[i];
V3[i] = -Vp + delta*V3[i];
N[i] = -Vp + delta*N[i];
}
}

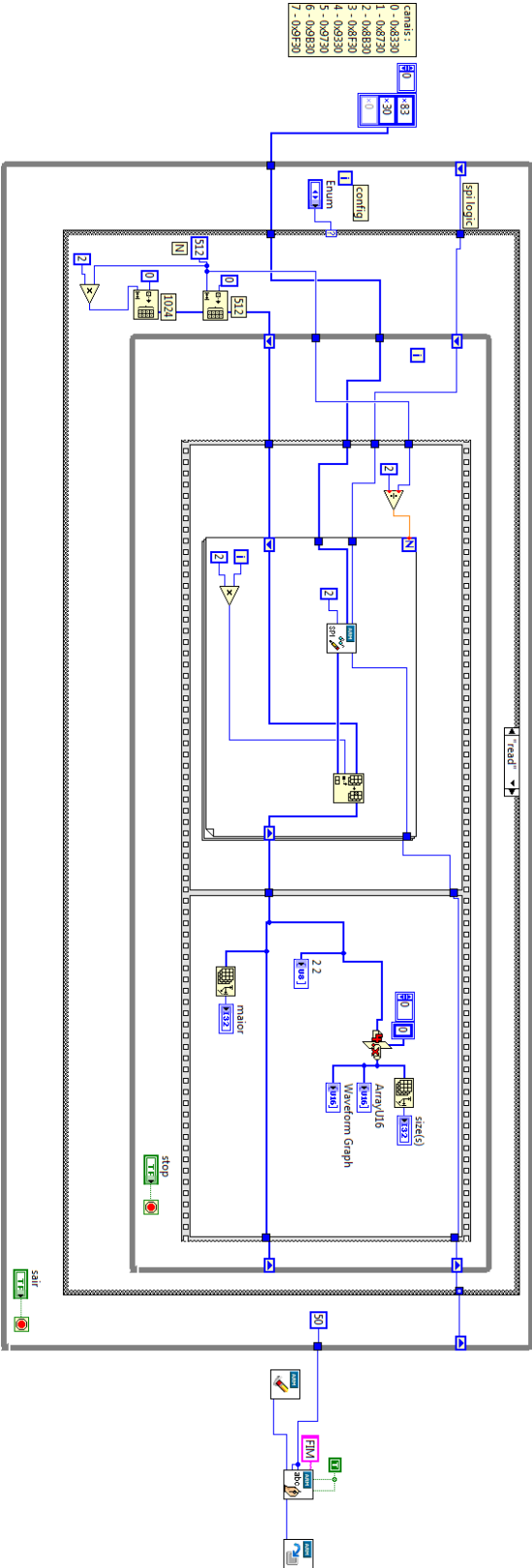
```

```

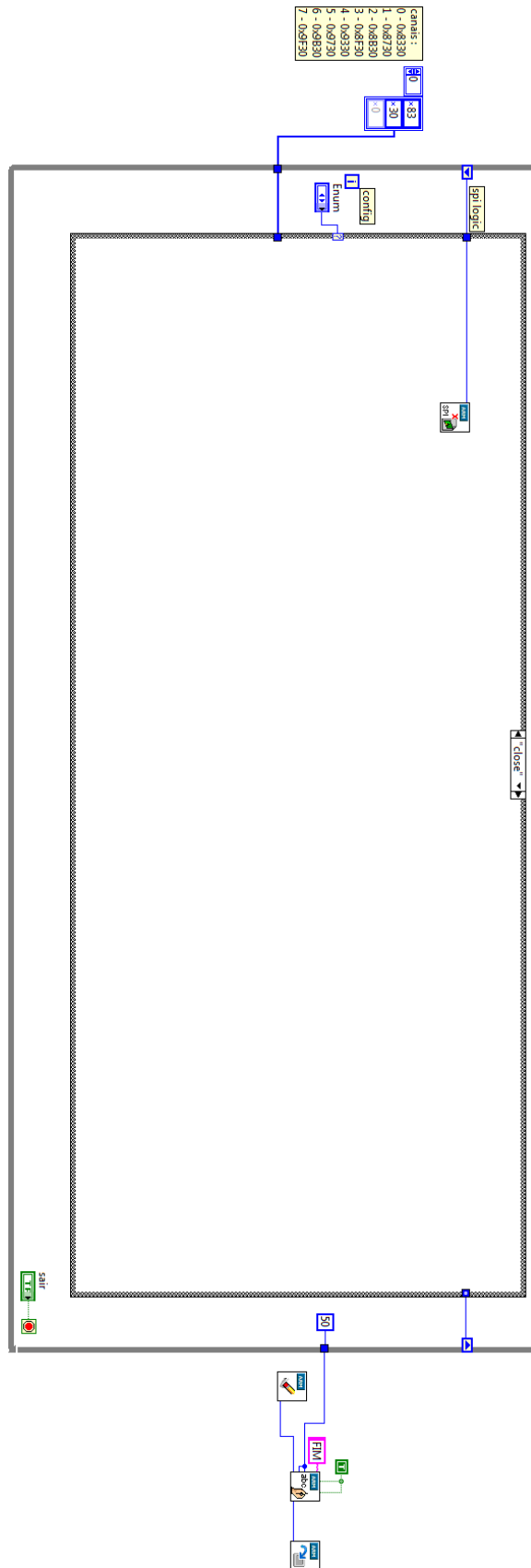
        I1[i]=-Vp + delta*I1[i];
        I2[i]=-Vp + delta*I2[i];
        I3[i]=-Vp + delta*I3[i];
    }
// ----- calculo do maximo e minimo -----
arm_max_f32(V1,NUM_SSI_DATA, &V1max, &V1maxInd);
arm_max_f32(V2,NUM_SSI_DATA, &V2max, &V2maxInd);
arm_max_f32(V3,NUM_SSI_DATA, &V3max, &V3maxInd);
arm_max_f32(N,NUM_SSI_DATA, &Nmax, &NmaxInd);
arm_max_f32(I1,NUM_SSI_DATA, &I1max, &I1maxInd);
arm_max_f32(I2,NUM_SSI_DATA, &I2max, &I2maxInd);
arm_max_f32(I3,NUM_SSI_DATA, &I3max, &I3maxInd);
arm_min_f32(V1,NUM_SSI_DATA, &V1min, &V1minInd);
arm_min_f32(V2,NUM_SSI_DATA, &V2min, &V2minInd);
arm_min_f32(V3,NUM_SSI_DATA, &V3min, &V3minInd);
arm_min_f32(N,NUM_SSI_DATA, &Nmin, &NminInd);
arm_min_f32(I1,NUM_SSI_DATA, &I1min, &I1minInd);
arm_min_f32(I2,NUM_SSI_DATA, &I2min, &I2minInd);
arm_min_f32(I3,NUM_SSI_DATA, &I3min, &I3minInd);
// ----- RMS de tensao e de corrente-----
arm_rms_f32(V1,NUM_SSI_DATA, &V1rms);
arm_rms_f32(V2,NUM_SSI_DATA, &V2rms);
arm_rms_f32(V3,NUM_SSI_DATA, &V3rms);
arm_rms_f32(N,NUM_SSI_DATA, &Nrms);
arm_rms_f32(I1,NUM_SSI_DATA, &I1rms);
arm_rms_f32(I2,NUM_SSI_DATA, &I2rms);
arm_rms_f32(I3,NUM_SSI_DATA, &I3rms);
V1rmsReg[j]=V1rms;           // guarda no registo para depois se fazer a media no tempo
necessario
NrmsReg[j]=Nrms;
// ----- Potencia aparente -----
PotAparC1 = V1rms*I1rms;
PotAparC2 = V2rms*I2rms;
PotAparC3 = V3rms*I3rms;
// ----- Potencia activa -----
arm_dot_prod_f32(V1, I1, NUM_SSI_DATA, &parc);    // The vectors are multiplied element-
by-element and then summed
PotActC1=parc/NUM_SSI_DATA;
arm_dot_prod_f32(V2, I2, NUM_SSI_DATA, &parc);    // The vectors are multiplied element-
by-element and then summed
PotActC2=parc/NUM_SSI_DATA;
arm_dot_prod_f32(V3, I3, NUM_SSI_DATA, &parc);    // The vectors are multiplied element-
by-element and then summed
PotActC3=parc/NUM_SSI_DATA;
// ----- Potencia reativa -----
PotReactC1 = sqrt(PotAparC1*PotAparC1 - PotActC1*PotActC1);
PotReactC2 = sqrt(PotAparC2*PotAparC2 - PotActC2*PotActC2);
PotReactC3 = sqrt(PotAparC3*PotAparC3 - PotActC3*PotActC3);
// ----- Factor de Potencia cos(fi) -----
FPontC1 = PotAparC1/PotActC1;
FPontC2 = PotAparC2/PotActC2;
FPontC3 = PotAparC3/PotActC3;
}
return(0);
}

```


Captura de valores:



Fecho da comunicação SPI:



Anexo C – Simulação de cargas e potências (Matlab)

```
clear all, clc, close all
t = 0:0.00001:40E-3;f1=50;desf=-pi/2;
tensao = 230*sqrt(2)*sin(2*pi*f1*t);
corrente = 5*sqrt(2)*sin(2*pi*f1*t + desf);
potencia = tensao.*corrente;
subplot(3,1,1);plot(t,tensao);xlabel('tempo [s]');ylabel('Tensão [V]');
grid on;subplot(3,1,2);plot(t,corrente);xlabel('tempo [s]');
ylabel('Corrente [A]');grid on;subplot(3,1,3);plot(t,potencia);
xlabel('tempo [s]');ylabel('Potência [VA]');grid on;
```

Anexo D – Código em linguagem C do segundo protótipo

```
#include "general.h"
/* Macros for Configuration Fuse Registers */
_FGS(GWRP_OFF & GSS_OFF & GCP_OFF);
_FOSSEL(FNOSC_FRC & IESO_ON);
_FOSC(POSCMD_NONE & OSCIOFNC_OFF & FCKSM_CSECMD);
_FWDT(PLLKEN_ON & FWDTEN_OFF);
_FICD(JTAGEN_ON & ICS_PGD2);
//void InitUART1(void);
//void SendString(char* str);
void TrueRMS(fractional * vector, float conversion, float * rms);
void calcFFT(fractional * p_data, float * magArray, float * magMax, int * peakFreqMaxBin, float *
peakFreqMax);
void interpolation2Points( fractional * p_data, float * resultTime);
//Max values
int VMaxIndRB7, VMaxIndRB3, VMaxIndRB4, VMaxIndRB5; // index of VMax
fractional VMaxFracRB7, VMaxFracRB3, VMaxFracRB4, VMaxFracRB5; // VMax fractional
float VMaxFltRB7, VMaxFltRB3, VMaxFltRB4, VMaxFltRB5; // Vmax Float
//Min Values
int VMinIndRB7, VMinIndRB3, VMinIndRB4, VMinIndRB5; // index of VMax
fractional VMinFracRB7, VMinFracRB3, VMinFracRB4, VMinFracRB5; // VMax fractional
float VMinFltRB7, VMinFltRB3, VMinFltRB4, VMinFltRB5; // Vmax Float
//RMS Values
float VrmsRB7,IrmsRB3,IrmsRB5;
// Apparent Power
float ApPwrL1, ApPwrL2, ApPwrL3;
// Real Power
float RealPwrL1, RealPwrL2, RealPwrL3;
// Reactive Power
float ReactPwrL1, ReactPwrL2, ReactPwrL3;
//Power factor
float PowerFactorL1, PowerFactorL2, PowerFactorL3;
// pointers to RAM Buffer to analyze
fractional *p_Ram_RB7, *p_Ram_RB3, *p_Ram_RB4, *p_Ram_RB5;
float auxFlt,auxFlt2; fractional auxFrac,auxFrac2; float a1,b2; fractional a2;
float RAM_BufferA_VER_VOLT1 [RAM_NUMSAMP_PER_CH];
float RAM_BufferA_VER_VOLT2 [RAM_NUMSAMP_PER_CH];
float RAM_BufferA_VER_CURR [RAM_NUMSAMP_PER_CH];
float RAM_BufferA_VER_NEUTRO [RAM_NUMSAMP_PER_CH];
fractional VectorAuxFrac [RAM_NUMSAMP_PER_CH]; float VectorAuxFlt
[RAM_NUMSAMP_PER_CH];
```

```

fractcomplex sigCmpx[RAM_NUMSAMP_PER_CH/2] __attribute__
((space(ymemory),aligned(RAM_NUMSAMP_PER_CH)));
fractcomplex twiddleFactors[RAM_NUMSAMP_PER_CH/2/2] /* Declare Twiddle Factor array in X-
space*/
__attribute__ ((space(xmemory),far,aligned(RAM_NUMSAMP_PER_CH)));
int peakFrequencyBin; /* Declare post-FFT variables to compute the */
float peakFrequencyFlt; unsigned long peakFrequency; fractional *p_real; fractcomplex *p_cmpx;
float magArray[HARMONIC_NUM+1]; // array with all 25 harmonic (25h + 1(0Hz))
float harmPerc[HARMONIC_NUM]; float peakFreqMax,magMax;
int peakFreqMaxBin; float THD; float t1,t2,count; float TempMed; float frequency; int t1bool;
int main(void) {
// Configure Oscillator to operate the device at 40Mhz
// Fosc= Fin*M/(N1*N2), Fcy=Fosc/2
// Fosc= 8M*40/(2*2)=80Mhz for 8M input clock
//OSCCONbits.NOSC = 0b001;
// PLLFBD=38; // M=40
// CLKDIVbits.PLLPOST=0; // N1=2
// CLKDIVbits.PLLPRE=0; // N2=2
// OSCTUN=18; // Tune FRC oscillator, if FRC is used
// Disable Watch Dog Timer
// RCONbits.SWDTEN=0;
// clock switching to incorporate PLL
// __builtin_write_OSCCONH(0x01); // Initiate Clock Switch to Primary Oscillator
with PLL (NOSC=0x03)
// __builtin_write_OSCCONL(OSCCON || 0x01); // Start clock switching
while (OSCCONbits.COSC != 0x01); // Wait for Clock switch to occur
while(OSCCONbits.LOCK!=1); // Wait for PLL to lock
SetupPorts(); // Initialize I/O
StartProg();
// initAdc1M(); // Initialize ADC1 to convert Channel 0
initAdc2M(); // Initialize ADC2 to convert Channel 0
initDmaM(); // Initialise DMA
// initTmr3();
initTmr5();
// InitUART1();
dummyWrite();
//-----FFT-----
#ifndef FFTWIDCOEFFS_IN_PROGMEM /* Generate TwiddleFactor Coefficients */
TwidFactorInit (LOG2_BLOCK_LENGTH-1, &twiddleFactors[0], 0);
#endif
//-----
// IEC0bits.AD1IE = 1; //0; // Enable ADC1 interrupts
IEC1bits.AD2IE = 1; //0; // Enable ADC2 interrupt
// AD1CON1bits.ADON = 1; // Turn on the A/D 1 converter
AD2CON1bits.ADON = 1; // Turn on the A/D 2 converter
//Start Timer 3 and 5
// T3CONbits.TON = 1;
T5CONbits.TON = 1;
// //Enable DMA channels Ch0 and Ch1 interrupts
IEC0bits.DMA0IE = 1;
// IEC0bits.DMA1IE = 1;
volatile int i;
while (1) {
if(RamFlag == 1) {
// look witch Buffer RAM is able to analyze
p_Ram_RB7 = &RAM_Buffer_RB7[0]; p_Ram_RB3 = &RAM_Buffer_RB3[0];
p_Ram_RB4 = &RAM_Buffer_RB4[0]; p_Ram_RB5 = &RAM_Buffer_RB5[0];
// remove the diferece voltage between GND and Neutral(RB3) of voltages

```

```

auxFrac = Float2Fract(40.0 / (float) CONVERSION_VOLT);
for(i = 0; i < RAM_NUMSAMP_PER_CH ; i++)
    VectorAuxFrac[i] = auxFrac;
VectorSubtract (RAM_NUMSAMP_PER_CH, p_Ram_RB7, p_Ram_RB7, &VectorAuxFrac[0] );
VectorSubtract (RAM_NUMSAMP_PER_CH, p_Ram_RB5, p_Ram_RB5, &VectorAuxFrac[0]);
//----- TRUE RMS Values -----)
TrueRMS(p_Ram_RB7, CONVERSION_VOLT, &VrmsRB7);
TrueRMS(p_Ram_RB3, CONVERSION_CURRENT, &IrmsRB3);
TrueRMS(p_Ram_RB5, CONVERSION_CURRENT, &IrmsRB5);
//TrueRMS(p_Ram_RB4, CONVERSION_VOLT, &VrmsRB4);
VrmsRB7 = VrmsRB7 + CORRECTION_VOLT;
IrmsRB3 = IrmsRB3 + CORRECTION_CURR;
IrmsRB5 = IrmsRB5 + CORRECTION_CURR;
//-----)
for(i=0;i<RAM_NUMSAMP_PER_CH;i++){
    RAM_BufferA_VER_VOLT1[i] = Fract2Float( p_Ram_RB7[i] );
    //RAM_BufferA_VER_VOLT2[i] = Fract2Float( p_Ram_RB4[i] );
    RAM_BufferA_VER_CURR[i] = Fract2Float( p_Ram_RB5[i] );
    RAM_BufferA_VER_NEUTRO[i] = Fract2Float( p_Ram_RB3[i] );
}
//--- Analyze data ---
// Max Values
VMaxFltRB7 = Fract2Float( VectorMax(RAM_NUMSAMP_PER_CH, p_Ram_RB7,
&VMaxIndRB7)) * CONVERSION_VOLT;
VMaxFltRB3 = Fract2Float( VectorMax(RAM_NUMSAMP_PER_CH, p_Ram_RB3,
&VMaxIndRB3)) * CONVERSION_CURRENT;
// VMaxFltRB4 = Fract2Float( VectorMax(RAM_NUMSAMP_PER_CH, p_Ram_RB4,
&VMaxIndRB4)) * CONVERSION_VOLT;
VMaxFltRB5 = Fract2Float( VectorMax(RAM_NUMSAMP_PER_CH, p_Ram_RB5,
&VMaxIndRB5)) * CONVERSION_CURRENT;
// Min Values
VMinFltRB7 = Fract2Float( VectorMin(RAM_NUMSAMP_PER_CH, p_Ram_RB7,
&VMinIndRB7)) * CONVERSION_VOLT;
VMinFltRB3 = Fract2Float( VectorMin(RAM_NUMSAMP_PER_CH, p_Ram_RB3,
&VMinIndRB3)) * CONVERSION_CURRENT;
// VMinFltRB4 = Fract2Float( VectorMin(RAM_NUMSAMP_PER_CH, p_Ram_RB4,
&VMinIndRB4)) * CONVERSION_VOLT;
VMinFltRB5 = Fract2Float( VectorMin(RAM_NUMSAMP_PER_CH, p_Ram_RB5,
&VMinIndRB5)) * CONVERSION_CURRENT;

// Apparent Power
ApPwrL3 = VrmsRB7 * IrmsRB5; // Line 3
// Real Power
for(i = 0, auxFlt = 0; i < RAM_NUMSAMP_PER_CH; i++)
    auxFlt = auxFlt + ((Fract2Float( p_Ram_RB7[i] ) * CONVERSION_VOLT) +
CORRECTION_VOLT) * ((Fract2Float( p_Ram_RB5[i] ) * CONVERSION_CURRENT));
RealPwrL3 = auxFlt / ( (float) RAM_NUMSAMP_PER_CH );
// See if power entry or out
if( RealPwrL3 < 0.0) // if power in
    LED1 = 1;
else // if power out
    LED1 = 0;
//Reactive Power
ReactPwrL1 = sqrtf( ApPwrL1 * ApPwrL1 - RealPwrL1 * RealPwrL1);
ReactPwrL2 = sqrtf( ApPwrL2 * ApPwrL2 - RealPwrL2 * RealPwrL2);
ReactPwrL3 = sqrtf( ApPwrL3 * ApPwrL3 - RealPwrL3 * RealPwrL3);
// Power Factor
PowerFactorL1 = RealPwrL1 / ApPwrL1;

```



```

PowerFactorL2 = RealPwrL2 / ApPwrL2;
PowerFactorL3 = RealPwrL3 / ApPwrL3;
//Frequency
for(i = 0, TempMed=0, count=0, t1bool = 0; i < RAM_NUMSAMP_PER_CH-1 ; i++){
    VectorMultiply (1,&auxFrac,&p_Ram_RB7[i],&p_Ram_RB7[i+1]);
    if(auxFrac <= 0) { // two sample with diferente signal
        if(t1bool == 0) {
            interpolation2Points(&p_Ram_RB7[i],&auxFlt);
            t1 = i * PERIOD_SAMPLING_RATE + auxFlt;
            t1bool = 1;
        }
        else{
            if(p_Ram_RB7[i+1] == 0) { // in case of zero value
                t2 = (i+1) * PERIOD_SAMPLING_RATE; // get the time os zero sample
                i++; // for jump the sample with 0 value
            }
            else{
                interpolation2Points(&p_Ram_RB7[i],&auxFlt);
                t2 = i * PERIOD_SAMPLING_RATE + auxFlt;
            }
            TempMed = TempMed + (t2-t1); // sum all diference of times
            count++; // count all changes of sinal
            t1 = t2; // t1 get new time
        }
    }
}
TempMed = TempMed / count;
frequency = 1.0 / (2.0 * TempMed);
//-----FFT-----
calcFFT(p_Ram_RB7, &magArray[0], &magMax, &peakFreqMaxBin, &peakFreqMax);
//-----THD-----
for(i = 2, auxFlt = 0 ; i < HARMONIC_NUM; i++)
    auxFlt = auxFlt + magArray[i] * magArray[i];
THD = sqrtf(auxFlt) / magArray[1];
//----- Individual harmonic voltages in percent-----
for( i = 0; i < (HARMONIC_NUM - 1); i++)
    harmPerc[i] = (100.0 * magArray[i+1] ) / magArray[1];
RamFlag = 0; // disable RamFlag
// T3CONbits.TON = 1; // Turn on Timer 3 for ADC start read again
T5CONbits.TON = 1; // Turn on Timer 3 for ADC start read again
}
} // end while(1)
} // end main
void interpolation2Points( fractional * p_data, float * resultTime){
    volatile float b,m;
    b = Fract2Float( p_data[0] );
    VectorSubtract(1, &auxFrac, &p_data[1], &p_data[0]);
    m = Fract2Float( auxFrac ) / PERIOD_SAMPLING_RATE;
    *resultTime = - (b / m);
}
void calcFFT(fractional * p_data, float * p_magArray, float * magMax, int * peakFreqMaxBin, float *
peakFreqMax){
    volatile int i;
    for(i = 0; i < RAM_NUMSAMP_PER_CH/2; i++){
        sigCmpx[i].real = p_data[i] >> 1;
        sigCmpx[i].imag = 0x0000;
    }
    FFTComplexIP (LOG2_BLOCK_LENGTH-1, &sigCmpx[0], &twiddleFactors[0], COEFFS_IN_DATA);

```

```

/* Store output samples in bit-reversed order of their addresses */
BitReverseComplex (LOG2_BLOCK_LENGTH-1, &sigCmpx[0]);
/* Compute the square magnitude of the complex FFT output array so we have a Real output vector */
SquareMagnitudeCplx(RAM_NUMSAMP_PER_CH/2, &sigCmpx[0],
&VectorAuxFrac[0]);//VectorAuxFrac[0]);
for(i = 0, (*magMax) = 0; i < HARMONIC_NUM+1; i++){
    p_magArray[i] = Fract2Float( (VectorAuxFrac[2*i]) ) * 2 * 425 * 8;
    if(p_magArray[i] > *magMax) {
        *magMax = p_magArray[i];
        *peakFreqMaxBin = i;
        *peakFreqMax = 2 * 2 * i *(SAMPLING_RATE / RAM_NUMSAMP_PER_CH);
    }
}
}
}
void TrueRMS(fractional * vector, float conversion, float * rms){
    volatile int i;
    for(i=0;i<RAM_NUMSAMP_PER_CH;i++) // x^2 and convert
        VectorAuxFlt[i] = Fract2Float(vector[i]) * conversion * Fract2Float(vector[i]) * conversion;
    for(i=0, auxFlt = 0; i < RAM_NUMSAMP_PER_CH; i++) // sum( x^2)
        auxFlt = auxFlt + VectorAuxFlt[i];
    *rms = sqrtf( auxFlt / ( (float)RAM_NUMSAMP_PER_CH ) );
}
void SetupPorts(void) {
    PORTD = 0;
    TRISD = 0xFF0F;
    PORTCbits.RC2=0;
    TRISCbits.TRISC2=0;
    AD1PCFGHbits.PCFG20 = 1;
    AD1PCFGHbits.PCFG21 = 1;
}
void dummyWrite(void) {
    unsigned int *txMsgBox = (unsigned int *) &_DMA_BASE;
    int i = 0;
    for (i = 0; i < 1024; i++) {
        txMsgBox[i] = 0xDEAD;
    }
}
void StartProg(void) {
    LED1 = 1; Nop(); LED2 = 1; Nop(); LED3 = 0; Nop(); LED4 = 1; Nop();
    while (S1 != 0);
    LED1 = 0; Nop(); LED2 = 0; Nop(); LED3 = 0; Nop(); LED4 = 0; Nop();
}
}
adcDrv2.c

```

```

#include "adcDrv2.h"
#include "general.h"
// Buffers in DMA RAM
fractional DMA_BufferA[DMA_BUFF_SIZE] __attribute__((space(dma)));
fractional DMA_BufferB[DMA_BUFF_SIZE] __attribute__((space(dma)));
unsigned int RamFlag = 0; // disable flag
//Buffers in RAM to analyse
fractional RAM_Buffer_RB7 [RAM_NUMSAMP_PER_CH];
fractional RAM_Buffer_RB3 [RAM_NUMSAMP_PER_CH];
fractional RAM_Buffer_RB4 [RAM_NUMSAMP_PER_CH];
fractional RAM_Buffer_RB5 [RAM_NUMSAMP_PER_CH];
fractional SumCh0 = 0,SumCh1 = 0,SumCh2 = 0,SumCh3 = 0;unsigned int posRAM = 0;
unsigned int contTMR3 = 0,contTMR5=0;unsigned int DmaBuffer0 = 0;float auxD;
void initAdc2M(void) {

```

```

AD2CON1bits.ADON = 0; // Turn off the A/D converter
AD2CON1 = 0;
AD2CON1bits.FORM = 3;//10 = Fractional // 0 = Data Output Format: Unsigned Int
AD2CON1bits.SSRC = 7; // 111 = Internal counter ends sampling and starts conversion (auto-convert);
010 = GP timer (Timer3 for ADC1, Timer5 for ADC2) compare ends sampling and starts conversion
AD2CON1bits.ASAM = 0; // ADC Sample Control: Sampling begins when SAMP bit is set
AD2CON1bits.SIMSAM = 1; // Samples CH0, CH1, CH2, CH3 simultaneously (when CHPS<1:0> = 1x);
AD2CON1bits.AD12B = 0; // 10-bit 2/4-channel operation
AD2CON1bits.ADDMABM = 0; // DMA buffers are written in Scatter/Gather mode. The module
provides a Scatter/Gather address to
AD2CON2 = 0;
AD2CON2bits.SMPI = 3; // Increment DMA address every 1 sample/conversion ( 4 ADC buffers -> SMPI
= Nbuf - 1 ) (DMA takes all four conversions)
AD2CON2bits.BUFM = 0;
AD2CON2bits.CHPS = 3; // Converts CH0/CH1/CH2/CH3
AD2CON2bits.VCFG = 0; // (Vdd and Vss) Converter Voltage Reference Configuration bits
AD2CON3 = 0;
AD2CON3bits.ADRC = 0; // ADC Clock is derived from Systems Clock
AD2CON3bits.SAMC = 10; // Auto Sample Time = 10* $T_{ad}$ 
AD2CON3bits.ADCS = 19; // ADC Conversion Clock  $T_{ad} = T_{cy} * (ADCS + 1) = (1/40M) * 20 = 500nS$ 
AD2CON4 = 0; // Allocate 1 words of buffer to each analog input
AD2CON4bits.DMABL = 1; // 7 - Each buffer contains 128 words- //3 - Each buffer contains 8 words
AD2CHS0 = 0;
AD2CHS0bits.CH0SA = 7; // MUXA +ve input selection (AIN6) for CH0
AD2CHS0bits.CH0NA = 0; // MUXA -ve input selection (Vref-) for CH0
AD2CHS123bits.CH123SA = 1; // 0 = CH1 positive input is AN3, CH2 positive input is AN4, CH3
positive input is AN5
AD2CHS123bits.CH123NA = 0; // MUXA -ve input selection (Vref-) for CH1 CH2 and CH3
AD2PCFGL = 0xFFFF;
AD2PCFGLbits.PCFG7 = 0; // AN0 as Analog Input
AD2PCFGLbits.PCFG3 = 0; // AN1 as Analog Input
AD2PCFGLbits.PCFG4 = 0; // AN2 as Analog Input
AD2PCFGLbits.PCFG5 = 0; // AN6 as Analog Input
AD2CSSL = 0x0000; // Channel Scan is disabled, default state
AD2CSSLbits.CSS7 = 1; // Enable AN0 for channel scan
AD2CSSLbits.CSS3 = 1; // Enable AN1 for channel scan
AD2CSSLbits.CSS4 = 1; // Enable AN2 for channel scan
AD2CSSLbits.CSS5 = 1; // Enable AN6 for channel scan
IFS1bits.AD2IF = 0; // Clear the A/D interrupt flag bit
IEC1bits.AD2IE = 0; // Do Not Enable A/D interrupt
}
void initDmaM(void) {
DMA0CON = 0x0000; // Configure DMA in Register indirect mode for (AD1CON1bits.ADDMABM=1)
// Continuous mode (ping-pong disabled)
DMA0CONbits.AMODE = 0; // Configure DMA for Register indirect with post increment
DMA0CONbits.MODE = 2; // 10 = Continuous, Ping-Pong modes enabled ; 00 = Continuous, Ping-
Pong modes disabled
DMA0PAD = (int) &ADC2BUF0; // ADC1BUF0 address
DMA0CNT = (DMA_BUFF_SIZE) - 1; // 127; //32 DMA request (4 buffers, each with 8 words)
// DMA0STA = (unsigned int) &_DMA_BASE; // __builtin_dmaoffset(&BufferATeste[0]);
DMA0STA = __builtin_dmaoffset(DMA_BufferA);
DMA0STB = __builtin_dmaoffset(DMA_BufferB);
DMA0REQ = 21; // 21 = ADC2 ;0001101 = ADC1 ? ADC1 convert done
IFS0bits.DMA0IF = 0; //Clear the DMA interrupt flag bit
// IFS0bits.DMA1IF = 0; //Clear the DMA interrupt flag bit
IEC0bits.DMA0IE = 1; // Enable Interrupt
// IEC0bits.DMA1IE = 1; // Enable Interrupt
DMA0CONbits.CHEN = 1;

```

```

// DMA1CONbits.CHEN = 1;
}
void initTmr3() {
    TMR3 = 0x0000;
    PR3 = 3125; // 3125 = 12.8Kz; 40 = 1MHz; 3199 = 12.5Kz ; 4999 = 8Kz ; 6249 = 6400Hz
    IFS0bits.T3IF = 0;
    IEC0bits.T3IE = 1;
    IPC0bits.T1IP = 1;
    T3CON = 0x0000;
}
void initTmr5() {
    TMR5 = 0x0000;
    PR5 = 3125; //40000= 1ms ;3199 = 12.5Kz ; 4999 = 8Kz
    IFS1bits.T5IF = 0;
    IEC1bits.T5IE = 1;
    IPC1bits.T2IP = 1;
    T5CON = 0x0000;
}
void __attribute__((interrupt, no_auto_psv)) _DMA0Interrupt(void) {
volatile unsigned int i = 0;
PORTCbits.RC2 = 1;
SumCh0=0; SumCh1=0; SumCh2=0; SumCh3=0; // reset all counters
// Look for which RAM Buffer should Write
if(DmaBuffer0 == 0) { // DMA BufferA
    //sum all elements per channel
    for(i = 0; i < DMA_BUFF_SIZE; i=i+4) {
        // first divid by 2 to prevent overflow || (x1+x2)/2 = (x1/2 +x2/2)
        DMA_BufferA[i] = DMA_BufferA[i] >> 1;
        DMA_BufferA[i+1] = DMA_BufferA[i+1] >> 1;
        DMA_BufferA[i+2] = DMA_BufferA[i+2] >> 1;
        DMA_BufferA[i+3] = DMA_BufferA[i+3] >> 1;
        // Add values
        VectorAdd (1,&SumCh0,&SumCh0,&DMA_BufferA[i]);
        VectorAdd (1,&SumCh1,&SumCh1,&DMA_BufferA[i+1]);
        VectorAdd (1,&SumCh2,&SumCh2,&DMA_BufferA[i+2]);
        VectorAdd (1,&SumCh3,&SumCh3,&DMA_BufferA[i+3]);
    }
} else{ //BufferB
    //sum all elements per channel
    for(i = 0; i < DMA_BUFF_SIZE; i=i+4) {
        // first divid by 2 to prevent overflow
        DMA_BufferB[i] = DMA_BufferB[i] >> 1;
        DMA_BufferB[i+1] = DMA_BufferB[i+1] >> 1;
        DMA_BufferB[i+2] = DMA_BufferB[i+2] >> 1;
        DMA_BufferB[i+3] = DMA_BufferB[i+3] >> 1;
        // Add values
        VectorAdd (1,&SumCh0,&SumCh0,&DMA_BufferB[i]);
        VectorAdd (1,&SumCh1,&SumCh1,&DMA_BufferB[i+1]);
        VectorAdd (1,&SumCh2,&SumCh2,&DMA_BufferB[i+2]);
        VectorAdd (1,&SumCh3,&SumCh3,&DMA_BufferB[i+3]);
    }
}
// Put the result in RAM position
RAM_Buffer_RB7[posRAM] = SumCh0;
RAM_Buffer_RB3[posRAM] = SumCh1;
RAM_Buffer_RB4[posRAM] = SumCh2;
RAM_Buffer_RB5[posRAM] = SumCh3;
posRAM++; // increment position for next time
}

```

```

if(posRAM >= RAM_NUMSAMP_PER_CH){
    posRAM = 0; // reset position to put values
    RamFlag = 1; // Enable CPU to do calculations
//    T3CONbits.TON = 0; // Turn off Timer 3
    T5CONbits.TON = 0; // Turn off Timer 3
}
DmaBuffer0 ^= 1; // change DMA Buffer
IFS0bits.DMA0IF = 0; //Clear the DMA0 Interrupt Flag
PORTCbits.RC2 = 0;
}
void __attribute__((interrupt, no_auto_psv)) _DMA1Interrupt(void) {
    IFS0bits.DMA1IF = 0; //Clear the DMA interrupt flag bit
}
void __attribute__((interrupt, no_auto_psv)) _ADC1Interrupt(void) {
    IFS0bits.AD1IF = 0; // Clear the A/D interrupt flag bit
}
void __attribute__((interrupt, no_auto_psv)) _ADC2Interrupt(void) {
    IFS1bits.AD2IF = 0; // Clear the A/D interrupt flag bit
}
void __attribute__((__interrupt__, no_auto_psv)) _T3Interrupt(void){
    IFS0bits.T3IF = 0;
    AD1CON1bits.SAMP = 1;
}
void __attribute__((__interrupt__, no_auto_psv)) _T5Interrupt(void){
    IFS1bits.T5IF = 0;
    AD2CON1bits.SAMP = 1;
}
}
general.c
#ifndef GENERAL_H
#define GENERAL_H
#include "p33Fxxx.h"
#include <dsp.h>
#include "math.h"
#include "stdio.h"
#include "stdlib.h"
#include "string.h"
#include "libq.h"
/* Local defines. */
#define FCY 4000000 // User must calculate and enter FCY here
#define BAUD 19200
#define Dly_Time (20E-6 * FCY) // ADC Off-to-On delay
#define MILLISEC FCY/10000 // 1 mSec delay constant
#define LED1 PORTDbits.RD4 // LED connected to RD4
#define LED2 PORTDbits.RD5 // LED connected to RD5
#define LED3 PORTDbits.RD6 // LED connected to RD6
#define LED4 PORTDbits.RD7 // LED connected to RD7
#define S1 PORTAbits.RA12 // switch S1
#define S2 PORTAbits.RA13 // switch S2
// adcDrv2
#define NUM_OF_CHS 8 // number of channels to read data
#define NUM_OF_CHS_PER_ADC 4 // number of channels to read data per ADC

#define DMA_BUFF_SIZE_PER_CH 2 // Size of the input buffer per analog input (DMA)
#define DMA_BUFF_SIZE (NUM_OF_CHS_PER_ADC * DMA_BUFF_SIZE_PER_CH) //(4*2 = 8) Total
size of DMA RAM used
#define NUM_OF_PERIOD 4 // Number of periods of Power Line frequency (50Hz) to capture
#define RAM_NUMSAMP_PER_CH (128 * NUM_OF_PERIOD) // Number of samples per Channel in
RAM

```

```

#define ADC_NUM_BITS 10
#define ADC_VOLT 3.3
#define RANGE (pow(2,ADC_NUM_BITS)) // 2^10=1024
#define ADC_RESOLUTION (ADC_VOLT / RANGE )
#define CONVERSION_VOLT 425 // conversion to real values(float) from frational values for VOLT
#define CONVERSION_CURRENT 4.872 // conversion to real values(float) from frational values for
Corrent
#define CORRECTION_VOLT 0.0 // Corretion/ Calibration
#define CORRECTION_CURR 0.0 // Corretion/ Calibration
#define LOG2_BLOCK_LENGTH 9 // 9 -> 512; 8 -> 256; ...
#define SAMPLING_RATE 6400.0 // Rate at which input signal was sampled
#define PERIOD_SAMPLING_RATE (1.0 / SAMPLING_RATE)
#define HARMONIC_NUM 25 // harmonic number to analyze
// Definitions for subroutines and interrupt vectors
void SetupPorts(void);
void StartProg (void);
void DelayNmSec(unsigned int N);
void dummyWrite(void);
// External variables
extern unsigned int DmaBuffer0, DmaBuffer1;
extern unsigned int RamFlag;
extern fractional RAM_Buffer_RB7 [RAM_NUMSAMP_PER_CH];
extern fractional RAM_Buffer_RB3 [RAM_NUMSAMP_PER_CH];
extern fractional RAM_Buffer_RB4 [RAM_NUMSAMP_PER_CH];
extern fractional RAM_Buffer_RB5 [RAM_NUMSAMP_PER_CH];
// External funtions
extern fractional Float2Fract(float aVal);
extern float Fract2Float(fractional aVal);
extern void initAdc1M(void);
extern void initAdc2M(void);
extern void initDmaM(void);
extern void initTmr3(void);
extern void initTmr5(void);
#endif /* GENERAL_H */

```