Ronny Knoch gieseler

Sistema de Avaliação de Sudorese para Detecção de Neuropatias autonômicas em Diabetes

Dissertação apresentada ao Curso de enegenheria elétrica, na área de engenheria biomédica da Universidade Federal de Santa Catarina, como requisito parcial para a obtenção do título de Mestre em engenharia elétrica.

Orientador: Jefferson Luiz Brum Marques, PhD.

FLORIANOPÓLIS

2016

ronny knoch gieseler

Sistema de Avaliação de Sudorese para Detecção de Neuropatias autonômicas em Diabetes

Dissertação apresentada ao Curso de enegenheria elétrica, na área de engenheria biomédica da Universidade Federal de Santa Catarina, como requisito parcial para a obtenção do título de Mestre em engenharia elétrica.

Orientador: Jefferson Luiz Brum Marques, PhD.

FLORIANOPÓLIS

2016

FICHA CATALOGRÁFICA

Sobrenome, Nome

Nome do trabalho nome do trabalho nome do trabalho nome do trabalho nome do trabalho nome do trabalho – Cidade, ano.

Nº de páginas

Área de concentração: Lorem ipsum.

Orientador: Prof. Dr. Fulano de Tal.

Tese (DOUTORADO ou Mestrado) – Instituição com toda a hierarquia.

1.Palavra chave; 2. Palavra chave; 3. Palavra chave

A minha esposa Camila, e as minhas irmãs Letícia, Ana Beatriz e Isabel, pelo tempo que deixamos de estar juntos...

Aos meus pais, Edmar e Sônia, a eles todos os créditos...

Dedico

Agradecimentos

A quem você agradece, exemplo:

A minha equipe de trabalho, sua organização tornou meu desafio de lidar com o tempo possível.

Ao Prof. Dr.Osvaldo Malafaia, pela dedicação nas correções e orientações neste período de aprendizado.

A Prof.ª Sirlei Terezinha Bittencourt, que mudou minha visão sobre a sala de aula.

Aos meus colegas de pós-graduação que tornaram um período de longa dedicação em algo divertido.

Ao Pabllo por fazer tudo.

A Larissa por forjar os resultados.

“Na realidade, os maiores bens vem-nos da loucura, que é sem dúvida um dom divino" / "Um tolo sempre encontra um mais tolo que o admire" Boileau-Despréaux

Platão.

.

Sumário

LISTA DE ILUSTRAÇÕES vii

Resumo viii

Abstract ix

1 Introdução 11

1.1 objetivo 11

2 Revisão da literatura 14

2.1 DIABETES Mellitus. 14

**2.1.1** **Neuropatias multiplas** 14

*2.1.1.1* *Neuropatia autonônica* 16

*2.1.1.2* *Dispersão elétrica e Relaxação* 18

*2.1.1.3* *Propriedades elétricas em frequência* 19

2.2 HARDWARE 19

2.2.1 Aquisição dos dados 20

2.2.2 Alimentação 21

2.3 Processamento de sinais 21

3 Material e Método 24

3.1 HARDWARE Desenvolvido 24

**3.1.1** **Controle de ganho** 26

**3.1.2** **Filtragem** 27

**3.1.3** **Alimentação** 28

3.2 Software 30

**3.2.1** **Processamento dos sinais** 30

**3.2.2** **Aplificação dos sinais** 31

**3.2.3** **Conversores AD-DA** 32

**3.2.4** **Alimentação** 33

**3.2.5** **Estrutural** 33

3.3 unidade de processamento 33

3.4 Processamento dos sinais 34

3.5 testes de funcionamento e validação 34

3.6 comitê de ética e coléta de dados 36

3.7 ANÁLISE estatística 37

4 RESULTADOS 39

4.1 Analise dos sinais 39

5 DISCUSSãO 41

6 CONCLUSÕES 43

6.1 Trabalhos futuros 43

ReferÊncias 44

Apêndice 1 - código VHDL principal 48

Apêndice 2 - código VHDL para algoritmo de goertzel 54

Apêndice 3 - código matlab 55

Apêndice 4 - código c# para processamento e interface 57

Anexo 1 - comite de ética - TCLE 66

LISTA DE ILUSTRAÇÕES

Nenhuma entrada de sumário foi encontrada.

Resumo

SISTEMA DE AVALIAÇÃO DE SUDORESE PARA DETECÇÃO DE NEUROPATIAS AUTONÔMICAS EM DIABETES

**Objetivo:** Estimular as glândulas sudoríparas de mãos e pés com um sinal de corrente contínua mais multifrequencial para detectar neuropatia autonômica ligada a deficiências de sudorese e as variações no espectro entre voluntários saudáveis e com diabétes. **Material e Método:** Teste com sessenta pacientes com diabétes tipo 2 dentro do Hospital Universitário da UFSC e trinta pacientes sadios. Avaliação a partir de um dispositivo desenvolvido para estimular mãos ou pés com uma tensão máxima de 5,0 Volts por meio de eletrodos de aço inoxidável em um período de 5 minutos, salvando os dados coletados e mostrado o resultado por meio de interface gráfica. **Resultados:** A carga máxima dos corpos de prova não solidarizados foi de 849,4N ± 386,8 a área 30,4 mm2± 7,7, tensão de 29 ± 17Mpa. Os solidarizados obtiveram carga máxima de 871,8N ± 484,9 área 35 mm2± 5,8, tensão de 24 ± 10Mpa. Não houve diferença estatística entre os dois grupos (p>0,05). **Conclusão**: A distribuição de probabilidade mostra que para 400 N os tendões não solidarizados apresentam confiabilidade de 83,8% e os solidarizados de 78,5%%.

**Descritores:** Neuropatia Autonômica, Deficiência de sudorese, Espectroscopia de Impedância Elétrica, Diabétes.

Abstract

SWEATING ASSESSMENT SYSTEM FOR DETECTION OF AUTONOMIC NEUROPATHY IN DIABETIC PACIENTS

**Purpose:** To evaluate the difference of the mechanical behavior of bended tied (sewed) and not tied tendons in order to evaluate if it has any effective function for the anterior cruciate ligament reconstruction. **Material and Method:** Twenty fresh bovine digital tendons had been used. Alginate to determine the area of the tendon transversal section was used. Ten bovine tendons had been bended and tied following the manufacturer orientation, ten others pairs had not been tied. Claws had been developed for the grafts fixation to the universal machine, for the setting simulation. The superior one was characterized to be bipartite and to have controlled passage for the pins and the inferior claw is characterized to have alternating teeth. **Results:** The maximum load of the not tied samples of the test was 849,4N ± 386,8 the area was 30,4 mm2 ± 7,7, and 29 ± 17Mpa of tension. The tied ones had gotten maximum load of 871,8N ± 484,9, area of 35 mm2 ± 5,8, and 24 ± 10Mpa of tension. It did not have statistical difference between the two groups (p>0,05). **Conclusion**: The probability distribution showed that with 400 N the not tied tendons represents 83,8% of trustworthiness and the tied ones represents 78,5%.

**Key words:** Anterior Cruciate Ligament, Femur, Orthopedic Fixation Device, Mechanics, Tendons

# Introdução

O Diabetes mellitus (DM) é uma das doenças de maior prevalência na população mundial [1]. Dentre os problemas mais graves causados pelo DM estão as neuropatias múltiplas causando problemas nos olhos, rins, coração, vasos sanguíneos e nervos; neste último, se afetar orgãos é conhecida como autonômica, se afetar o SNC é conhecida como central, se afetar mãos e pés é conhecida como neuropatia periférica , neste caso os primeiros a serem afetado são os pés, uma vez que possuem fibras nervosas mais longas. A neuropatia causará uma agressão metabólica nas fibras nervosas, insuficiência neurovascular, danos auto-imune e deficiência de hormônio do fator de crescimento [13]. Um dos poucos exames que diagnosticam eficientemente a neuropatia é a medição da condução nervosa [11], linkado a neuropatia periférica, mas existem vários exames secundários que podem acusar problemas decorrentes da condução nerval. Recentemente foi mostrado que há uma correlação entre a densidade de fibras nervosas nas glândulas com a atuação do sistema nervoso, deficiências neurológicas e produção de suor em indivíduos com diabetes[2][3][6][7][10] o que justifica a análise de sudorese para problemas de neuropatias.

## objetivo

Desenvolver um sistema que possa avaliar problemas relacionados ao DM, principalmente as neuropatias autonômicas, obtendo os valores de análise por estimulação elétrica das glândulas sudoríparas a partir de iontoforese inversa. O sistema vai estimular as glândulas das mãos e pés com uma tensão menor que 5,0 Volts. Por iontoforese inversa, este estimulo atrairá íons em seu cátodo. Medindo o sinal resultante de resistência entre membros, poderá ser estimado o valor de resistência de contato entre membro e eletrodo. Quanto maior for o valor desta impedância de contato, menor será o nível de íons de cloreto, suor, e menor será a condução de sinal pelo sistema nervoso simpático do paciente para estimulação das glândulas, o que servirá de indicativa para problemas de condução nervosa e consequentemente de neuropatias. Para cumprir este propósito, foram traçados os seguintes objetivos específicos.

1. construir um equipamento de detecção de neuropatia autonômicas a partir de problemas de sudorese capas de acusar risco de diabetes
2. desenvolver software para processamento dos sinais adquiridos
3. desenvolver um sistema de interface para facil visualização dos resultados
4. realizar um estudo com indivíduos portadores de diabetes ou neuropatia autonômica e compará-los aos valores de indivíduos sadios

# Revisão da literatura

## DIABETES Mellitus.

O diabete mellitus é uma patologia que descreve uma desordem metabólica que pode ter multiplas origens caracterizada por uma hyperglicemia crônica resultado da falta de secreção de insulina, falta de ação da mesma ou ambos. Existem dois tipos de diabetes principais, o tipo 1, também conhecida como insulinodependente, infanto-juvenil e imunomediado, que ocorre normalmente em crianças e requer injeção de insulina diária para repor a falta da fabricação pelo corpo. A origem deste tipo ainda é desconhecida e não é prevenida com o conhecimento atual. O tipo 2, também conhecida como não insulinodependente ou diabetes do adulto, ocorre normalmente em adultos e está relacionado a obesidade, falta de atividade física e dietas com excesso de carboidratos, sendo este o tipo mais comum, representando 90% de todos os casos de diabetes em todo mundo. Outros tipos de diabetes são a gestacional, que é um estado de hyperglicemia que ocorre durando a gravidez, inchaço do pâncreas conhecido como pancreatite, doenças genéticas como a fibrose cística e outras formas de diabetes geradas por drogas, viroses ou causas ainda desconhecidas. [who]

O diabetes causa danos aos olhos, levando a cegueira, rins, podendo causar falha renal e nervos, levando a desordens em membros podendo causar amputações. Também aumenta o risco de doenças no coração, sendo que 50% dos pacientes com diabétes morrem por problemas cardiovasculares, podendo ocorrer ainda derrames e insuficiência de fluxo de sangue para os pés [who].

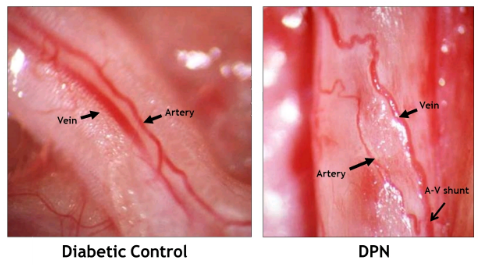
### **Neuropatias multiplas**

Segundo Solomon el at. (2010) as neuropatias podem ser classificadas em dois grupos diferentes. As Neuropatias multiplas típicas em diabetes (DSPN) são patologias crônicas, causadas por longos periodos de hiperglicemia associados a desajustes metabólicos (aumento de carboidratos, acumulo de produtos de glicação, stress oxidativo e alterações de estruturas lipídicas, dentre outros), disfunções autonomicas podem se desenvolver com o tempo. O teste de condução nerval é o teste padrão para diagnosticar esse tipo de patologia. Já as Neuropais multiplas atípicas em diabetes estão relacionadas a dores e disfunções autonomicos, podendo ocorrer em qualquer estágio do diabetes, seu diagnóstico ainda é pouco claro, e atualmente se usa apenas a dor que o paciente diz sentir como diagnóstico. (Die care 2010).

Algumas subclassificações de neuropatias são a neuropatia autonômica, se afetar os orgãos, periférica se for apenas nas fibras nervosas para os membros e central se afetar o sistema nervoso central (SNC).

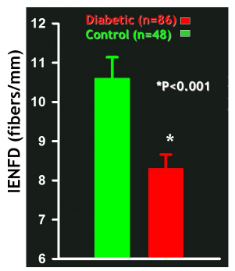
Segundo Yagihashi, um dos principais problemas que a hiperglicemia causa, é a hipoxia de nervos, isso ocorre devido a conexões chamadas shunt, entre artérias e veias. Em um ser humano saudável, as fibras nervosas são vascularizados integralmente, enquanto que em indivíduos com neuropatias multiplas o sangue oxigenado tende a seguir apenas o caminho que o liga as veias. A abaixo mostra a diferença entre indivíduos de controle e com diabetes com neuropatias multiplas. No último as veias são mais grossa, pois absorvem grande parte do sangue oxigenado, enquanto que o nervo tende a sofrer hipoxia e perder sua densidade fibrosa. A diminuição da densidade fibrosa foi publicada por Ziegler et al, 2014 e é mostrada na .

Figura 1 - DIFERENÇAS ENTRE OXIGENAÇÃO DE FIBRAS NERVOSA ENTRE INDIVÍDUOS DE CONTROLE E DIABÉTICOS COM MULTIPLASNEUROPATIAS



Fonte: (Tesfaye et al. 1993)

Figura 2 - diferenças de densidade fibrose entre indivíduos de controle e diabéticos tipo 2 (Ziegler et al. 2014)



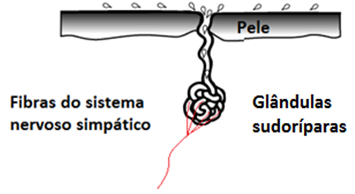
Fonte: Ziegler et al. 2014

#### *Neuropatia autonônica*

Este tipo de neuropatia é causa uma desordem no sistema autonômico relacionados a problemas em fibras nervosas interconectadas com diferentes orgãos, podendo afetar a vascularização cardíaca, o sistema gastrointestinal, sistema urogenital e função sudorípara. Nesta última as glândulas são enervadas por fibras sem mielina, do tipo C, do sistema nervoso simpático. A disfunção de suor causa aridez e foi associada com ulcerar de pés e mãos, um dos grandes problemas relacionados a diabetes (TENTOLOURIS, 2009)

As fibras nervosas das glândulas sudoríaras irão comprimir a glândula fazendo com que o suor seja filtrado do sangue em um duto quase reto que levará o líquido até o poro. A parte mais externa da pele é chamada de Stratum Corneum.

Figura 3- Glandula eccrina que transporta corrente CC no eletrôlito, suor. O cloreto estará no anodo enquanto os protons estarão no cátodo

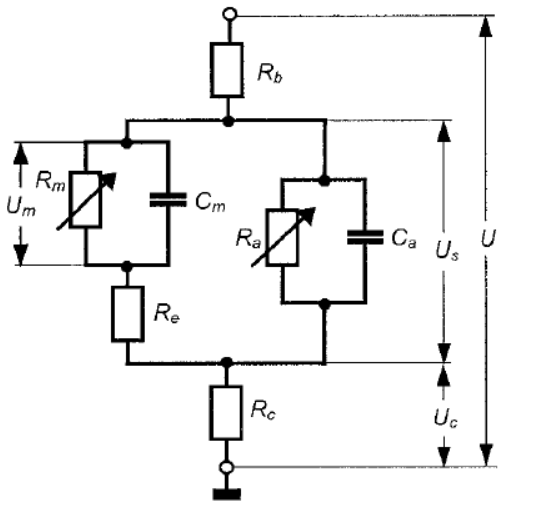


Fonte: Próprio autor

Recentemente foi mostrado que há uma correlação entre a densidade de fibras nervosas nas glândulas com a atuação do sistema nervoso, deficiências neurológicas e produção de suor em indivíduos com diabetes[2][3][6][7][10] o que justifica a análise de sudorese para problemas de neuropatias.

Para medição da função sudomotora, são colocados eletrodos em regiôes que possuam grande quantidade de glândulas sudoriparas e é medida a condutância. Normalmente são avaliados entre mãos ou entre pés. Acredita-se que existam apenas dois caminhos para a corrente circular ao passar pelo tecido epitelial (Potts et al., 1992), um passando pela matrix lipídica do *stratum corneum* (SC)e outro que passa pelas glândulas de sudorese e seus folículos (Chizmadzhev et. al.1998). A mostra uma representação elétrica do sistema estudado.

Figura 4 - Circuito equivalente da epiderme. Ra e Ca representam as glândulas. Rm e Cm representam a camada lipídica. Rb, Re e RC são resistências referêntes ao líquido extracelular.



Fonte: Chizmadzhev et al. 1998

As análises de função de sudorese são realizas sempre em nível DC [2][5][6]. Desta forma uma avaliação multi-frequêncial tornará possível avaliar o modelo atualmente utilizado, propondo alterações, discutindo valores e analisando diferenças entre indivíduos com diabetes e sadíos.

Outra representação, mais simplificada, possível seria pelo modelo já bem conhecido de Cole-Cole, 1940.

|  |  |
| --- | --- |
|  |  |

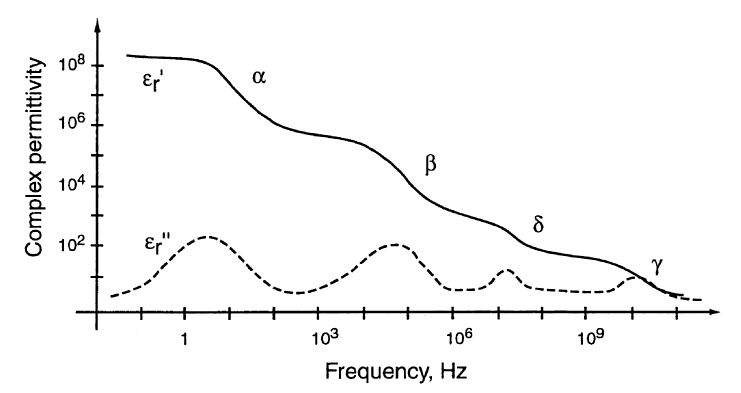
Onde ω é a frequencia, Ro e R∞ são as componentes de impedância na frequencia de 0Hz e infinitos Hz e α é o fator de Cole (0< α <1) depentendo do tempo de relaxação da estrutura.

#### *Dispersão elétrica e Relaxação*

Segundo Martinsen, 2011 o fenônemo de relaxação se dá pelo alinhamento e distribuição das cargas sobre um determinado material biológico a partir da estimulação de um campo elétrico.

Schawn, 1957, mostrou que que existem três regiões principais diferentes no espectro de impedância elétrica devido a dispersão dielétrica, chamadas de α (alfa), β (beta) e γ (gama) sendo δ(delta) apenas uma região intermediária.

Figura - Relaxação



Fonte: Martinsen et al. 2002

Esses resultados são empíricos e tem-se algumas sugestões para a origem dos mesmos. até kHz sabe-se que os efeitos causados são devido aos íons presentes na membrana celular (SCHWAN; TAKASHIMA, 1993).

#### *Propriedades elétricas em frequência*

Através da analíse em frequência de um biomaterial, é possível perceber diferenças causadas pelas suas diferenças partes. Um biomaterial será formado por células e fluidos extracelulares e, devido a existências de íons em no material intra e extra-celular, ambos podem ser considerados condutores de corrente elétrica.

Como a presenças de uma membrana celular semi-permeável nas celulas biológicas agrega uma característica capacitiva, o sistema equivalente passa a ter uma impedância elétrica pelo meio extra-celular muito menor em baixas frequências. A partir do momento em que a frequência da corrente de excitação começa a aumentar, a influência da capacitância no valor de impedância resultando acaba se tornando desprezível , e a corrente já passa de forma mais homogênea por meio do material [VEIGA].

As membranas celulares são compostas principalmente de proteínas e lipidios, possuindo uma condutividade de 0.1uS/m e uma capactiância típica de 10mF/m² (PETHIG, 1984). Segundo Walker et al. (2000), os valores típicos de condutância intra e extra-celular são da ordem de 1.0S/m e 2.0S/m respectivamente.

## HARDWARE

O hardware a ser desenvolvido seguirá as especificações da , contendo uma FPGA para envio, recepção e transmissão dos dados via USB, uma interface homem-máquina para disponibilização dos resultados e controle do equipamentos, um conversor digital-analógico (DAC), um conversor analógico-digital (ADC), um sistema de filtragem pra a saída do DAC e um sistema de sensoriamente composto por um resistor shunt e um sistema de amplificação.

Figura 6 - Demonstrativo do sistema de completo

Fonte: Próprio autor

### Aquisição dos dados

Para controlar as frequências a serem injetadas, o sistema utilizará um conversos digital-analógico (DAC) de alta banda. Para regular a impedância de saída do conversor, será necessário um conversor tensão/corrente, que pode ser feito por meio um amplificador operacional seguindo o layout expecificado pelo datasheet do própria fabricante do DAC. Depois disso será necessario um filtro passa baixas para retirar as variações de alta frequência geradas pelo conversor.

Para aquisição do sinal de tensão, será necessário um conversor analógico-digital (ADC), também de alta banda. O sinal de tensão precisará ser ajustado por meio de amplificadoress para se adequar aos valores de entrada requeridos pelo ADC.

### Alimentação

## Processamento de sinais

Para análise de espectroscopia de impedância são utilizadas diversas frequencias espalhadas no espectro. O equipamento que pode fazer essas medições é o analisador de impedância da agilent 4294, que utiliza 200 pontos entre 40 e 110MHz, mas essa quantidade acaba sendo superior ao necessário. Veiga, 2013 utilizou apenas 30 pontos de frequencia com uma separação logarítima entre 1kHz e 1MHz, mas os dados só deixaram de ser constantes a partir de 300kHz onde estava localizada o decaimento de impedância.

A utilização de uma frequencia unica de excitação, normalmente de 10kHz ou 50kHz, é utilizada para adquirir informações importantes sobre o comportamento do material e alguns trabalhos como Yúfero, 2008 e Neves, 1999 utilizam apenas essa estimulação. Esta análise é interessante justamente pela impedância de tecidos biomédicos seguir o equacionamento de Cole-Cole, 1940 mantendo-se constante até uma determinada frequência.

Para análise multifrequencial, os sistemas de processamento podem ser feitos por meio de microcontroladores e processadores de sinais digitais (DSP), porém eles possuem processamento sequencial e falta de memória. Processadores de sinais digitais podem ser usados para geração dos sinais e posterior conversão digital-analógica mas necessitarem de sistemas de condicionamento dos sinais analógicos para excitação e captação.

Em contra partida, os sistemas baseados em FPGA são indicados para tratamento de sinais que necessitem de forte integração e redução de ruído, sendo que o sistema é intrinsicamente feito com processamento em tempo real, uma vez que o mesmo é feito em nível de hardware, além de possuir consumo menor que as DSPs (Veiga, 2013).

O FPGA é um circuito integrado que contém centenas de milhares de unidades lógicas programáveis idênticas. Elas são programadas e modificadas para criar circuitos digitais específicos que interagem entre sí formando matrizes. A FPGA pode disponibilizar diversos componentes e solucionar um determinado problema, dentre eles contadores, multiplicadores, acumuladores, simular microcontroladores, interfaces de comunicação dentre outros.

Um código para FPGA pode ser escrito basicamente por meio de Verilog ou VHDL. Mas dependendo do compilador, outras linguagens são aceitas, sendo que todas elas serão convertidas em VHDL quando forem programar o dispositivo. Essa transformação faz com que haja perda de rendimento com a criação de linhas de código desnecessárias para uma determinada aplicação.

A Altera, fabricante de FPGAs, disponibiliza diversos "IP Cores" (Núcleos de Propriedade intelectual), que são funções complexas disponibilizadas para facil integração com o hardware. Eles são disponibilizados em diferentes linguagens suprindo partes como comunicação USB, gerenciamento e tratamento de sinais do conversor AD, tratamento de audio, gerenciamento de memória dentre outros. Facilitando a programação e desenvolvimento dos protótipos

Entretanto calculos mais complexos exigem muita dedicação do programador para serem executadas em uma FPGA. Divisões, por exemplo, acabam consumindo diversas células logicas fazendo com que a ultilização desdes dispositivos seja mais utilizadas para o envio, processamento inicial e condicionamento dos sinais, enquanto que os calculos como transformada de fourier, compilação dos dados, interface com o usuário são feitas por meio de um computador com clock, processamento e memória adequados.

3

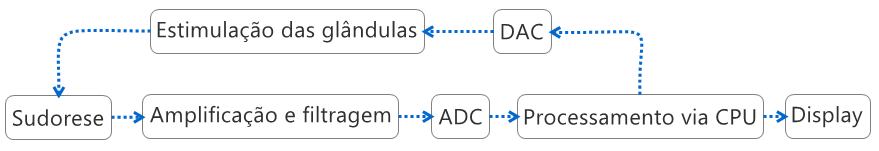
# Material e Método

A presente dissertação de mestrado trata do desenvolvimento de um sistema integrado de aquisição e análise de sinais decorrente da estimulação de glândulas sudoríparas. Este trabalho pode ser dividido basicamente em três partes: Hardware, Software e o Processamento dos sinais coletados. Para melhor entendimento cada parte será apresentada separadamente neste capítulo.

## HARDWARE Desenvolvido

O analisador de neuropatia autonômica é composto por um medidor de sudorese que tem um sensor resistivo tipo ponte de wheatstone para determinação dos valores das resistências[9]. O sinal excitatório DC menor que 4,0 Volts juntamente com componentes multifrequenciais também de baixa amplitude são enviado pela FPGA via um conversor DAC. Esse sinal é filtrado e usado para estimular as glândulas. O valor de tensão lido do sensor é amplificado para condicioná-lo a entrada do conversor ADC. O sinal será organizado e enviado para um computador via USB pela FPGA realizando filtros digitais para retirar ruído, calcular a derivada do valor de resistência com relação ao tempo e determinar o valor de resistência inicial (basal), que servirá para comparar diferentes pacientes, analisando assim somente as resistências de contato. Posteriormente o valor absoluto em que o sistema se estabilizou, o gráfico de variação, sua derivada e a comparação com seu valor basal são disponibilizados de forma visual por meio de uma interface gráfica em c#.

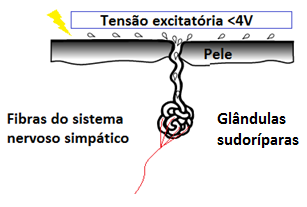
Figura 7 - Diagrama em Blocos do sistema completo a ser desenvolvido



Fonte: Próprio autor

A apresenta o sistema de excitação de uma das mãos. A impedância será medida baseada na corrente que passa entre os membros. Essa baixa tensão aplicada na pele vai gerar uma corrente através de iontoforese inversa, i.e pelo movimento de íons via os poros de suor em resposta a estimulação elétrica [8]. A tensão aplicada será menor que 4,0 Volts, visto que com tensões baixas a corrente não consegue passar pela parte lipídica devido a alta capacitância da mesma [5][11]. Posteriormente será aplicado a tensão de excitação DC juntamente com um sinal multifrequencial para determinação de características elétricas do sistema.

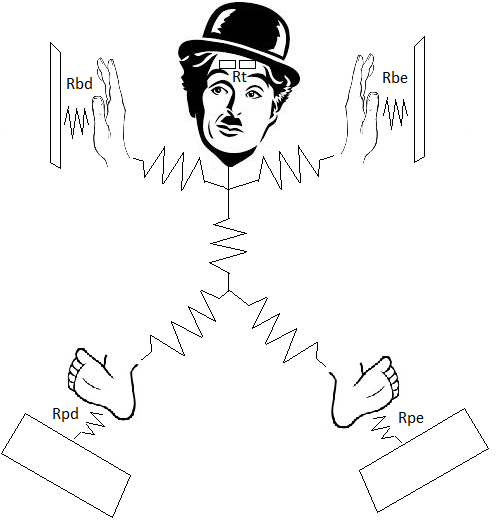
Figura 8 - Sistema de excitação de sudorese. A tensão excitatório de baixa magnitude vai estimular as glandulas sudoriparas a liberar cloreto sem grandes perdas pela camada lipídica da pele



Fonte: Próprio autor

O paciente ficará sentado, com os membros relaxados, com as mãos, pés e testa conectados a eletrodos de aço inoxidável. Esse tipo de eletrodo foi utilizado pela alta capacidade de detectar variações de Clˉ [12]. A estimulação se dará pelas mãos, pés, ou por ambas. A sudorese na região da testa será avaliada separadamente. A mostra um circuito elétrico equivalente para uma pessoa. Estimulando o paciente com uma diferença de potencial entre os eletrodos dos dois braços, por exemplo, será possível verificar uma alteração no valor de Rbe ou Rbd, dependendo da localização do cátodo da fonte, uma vez que o valor da resistência do braço, tronco e pernas se manterá relativamente constante.

Figura 9 - Resistências consideradas. Rbe e Rbd são as resistências de Contato entre os eletrodos e as mãos. Rpe e Rbd são as resistências de contato entre os eletrodos e os pés. Rt é a resistência equivalente de contato entre a testa e ambos os eletrodos.

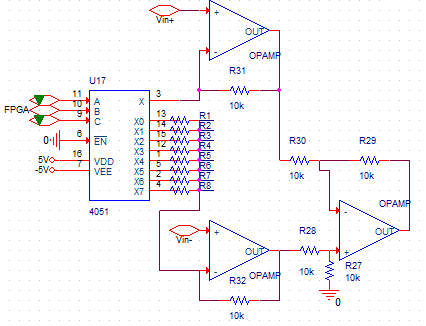


Fonte: Próprio autor

### **Controle de ganho**

O controle de ganho é utilizado para garantir que a tensão lida pelo conversor ADC, e que será enviada para a FPGA, seja da ordem esperada. A amplificação será feita por três ampops em configuração de amplificador de instrumentaçãor. Para o controle de ganho será utilizado um multiplexador com diferentes resistores escolhidos por meio de três sinais oriundos da FPGA.

Figura 10 - Controle de ganho



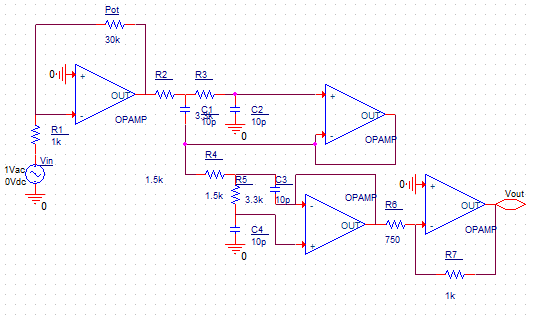
Fonte: Próprio autor

Os resistores R1 a R8 serão resistores para controle do ganho de aproximadamente 1 até 25. O multiplexador 4051 foi usado pois possui uma resistência das portas lógicas de apenas 200Ω e alimentação dual.

### **Filtragem**

O sinal injetado pela FPGA e convertido pelo conversor DA passará por um filtro e 3MHz como mostra a Figura 11. O primeiro estágio é composto por um ampop na configuração de amplificador inversor de ganho variável. O segundo estágio, composto por 2 ampops, é um filtro passa baixa de 3MHz de 4 ordem. A última etapa, composta por 1 ampop é um ganho fixado.

Figura - FIltro passa-baixas

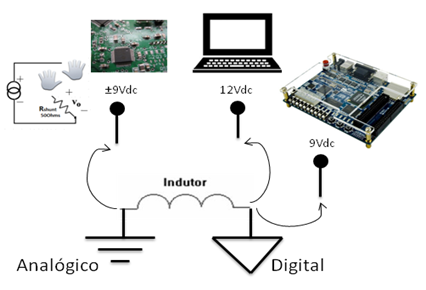


Fonte: Próprio autor

### **Alimentação**

O sinal que é injetado no tecido será provido diretamente por quatro baterias de 3,7 Volts ligadas em série com o terra ligado entre a segunda e a terceira pilha, formando uma tensão positiva e uma negativa de 7,4 Volts. Serão necessários circuitos para conversão desta tensão para a tensão de excitação do tecido e tensão de alimentação dos CIs. As únicas partes ligadas a rede elétrica serão o notebook que terá a interface com o usuário e a placa da FPGA que fará os primeiros processamentos nos dados recolhidos. Essa alimentação, entretanto, será por meio de um transformador isolador e conversores de tensão para 12.0 e 9.0 Volts, e a ligação entre os terras analógico e digital se dará por meio de um indutor de 10mH, auxiliando para que os ruídos da parte digital não aparecem na parte analógica (Johnson, H.;Graham, M, 1993). A Figura 12 - Alimentação do sistemamostra o esquemático de separação das alimentações.

Figura - Alimentação do sistema

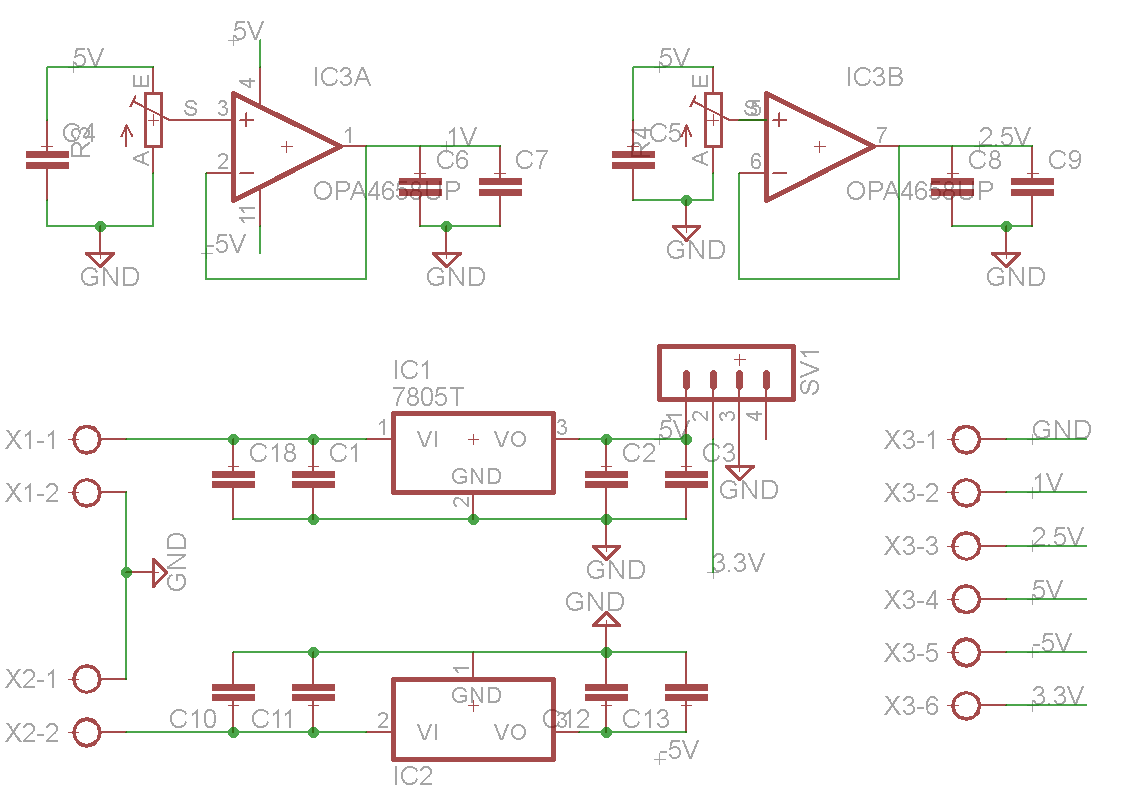


Fonte: Próprio autor

A alimentação do Controle de Ganho e dos filtros é de ±5,0V, a tensão dos conversores ADC e DAC é de +3,3V e eles ainda precisam de tensões de referências de +2,5V para o ADC e +1,0V para o DAC. A alimentação principal é feita por meio de quatro baterias de 3,7 Volts ligadas em série para geral uma refêrencia de terra. As tensões de ±5,0V e +3,3V são geradas por reguladores de tensão 7805, 7905 e 78R33. Divisores resistivos com amplificadores em configuração de buffer são usados para gerar as tensões de referência.

A Figura 13 mostra o circuito de alimentação com as entradas X1/X2 sendo de quatro baterias de 3.7 Volts ligadas em série. Os capacitores de regulação são de 10uF e 10nF.

Figura - Alimentação



Fonte: Próprio autor

## Software

A interface gráfica com o usuário foi feita em C#, para o usuário ter a possibilidade de fazer análises e imprimir os resultados. Foi utilizado o software Visual Studio 2013 e comunicação USB para comunicação com a FPGA. A comunicaçao USB foi feita de forma Bulk, com o a FPGA agindo como escrava na comunicação e preenchendo o buffer do End Point a todo instante. Foi utilizada a biblioteca LibUsbDotNet criada por Travis Robinson.

O programa recebe os dados da análise da função sudomotora das mãos, pés e testa. mostrando-os em um gráfico juntamente com sua taxa de variação, a estatística mais relevante na iontoforese reversa.

A programação da FPGA foi feita utilizando a IDE da altera Quartus II, a licensa do software foi obtida gratuitamente por meio de vinculo entre altera e professor orientador. A comunicação USB básica foi feita utilizando a biblioteca fornecida por Mikhail Zakhaov. As operações matemáticas necessárias para a transformada de Goertzel foram feitas utilizando bibliotecas float\_pkg para manipulação de variáveis no formado de ponto flutuante disponibilizadas por David Bishop.

### **Processamento dos sinais**

Os sinais adquiridos possuem pouco informação útil quando avaliados no tempo, mas ao sofrerem uma transformada em frequência podem ser separados e avaliados. Existem duas formas principais para realização desta transformada. Por meio da transformada rápida de Fourier ou por meio do Algoritmo de Goertzel. O algoritmo de Goertzel é uma forma simplificada da transformada de fourier, calculando o apenas uma componente expecífica no espectro de modo recursivo seguindo a equação (3) com base na equação (2) da DFT

|  |  |
| --- | --- |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  |  |

Sendo o ponto de frequência de interesse para o qual usam-se as condições iniciais e na equação (4) dada por:

|  |  |  |
| --- | --- | --- |
|  |  |  |

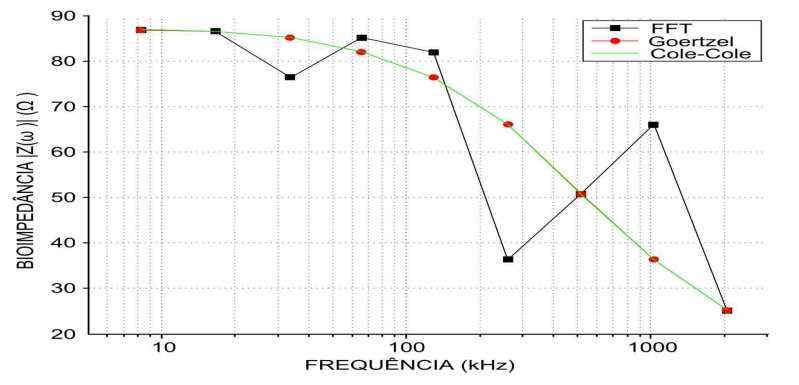
O algoritmo de Goertzel reduz o número de operações aritméticas efetuadas no cálculo da transformada de fourier (FFT), já que a calcula para uma única frequência. Essa técnica é relevante para o problema quando se está usando uma FPGA, pois a mesma não precisa fazer mais tratamentos com os dados recolhidos para encontrar as frequências de excitação dentro de um grande vetor de dados. Desta forma foi desenvolvido um algoritmo de Goertzel em VHDL utilizando a bibliotecas float\_pkg para manipulação de variáveis no formado de ponto flutuante disponibilizadas por David Bishop, 2013. O algoritmo teve um bom comportamento para realização da transformada, mas apresentou problemas para encontrar a relação , que relaciona a frequência de excitação com a frequência de amostragem, podendo ser vista no Apêndice 2 - . Desta forma decidiu-se aplicar a transformada rápida de fourier via o computador, não possuindo então a necessidade de limitar o numero de operações ou o processamento. A FFT foi desenvolvida em linguagem C e pode ser vista no Apêndice 4 - Apêndice 1 - .

### **Aplificação dos sinais**

O sinal utilizado para caracterizar a impedância de contato é formada por uma somatória de sinais, contendo uma componente DC, e sinais espalhados no espectro. O sinal DC deve possui um valor de baixa amplitude para não interferiar na impedância da camada lipídica por efeito de eletroporação celular[5][11], já os sinais multifrequencias de 50kHz, 100kHz, 200kHz, 300kHz, 400kHz, 500kHz, 600kHz, 700kHz, 800kHz, 900kHz foram espalhados de tal forma para se ter uma ideia melhor do espectro, ou seja, em regiões onde exista um proeminência de capacitâncias podendo, desta forma, encontrar um equivalente de circuitos que sejá valido para o *stratum corneum*. A menor frequencia foi a de 25kHz, pois o numero de pontos para um período encaixa perfeitamente no valor do buffer da comunicação buck USB.

A mostra uma simulação de análise de bioimpedância com subamostragem para um fígado bovino utilizando algoritmo de goertzel e FFT. É possível perceber que a curva ficou bem caracterizada pois apresenta poucos pontos abaixo de 20kHz, onde há pouca variação do módulo da bioimpedância e muitos pontos na região de inflexão da curva, entre 20kHz e 2MHz. A desorganização dos dados gerados pela FFT são causados pelo subamostragem empregada, a qual não afeta o algoritmo de goertzel. Cole-Cole é o modelo do fígado bovino[14].

Figura 14 - análise frequêncial



Fonte: Gieseler, 2013

### **Conversores AD-DA**

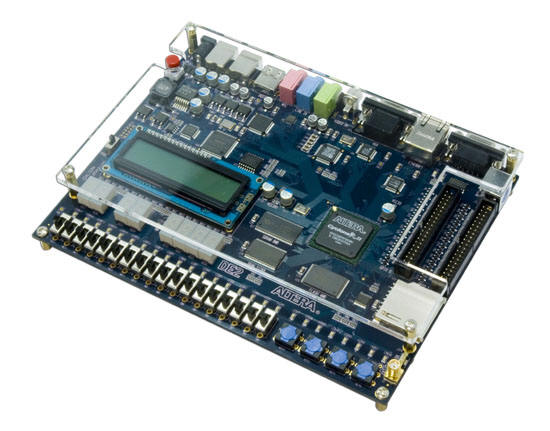
O sinal será gerado via FPGA e enviado para um conversor DA de 12 bits ad5445 [DATASHEET], O sinal passará por um amplificador com ganho variável, seguido de um filtro passa-baixas anti-aliasing em 3MHz, uma vez que a frequência máxima que se deseja injetar é de 1MHz. Em seguida o sinal é convertido novamente em forma digital por uma conversor AD de 8 bits ad9228 [datasheet] para ser tratado pela FPGA.

### **Estrutural**

## unidade de processamento

O kit de desenvolvimento utilizado por ser visto na foi fornecido pela Altera®. Ela possui comunicação USB 2.0 para comunicação com o computador para interface e dois barramentos para controle do conversor ADC, DAC e do controle de ganho.

Figura 15 - Kit de desenvolvimento DE2 da ALTERA



Fonte: Altera

Para processamento dos dados e interface gráfica será utilizado um notebook ASUS intel core i7 @ 2.30GHz, 8.00GB de RAM, Windows 8, 64bits.

## Processamento dos sinais

A função da FPGA é enviar o sinal, controlar o ADC, DAC e controle de ganho, receber o sinal e organizá-lo para envio USB. O array com os dados enviados são gerado pelo Matlab®, que cria um arquivo \*.txt com as linhas de código em VHDL para o sinal que irá conter as frequencias de interesse. O código pode ser visto no Apêndice 3 - .

O clock principal da FPGA é de 8.33MHz, mas a comunicação USB via bulk não consegue suprir essa velocidade pois trabalha com pacotes de 1024 dados a 23kHz aproximadamente. Desta forma é necessário armazenar os dados para enviá-los com a taxa de amostragem adequada, uma vez que trabalhar com subamostragem diminui a banda e pode causar problemas de Aliasing.

## testes de funcionamento e validação

O equipamento foi testato por um período de um mês, para detectar problemas de hardware e software, corrigindo pequenas falhas e bugs.

O sistema foi testado a partir de duas peças de teste, *phantom*. ambas foram produzidas utilizando um pacote de 12 gramas de gelatina sem sabor, juntamente com 70ml de água deionizada, com uma condutância de 2,5uS/cm medida pelo condutivimetro HI 8733 da empresa Hanna®. Uma delas foi feita com gelatina da marca Oetker® (L: 021215) e a outra com a marca Freischmann® (L: 141215) O experimento é mostrado na figura a seguir.

Os *phantoms* foram medidos utilizando o analisador de impedância da agilent® agilent 4294a e comparado com os valores obtidos pelo sistema desenvolvido. O resultado segue abaixo.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Frequência** | **Agilent Oetker** | **Sistema**  **Oetker** | **Agilent Freischmann** | **Sistema Freischmann** |
| 25kHz | 301 Ω | 312 Ω | 287 Ω | 313 Ω |
| 50kHz | 289 Ω | 303 Ω | 286 Ω | 311 Ω |
| 100kHz | 288 Ω | 303 Ω | 285 Ω | 30 9 Ω |
| 200kHz | 287 Ω | 295 Ω | 284 Ω | 296 Ω |
| 300kHz | 286 Ω | 308 Ω | 284 Ω | 302 Ω |
| 400kHz | 285 Ω | 300 Ω | 283 Ω | 302 Ω |
| 500kHz | 284 Ω | 294 Ω | 282 Ω | 300 Ω |
| 900kHz | 281 Ω | 282 Ω | 280 Ω | 293 Ω |

Em uma segunda rodada de testes, foram obtido os seguintes resultados

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Frequência** | **Agilent Oetker** | **Sistema**  **Oetker** | **Agilent Freischmann** | **Sistema Freischmann** |
| 50kHz | 261 Ω | 254 Ω | 295 Ω | 306 Ω |
| 100kHz | 260 Ω | 254 Ω | 294 Ω | 284 Ω |
| 200kHz | 259 Ω | 254 Ω | 292 Ω | 293 Ω |
| 300kHz | 258 Ω | 252 Ω | 291 Ω | 290 Ω |
| 400kHz | 258 Ω | 249 Ω | 291 Ω | 297 Ω |
| 500kHz | 257 Ω | 248 Ω | 290 Ω | 285 Ω |
| 600kHz | 257 Ω | 251 Ω | 289 Ω | 288 Ω |
| 700kHz | 257 Ω | 245 Ω | 289 Ω | 283 Ω |
| 800kHz | 256 Ω | 241 Ω | 288 Ω | 284 Ω |
| 900kHz | 255 Ω | 241 Ω | 287 Ω | 284 Ω |

## comitê de ética e coléta de dados

A validação da metodologia foi realizada a partir de medições com pacientes portadores de diabetes, diabetes e neuropatias ou saudáveis, analisando os efeitos na deficiência de sudorese causadas por cada um deles. Para analisar de forma fidedigna os efeitos causados pelas neuropatias, foi escolhida uma faixa etária de 20-55 anos, com pessoas de ambos os sexos, uma vez que não há diferença considerável na sudorese entre gêneros e ocorre variação típica nessa faixa de idade[14].

O ambulatório foi feito no Hospital Universitário, sob o comite de ética CAAE: 24822114.5.0000.0121. Sendo que devido a diferença de potencial de apenas 4,0 Volts aplicado entre os membros, o sistema oferece risco mínimo aos pacientes.

## ANÁLISE estatística

O último subcapítulo deve mostrar como foi feita a analise estatística, veja um exemplo:

Os resultados foram submetidos à análise de variância ANOVA. Também foi realizada a distribuição de probabilidade de Weibull, para comparar a probabilidade de falha entre os dois grupos, construindo-se o intervalo de 95% de confiança para a confiabilidade na carga máxima de 400N. Foi considerado falha, a rotura do primeiro tendão, ou seja, o ápice do primeiro grande pico. Os testes foram realizados com os programas Microsoft® Excel XP e Origin Pro® 6.1.

4

# RESULTADOS

As placas de circuito foram colocadas dentro de uma caixa plastica, onde as únicas partes de ligação externa estão na alimentação da FPGA, na comunicação USB e nos cabos para os eletrôdos de aço inoxidável.

## Analise dos sinais

5

# DISCUSSãO

Um dos problemas em utilizar a análise de sudorese é o falso positivo na detecção de neuropatia autonônica , visto que deficiências de sudorese podem estar ligados com outros problemas de saúde, como Fibrose cística, um problema genêtivo que altera o transporte de íons através das membranas celulares, doenças de pele, uso de determinados medicações ou até mesmo deficiência de vitamina D.

6

# CONCLUSÕES

Aqui deve responder as questões levantadas no objetivo. Exemplo:

Pode-se concluir que:

1. É valido a análise do valor absoluto de impedância como diagnóstico de neurpatias autonômicas
2. Derivada

## Trabalhos futuros

ReferÊncias

AYOUB, H., LAIR, V., GRIVEAU, S., BRUNSWICK, P. B. "*Electrochemical Characterization of Stainless Steel as a New Electrode Material in a Medical Device for the Diagnosis of Sudomotor Dysfunction*". Electro analysis, Vol.24 No.6, 2012.

BISHOP, D., "Biblioteca de ponto flutuante para FPGA". Disponível em http://www.vhdl.org/fphdl/Float\_ug.pdf, acesso em 12/06/2015

*BRUNSWICK, P., MAYAUDON, H., ALBIN, V., LAIR, V., RINGUEDE, A., CASSIR, M. "Use of Ni electrodes chronoamperometry for improved diagnostics of diabetes and cardiac diseases” Conference of the IEEE EMBS. 2007*

COLE, K. S. Permeability and impermeability of cell membranes for ions. In: COLD SPRING HARBOR LABORATORY PRESS. Cold Spring Harbor Symposia on Quantitative Biology. [S.l.], 1940. v. 8, p. 110-122.

CHIZMADZHEV, Y. A., INDENBOM, A. V, KUZMIN, P. I., GALICHENKO, S. V., WEAVER, J. C. POTTS, R. O. "*Electrical Properties of Skin at Moderate Voltages: Contribution of Appendageal Macropores".* Biophysical Journal. v. 74. 1998.

GIBBONS, C. H, ILLIGENS, B. M., WANG N., FREEMAN,R. "*Quantificationofsweatglandinnervation: a clinical-pathologiccorrelation. Neurology"* 1479–1486, 2009.

GIESELER, R. K., SANCHES, G. S. CUNHA, J. C., NEGRI, L. H., PATERNO, A. S., "Subamostragem e o uso do algoritmo de goertzel para otimização de espectrômetros de bioimpedância", CBEB, 2013

GRIMNES, S.; RIKSHOSPITALET, O.; SCHWAN, N. H. P. Interface Phenomena and Dielectric Properties of Biological Tissue. [S.l.]: University of Oslo, 2002.

*HUBERT, D., BRUNSWICK, P., CALVET, J., DUSSER, D., FAJAC, I. "Abnormal electrochemical skin conductance in cystic fibrosis".* Journal of Cystic Fibrosis. 2010.

*BIANCHI, C. "A sensor-based approach to safe trans-dermal iontophoresis" SAME conference. 2013.*

International Diabetes Federation. *"Diabetes surge hits every nation".* 2013

JOHNSON, H., GRAHAM, M. "High Speed Digital Design: A Handbook of Black Magic", 1st edition, 1993

Mechanism of diabetic neuropathy: Where are we

now and where to go?

Mechanism of diabetic neuropathy: Where are we

now and where to go?

KHALFALLAH, K., AYOUB, H., CALVET J. H., NEVEU, X., BRUNSWICK, P., GRIVEAU, S., LAIR, V., CASSIR, M., BEDIOUI, F. *"Non-invasive galvanic skin sensor for early diagnosis of sudomotor dysfunciton: Application to Diabetes"*. IEE SENSORS JOURNAL, v. 12, n. 3. 2012*.*

NEVES, C. E. B., LEITE, B. B., SOUZA, M. N. "Body impedance spectroscopy based on a step response", BMES/EMBS Conference, 1999

PETHIG, R. Dielectric properties of biological materials: Biophysical and medical applications. Electrical Insulation, IEEE Transactions on, 1984. IEEE, n. 5, p. 453474, 1984.

PHILLIP A. LOW, JONG-CHYOU DENQ, TONETTE L., PETER J. DYCK, PETER C. O’BRIEN, JEFF M. SLEZAK. "Effect of age and gender on sudomotor and cardiovagal function and blood pressure response to tilt in normal subjects". MUSCLE & NERVE. 1997.

ROBINSON, T., "Biblioteca USB para C#". Disponível em http://libusbdotnet.sourceforge.net/V2/Index.html, acesso em 12/06/2015

*SOLOMON, T., BOULTON, A. J. M., FREEMAN, R., HOROWITZ, M., KEMPLER, P., LAURIA, G., MALIK, R. A., SPALLONE, V., VINIK, A., BERNARDI, L., VALNSI, P.*

*SCHWAN, H. P. Electrical properties of tissue and cell suspensions. Advances in biological and medical physics, 1957. v. 5, p. 147, 1957.*

SCHWARZ, P., BRUNSWICK, P., CALVET, J.*"EZSCAN™ a new technology to detect diabetes risk".* The British Journal of Diabetes & Vascular Disease. 2011.

American Diabetes Assocition *"Diagnosis and Classification of Diabetes Mellitus". Diabetes Care. v. 37 sup. 1. 2014.*

SHENG, C., ZENG, W., HUANG, Q., DESLYPERE, J., WANG, J. *"Accuracy of a Novel Non-Invasive technology based EZSCAN system for the diagnosis of diabetes mellitus in Chinese".* Diabetology & Metabolic Syndrome. 2011.

TENTOLOURIS N. el al. Sudomotor dysfunction is associated with foot ulceration in diabetes. Diabet Med 2009;26:302–305

TESFAYE S1, BOULTON AJ, DYCK PJ, FREEMAN R, HOROWITZ M, KEMPLER P, LAURIA G, MALIK RA, SPALLONE V, VINIK A, BERNARDI L, VALENSI P. *"Diabetic Neuropathies: Update on Definitions, Diagnostic Criteria, Estimation of Severity, and Treatments ". Diabetes Journal. 2010.*

VEIGA, E. "Desenvolvimento de um sistema de bioimpedância elétrica baseado em fpga", Dissertação - UDESC, 2013.

VINIK, A. K., MASER, R. E., MITCHELLI B. D., FREEMAN. R., "*Diabetic Autonomic Neuropathy"* , Diabetes Care, Vol. 26, No.6, 2003.

WALKER, D., Brown, B. H., Hose D. R., Smallwood, R. H. "Modelling the electrical impedivity of normal and premalignant cervical tissue. Electronics Letters", 2000. IET, v. 36, n. 19, p. 16031604, 2000.

*World Health Organization <http://www.who.int/mediacentre/factsheets/fs312/en/>. Acesso em: 23 mar. 2016.*

YAGIHASHI, S. Mechanism of diabetic neuropathy: Where are we now and where to go?. Journal of Diabetes Investigation, 2011.

YÚFERO, A., RUEDA, A. "A method for bioimpedance measure with four- and two-electrode sensor systems", IEEE EMBS, 2008.

1. código VHDL principal

Código pra envio dos sinais de estimulação, controle do DAC, ADC, controle de ganho, recepção dos sinais e transmissão USB.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*LIBRARY IEEE;*

*USE IEEE.STD\_LOGIC\_1164.all;*

*USE IEEE.STD\_LOGIC\_ARITH.all;*

*USE IEEE.STD\_LOGIC\_UNSIGNED.all;*

*use work.isp\_hal.all;*

*use work.isp\_drv.all;*

*use work.devreq\_inc.all;*

*LIBRARY IEEE\_PROPOSED;*

*use IEEE\_PROPOSED.fixed\_float\_types.all;*

*use IEEE\_PROPOSED.float\_pkg.all;*

*use work.types.all; --needed for my\_library (declarations)*

*ENTITY DE2\_CLOCK IS*

*PORT(--DE2 board config USB*

*OTG\_FSPEED : out std\_logic; --USB Full Speed*

*OTG\_LSPEED : out std\_logic; --USB Low Speed*

*--ISP1368*

*OTG\_DATA : inout std\_logic\_vector(15 downto 0); --ISP1362 Data bus 16 bits*

*OTG\_INT1 : in std\_logic; --ISP1362 Interrupt 2 (Peripheral Interrupts)*

*OTG\_RST\_N : out std\_logic; --ISP1362 Reset pin*

*OTG\_ADDR : out std\_logic\_vector(1 downto 0); --ISP1362 Address 2 Bits[peripheral,command]*

*OTG\_CS\_N : out std\_logic; --ISP1362 Chip Select*

*OTG\_RD\_N : out std\_logic; --ISP1362 Write*

*OTG\_WR\_N : out std\_logic; --ISP1362 Read*

*-- not used*

*OTG\_DACK0\_N : out std\_logic; --ISP1362 DMA Acknowledge 1*

*OTG\_DACK1\_N : out std\_logic; --ISP1362 DMA Acknowledge 2*

*END DE2\_CLOCK;*

*ARCHITECTURE a OF DE2\_CLOCK IS*

*TYPE STATE\_TYPE IS (HOLD, FUNC\_SET, DISPLAY\_ON, MODE\_SET, WRITE\_CHAR1,*

*WRITE\_CHAR2,WRITE\_CHAR3, RETURN\_HOME, TOGGLE\_E, RESET1, RESET2,*

*RESET3, DISPLAY\_OFF, DISPLAY\_CLEAR); -- WRITE\_CHAR4,WRITE\_CHAR5,WRITE\_CHAR6,WRITE\_CHAR7,WRITE\_CHAR8, WRITE\_CHAR9, WRITE\_CHAR10,*

*type memory is array (INTEGER range <>) of std\_logic\_vector(15 downto 0); --Was 7 downto 0 and Send "00000000" & imp\_dummy*

*type memory1 is array (INTEGER range <>) of std\_logic\_vector(7 downto 0);*

*type memoryint is array (INTEGER range <>) of float (float\_exponent\_width downto -float\_fraction\_width);*

*--type memoryDAC is array (INTEGER range <>) of std\_logic\_vector(11 downto 0);*

*SIGNAL state, next\_command: STATE\_TYPE;*

*SIGNAL DATA\_BUS\_VALUE: STD\_LOGIC\_VECTOR(7 DOWNTO 0);*

*SIGNAL CLK\_COUNT\_400HZ: STD\_LOGIC\_VECTOR(19 DOWNTO 0);*

*SIGNAL CLK\_COUNT\_10HZ: STD\_LOGIC\_VECTOR(7 DOWNTO 0);*

*SIGNAL CLK\_COUNT\_2MHZ: STD\_LOGIC\_VECTOR(7 DOWNTO 0);*

*SIGNAL CLK\_COUNT\_12MHZ: STD\_LOGIC\_VECTOR(7 DOWNTO 0);*

*SIGNAL CLK\_COUNT\_1MHZ: STD\_LOGIC\_VECTOR(7 DOWNTO 0);*

*SIGNAL BCD\_SECD0,BCD\_SECD1,BCD\_MIND0,BCD\_MIND1: STD\_LOGIC\_VECTOR(3 DOWNTO 0);*

*SIGNAL BCD\_HRD0,BCD\_HRD1,BCD\_TSEC: STD\_LOGIC\_VECTOR(3 DOWNTO 0);*

*SIGNAL DAC\_SIGNAL: memoryDAC(0 to 4095) := ((others=> (others=>'0')));*

*SIGNAL CLK\_400HZ, CLK\_10HZ: STD\_LOGIC;*

*SIGNAL CLK\_1MHZ : std\_logic := '0';*

*SIGNAL CLK\_2MHZ : std\_logic := '0';*

*SIGNAL CLK\_12MHZ : std\_logic := '0';*

*SIGNAL ADC1: memoryint(0 to 10) := ((others=> (others=>'0')));*

*SIGNAL ADC3: memoryint(0 to 10) := ((others=> (others=>'0'))); --FOR TEST GOERTZEL*

*SIGNAL ADC2: memory1(0 to 2047) := ((others=> (others=>'0')));*

*SIGNAL COUNT\_RMS: INTEGER RANGE 0 to 10:= 0;*

*SHARED VARIABLE COUNT\_HF: INTEGER RANGE 0 to 4095:= 0;*

*SHARED VARIABLE COUNT\_HF1: INTEGER RANGE 0 to 4095:= 0;*

*CONSTANT COUNT\_RMS\_FINAL: INTEGER := 10;--24999;*

*SIGNAL r\_w ,r\_w1 : std\_logic:= '0';*

*SIGNAL DELAY\_90: INTEGER RANGE 0 to 1:= 0;*

*SIGNAL ADC1\_AVC: STD\_LOGIC\_VECTOR(2 DOWNTO 0) := "000"; --(GPIO0 (14), GPIO0 (16),GPIO0 (18) A ERROR*

*SIGNAL ADC2\_AVC: STD\_LOGIC\_VECTOR(2 DOWNTO 0) := "000"; --(GPIO1 (13), GPIO1 (15),GPIO1 (17) A*

*SIGNAL ADC3\_AVC: STD\_LOGIC\_VECTOR(2 DOWNTO 0) := "000"; --(GPIO0 (21), GPIO0 (23),GPIO0 (25) A*

*SIGNAL realW: memoryint(0 to 19);*

*SIGNAL imagW: memoryint(0 to 19);*

*SIGNAL IMP\_DUMMY: memory(0 to 5) := ((others=> (others=>'0'))); --(3999999 downto 0); -- STD\_LOGIC\_VECTOR(7 DOWNTO 0) :="00000000";*

*signal clk : std\_logic;*

*signal slowclk\_en : bit;*

*signal reset\_usb, reset\_synch : std\_logic;*

*--USB stuff*

*signal hal\_i : isp\_hal\_in\_t;*

*signal hal\_o : isp\_hal\_out\_t;*

*signal drv\_i : isp\_drv\_in\_t;*

*signal drv\_o : isp\_drv\_out\_t;*

*type stateT is (loopOuter, loopInner);*

*signal state\_usb: stateT;*

*signal counter, innerCounter, counter1: integer := 0;*

*signal sending: std\_logic := '0';*

*shared Variable COUNT\_DAC: INTEGER RANGE 0 to 4095:= 0;*

*component my\_library*

*port (*

*DAC\_SIGNAL: OUT memoryDAC(0 to 4095);-- := ((others=> (others=>'0')));*

*RESET : IN STD\_LOGIC*

*);*

*end component;*

*BEGIN*

*A1: my\_library port map(*

*DAC\_SIGNAL => DAC\_SIGNAL,*

*RESET => RESET*

*);*

*(GPIO0 (13), GPIO0 (15),GPIO0 (17)) <= ADC1\_AVC; --Controle de ganho*

*(GPIO1 (19), GPIO1 (21),GPIO1 (23)) <= ADC2\_AVC;*

*(GPIO1 (11), GPIO1 (13),GPIO1 (15)) <= ADC3\_AVC;*

*GPIO0 (12) <= CLK\_1MHZ; --CS = 0 Enable DAC read // Must be pulsed to load new data*

*GPIO0 (14) <= r\_w;--'0'; --CLK\_1MHZ; --R/W = 1 Enable DAC read // 0 to load new data*

*GPIO0 (16) <= r\_w1;--'0';*

*GPIO1 (8) <= CLK\_12MHZ; --DCLK ADC*

*-------------------------------USB------------------------------*

*--DE2 USB config*

*OTG\_FSPEED <='0'; -- 0 = Enable, Z = Disable*

*OTG\_LSPEED <='Z'; -- 0 = Enable, Z = Disable*

*--DE2 clock and reset config*

*clk <= clk\_50Mhz;*

*p\_reset: process*

*begin*

*wait until rising\_edge(clk);*

*reset\_synch <= not(RESET);*

*reset\_usb <= reset\_synch;*

*end process;*

*p\_pulse25MHz: process--produces 25MHz clock enable pulse for HAL*

*begin*

*WAIT UNTIL CLK\_50MHZ'EVENT AND CLK\_50MHZ = '1';*

*slowclk\_en <= not(slowclk\_en);*

*end process;*

*-- hal port map*

*hal\_i.slowclk\_en <= slowclk\_en;*

*OTG\_RST\_N <= hal\_o.rst\_n;*

*OTG\_ADDR <= hal\_o.addr;*

*OTG\_CS\_N <= hal\_o.cs\_n;*

*OTG\_RD\_N <= hal\_o.rd\_n;*

*OTG\_WR\_N <= hal\_o.wr\_n;*

*OTG\_DACK0\_N <= hal\_o.dack0\_n;*

*OTG\_DACK1\_N <= hal\_o.dack1\_n;*

*--hal*

*hal\_i.drv <= drv\_o.hal;*

*h: hal port map (clk, reset\_usb, OTG\_DATA , hal\_i, hal\_o);*

*--driver port map*

*drv\_i.int <= OTG\_INT1; --IF LOOPBACK, DON'T NEED*

*--driver*

*drv\_i.hal <= hal\_o.drv;*

*d: drv port map(clk,reset\_usb, drv\_i , drv\_o);*

*--device request handler*

*dvrq: devreq port map(clk, reset\_usb, drv\_o.devreq, drv\_i.devreq);*

*demo\_send: process(clk, SW)*

*VARIABLE a: integer range 0 to 1 := 0;*

*VARIABLE k: integer range 0 to 3 := 0;*

*VARIABLE dummy\_std: STD\_LOGIC\_VECTOR(15 DOWNTO 0);*

*begin*

*if rising\_edge(clk) then*

*if drv\_o.io.Rdy = '1' OR innerCounter > 0 OR COUNT\_HF1 > 0 then*

*if counter < 15 then*

*drv\_i.io.Sdata(drv\_o.io.Sdata'high downto 0) <= "00000000" & ADC2(COUNT\_HF1);*

*if innerCounter = 0 then*

*drv\_i.io.Rdy <= '1';*

*innerCounter <= innerCounter + 1;*

*elsif innerCounter > 0 and innerCounter < 1850 then*

*drv\_i.io.Rdy <= '0';*

*innerCounter <= innerCounter + 1;*

*elsif innerCounter = 1850 then*

*innerCounter <= 0;*

*counter <= counter + 1;*

*COUNT\_HF1 := COUNT\_HF1+1;*

*IF COUNT\_HF1=1000 THEN*

*COUNT\_HF1 := 0;*

*ELSE*

*COUNT\_HF1 := COUNT\_HF1 +1;*

*END IF;*

*end if;*

*elsif counter = 15 then*

*counter <= 0;*

*innerCounter <= 0;*

*drv\_i.io.Rdy <= '0';*

*LEDR(drv\_o.io.Sdata'high downto 0) <= "0000000000000000";*

*drv\_i.io.Sdata(drv\_o.io.Sdata'high downto 0) <= (others => '0');*

*end if;*

*else*

*counter <= 0;*

*innerCounter <= 0;*

*drv\_i.io.Rdy <= '0';*

*drv\_i.io.Sdata(drv\_o.io.Sdata'high downto 0) <= (others => '0');*

*end if;*

*end if;*

*end process;*

*-------------------------------USB------------------------------*

*PROCESS*

*BEGIN*

*WAIT UNTIL CLK\_50MHZ'EVENT AND CLK\_50MHZ = '1';*

*IF RESET = '0' THEN*

*CLK\_COUNT\_400HZ <= X"00000";*

*CLK\_400HZ <= '0';*

*ELSE*

*IF CLK\_COUNT\_400HZ < X"0F424" THEN*

*CLK\_COUNT\_400HZ <= CLK\_COUNT\_400HZ + 1;*

*ELSE*

*CLK\_COUNT\_400HZ <= X"00000";*

*CLK\_400HZ <= NOT CLK\_400HZ;*

*END IF;*

*END IF;*

*END PROCESS;*

*PROCESS*

*BEGIN*

*WAIT UNTIL CLK\_50MHZ'EVENT AND CLK\_50MHZ = '1';*

*IF RESET = '0' THEN*

*CLK\_COUNT\_12MHZ <= X"00";*

*CLK\_12MHZ <= '0';*

*ELSE*

*IF CLK\_COUNT\_12MHZ < x"01" THEN -- "03" - 8,33Mhz=50/6*

*CLK\_COUNT\_12MHZ <= CLK\_COUNT\_12MHZ + 1;*

*ELSE*

*CLK\_COUNT\_12MHZ <= X"00";*

*CLK\_12MHZ <= NOT CLK\_12MHZ;*

*END IF;*

*END IF;*

*END PROCESS;*

*PROCESS*

*BEGIN*

*WAIT UNTIL CLK\_50MHZ'EVENT AND CLK\_50MHZ = '1';*

*IF RESET = '0' THEN*

*CLK\_COUNT\_2MHZ <= X"00";*

*CLK\_2MHZ <= '0';*

*ELSE*

*IF CLK\_COUNT\_2MHZ < x"05" THEN -- "00" - 25Mhz "01" - 12,5Mhz "02" - 12,5Mhz "03" - 8,33Mhz=50/6 "04" - 50/8 Mhz! -- 2.275 - "0A"*

*CLK\_COUNT\_2MHZ <= CLK\_COUNT\_2MHZ + 1;*

*ELSE*

*CLK\_COUNT\_2MHZ <= X"00";*

*CLK\_2MHZ <= NOT CLK\_2MHZ;*

*END IF;*

*END IF;*

*END PROCESS;*

*PROCESS*

*BEGIN*

*WAIT UNTIL CLK\_50MHZ'EVENT AND CLK\_50MHZ = '1';*

*IF RESET = '0' THEN*

*CLK\_COUNT\_1MHZ <= X"00";*

*CLK\_1MHZ <= '0';*

*ELSE*

*IF CLK\_COUNT\_1MHZ < x"02" THEN --2Mhz 0,568 - "15"*

*CLK\_COUNT\_1MHZ <= CLK\_COUNT\_1MHZ + 1;*

*ELSE*

*CLK\_COUNT\_1MHZ <= X"00";*

*CLK\_1MHZ <= NOT CLK\_1MHZ;*

*END IF;*

*END IF;*

*END PROCESS;*

*PROCESS (CLK\_400HZ, reset)*

*BEGIN*

*IF reset = '0' THEN*

*state <= RESET1;*

*CLK\_COUNT\_10HZ <= X"00";*

*CLK\_10HZ <= '0';*

*ELSIF CLK\_400HZ'EVENT AND CLK\_400HZ = '1' THEN*

*IF CLK\_COUNT\_10HZ < 19 THEN*

*CLK\_COUNT\_10HZ <= CLK\_COUNT\_10HZ + 1;*

*ELSE*

*CLK\_COUNT\_10HZ <= X"00";*

*CLK\_10HZ <= NOT CLK\_10HZ;*

*END IF;*

*END IF;*

*END PROCESS;*

*(GPIO0 (11), GPIO0 (9), GPIO0 (7), GPIO0 (5), GPIO0 (3), GPIO0 (1), GPIO0 (0), GPIO0 (2), GPIO0 (4),GPIO0 (6),GPIO0 (8), GPIO0 (10)) <= DAC\_SIGNAL(COUNT\_DAC); --Maybe overhead outside the for*

*PROCESS (CLK\_50MHZ,RESET )*

*--VARIABLE COUNT\_DAC: INTEGER RANGE 0 to 146:= 0;*

*BEGIN*

*IF CLK\_2MHZ'EVENT AND CLK\_2MHZ = '0' THEN*

*IF RESET = '0' THEN*

*COUNT\_HF := 0;*

*ELSE*

*IF COUNT\_HF1 = 0 THEN*

*COUNT\_HF := 0;*

*ELSIF COUNT\_HF = 2048 THEN*

*COUNT\_HF := 2048;*

*ELSE*

*COUNT\_HF := COUNT\_HF+1;*

*END IF;*

*ADC2(COUNT\_HF) <= GPIO1 (16) & GPIO1 (18) & GPIO1 (20) & GPIO1 (22) & GPIO1 (24) & GPIO1 (26) & GPIO1 (28) & GPIO1 (30);--ADC2 <= GPIO1 (24) & GPIO1 (26) & GPIO1 (28) & GPIO1 (30) & GPIO1 (32) & GPIO1 (34) & GPIO1 (35) & GPIO1 (33); --IMP\_DUMMY(0) <= GPIO1 (25) & GPIO1 (27) & GPIO1 (29) & GPIO1 (31) & GPIO1 (33) & GPIO1 (35) & GPIO1 (34) & GPIO1 (32); --ADC2(COUNT\_RMS) <= GPIO1 (24) & GPIO1 (26) & GPIO1 (28) & GPIO1 (30) & GPIO1 (32) & GPIO1 (34) & GPIO1 (35) & GPIO1 (33); --OK --ADC2(COUNT\_RMS) <= GPIO1 (11) & GPIO1 (9) & GPIO1 (7) & GPIO1 (5) & GPIO1 (3) & GPIO1 (1) & GPIO1 (0) & GPIO1 (2);*

*END IF;*

*END IF;*

*END PROCESS;*

*PROCESS (CLK\_50MHZ, reset)*

*--VARIABLE COUNT\_DAC: INTEGER RANGE 0 to 146:= 0;*

*BEGIN*

*IF CLK\_1MHZ'EVENT AND CLK\_1MHZ = '0' THEN*

*IF RESET = '0' THEN*

*COUNT\_DAC := 0;*

*ELSE --IF DELAY\_90 = 0 THEN*

*COUNT\_DAC := COUNT\_DAC +1;*

*IF COUNT\_DAC = 3332 THEN*

*COUNT\_DAC := 0;*

*END IF;*

*r\_w1 <= NOT r\_w1;*

*END IF;*

*END IF;*

*END PROCESS;*

*END a;*

1. código VHDL para algoritmo de goertzel

Código para transformada de goertzel em FPGA.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*PROCESS (CLK\_50MHZ, reset) --*

*VARIABLE d1\_1: memoryint(0 to 5) := ((others=> (others=>'0')));--float (float\_exponent\_width downto -float\_fraction\_width) := (others=>'0');*

*VARIABLE d1\_2: memoryint(0 to 5) := ((others=> (others=>'0')));--float (float\_exponent\_width downto -float\_fraction\_width) := (others=>'0');*

*VARIABLE y: memoryint(0 to 5) := ((others=> (others=>'0')));--float (float\_exponent\_width downto -float\_fraction\_width) := (others=>'0');*

*VARIABLE x: memoryint(0 to 5) := ((others=> (others=>'0')));*

*BEGIN*

*IF RESET = '0' THEN*

*COUNT\_RMS <= 0;*

*ELSIF CLK\_1MHZ'EVENT AND CLK\_1MHZ = '1' THEN --400HZ overhead WTF? NEED THE ELSIF*

*IF COUNT\_RMS = COUNT\_RMS\_FINAL THEN*

*x(0) := resize((sqrt((realW(0)\*d1\_1(0)\*to\_float(0.5) - d1\_2(0)) \* (realW(0)\*d1\_1(0)\*to\_float(0.5) - d1\_2(0))+(imagW(0)\*d1\_1(0))\*(imagW(0)\*d1\_1(0)))),y(0));*

*IMP\_DUMMY(0) <= std\_logic\_vector(to\_unsigned(x(0),16,round\_nearest)); --Was 8 bits*

*d1\_1 := ((others=> (others=>'0')));*

*d1\_2 := ((others=> (others=>'0')));*

*y := ((others=> (others=>'0')));*

*COUNT\_RMS <= 0;*

*ELSE*

*d1\_2 := d1\_1;*

*d1\_1 := y;*

*END IF;*

*END IF;*

*END PROCESS;*

1. código matlab

Código para gerar o sinal de excitação que será enviado pela FPGA

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*clear all*

*close all*

*clc*

*Fs=[8.33e6 27.000e3];%[1.956e6 12785 25585 51185 102385 204785 409585 819185 1638385]; %2\*k-15 Satisfatório para B=10;*

*fbase =5e3;*

*k = [ 10 \* fbase 20 \* fbase 40 \* fbase 60 \* fbase 80 \* fbase 100 \* fbase 120 \* fbase 140 \* fbase 160 \* fbase 180 \* fbase]; %600/385*

*Tfinal=2/fbase - 1/Fs(1);*

*z=1; %quantidade de taxas de amostragem*

*t=0:1/Fs(z):Tfinal;*

*x(1,1:length(t))=0;*

*f=[0:Fs(z)/length(t):Fs(z)-Fs(z)/length(t)];*

*N=length(t);*

*Fx=[0:Fs(z)/N:Fs(z)-Fs(z)/N];*

*for j=1:length(k)*

*realW(j) = 2.0.\*cos(2.0.\*pi.\*k(j)./Fs(z)); %Goertzel*

*imagW(j) = sin(2.0\*pi\*k(j)./Fs(z));*

*x(1,:)=(0.5\*cos(2\*pi\*k(j).\*t))+x(1,:); %Corrente injetada*

*end*

*x(1,:)=x(1,:)-min(x)+4; %Nível DC*

*corrente\_t=x(1,:);*

*figure*

*subplot(1,1,1); plot(t,corrente\_t);*

*hold on*

*subplot(1,1,1); plot(Tfinal+t,corrente\_t);*

*%%% INICIA GOERTZEL %%%*

*for j=1:length(k)*

*d1(1) = 0.0;*

*d2(1) = 0.0;*

*y=0;*

*for n = 0: N-1*

*y = corrente\_t(n+1) + realW(j)\*d1(1) - d2(1);*

*d2(1) = d1(1);*

*d1(1) = y;*

*end*

*corrente(j)=sqrt((0.5\*realW(j)\*d1(1) - d2(1))^2+(imagW(j)\*d1(1))^2);*

*end*

*%%% FIM GOERTZEL %%%*

*corrente\_fft=fft(corrente\_t,1\*N);*

*realifft = ifft(real(corrente\_fft));*

*figure*

*subplot(1,1,1); plot(f,abs(corrente\_fft));*

*hold on*

*subplot(1,1,1); plot(k,corrente,'bs');*

*title('FFT Corrente');*

*format long, x;*

*fide = fopen('dados.txt', 'w');*

*%fprintf(fide, 'Frequencia Goertzel FFTorg FFT Imp\_RC imp\_Cole Fs=%iHz',Fs);*

*for i=1:length(x)*

*fprintf(fide,'\r\nDAC\_SIGNAL(%i) <= conv\_std\_logic\_vector(%i,12) WHEN RESET = ''1'' ELSE (others=>''0'');',i-1, round(x(i)\*4095/max(x)));%(length(k)\*2)));*

*end*

*fclose(fide);*

1. código c# para processamento e interface

Código para receber dados via USB, transformada de Fourier e interface gráfica

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*using System;*

*using System.Collections.Generic;*

*using System.ComponentModel;*

*using System.Data;*

*using System.Drawing;*

*using System.Linq;*

*using System.Text;*

*using System.Text.RegularExpressions;*

*using System.Windows.Forms;*

*using LibUsbDotNet;*

*using LibUsbDotNet.Main;*

*using System.Drawing.Printing;*

*namespace Examples*

*{*

*public partial class Form1 : Form*

*{*

*public static UsbDevice MyUsbDevice;*

*#region SET YOUR USB Vendor and Product ID!*

*public static UsbDeviceFinder MyUsbFinder = new UsbDeviceFinder(0x04cc, 0x1a62);*

*#endregion*

*public Form1()*

*{*

*InitializeComponent();*

*}*

*private Button printButton = new Button();*

*private PrintDocument printDocument1 = new PrintDocument();*

*private void Form1\_Load(object sender, EventArgs e)*

*{*

*printButton.Text = "Print Form";*

*printButton.Click += new EventHandler(printButton\_Click);*

*printDocument1.PrintPage += new PrintPageEventHandler(printDocument2\_PrintPage);*

*this.Controls.Add(printButton);*

*// label1.Text = "Hello World";*

*}*

*private bool \_stopLoop;*

*private void button1\_Click(object sender, EventArgs e)*

*{*

*\_stopLoop = false;*

*this.button1.Enabled = false;*

*this.button2.Enabled = true;*

*float fbase = 5000;*

*float[] k = { 10 \* fbase, 20 \* fbase, 40 \* fbase, 60 \* fbase, 80 \* fbase, 100 \* fbase, 120 \* fbase, 140 \* fbase, 160 \* fbase, 180 \* fbase};*

*double[] k\_ajust1\_0kant = { -0.186, -0.416, -0.524, -0.804, -0.603, -1.140, -1.358,0,0,0,0,0};*

*double[] k\_ajust0kant = { -0.18, -0.24, -0.31, -0.48, -0.35, -0.77, -0.93, 0, 0, 0, 0, 0 };*

*double[] k\_ajust1 = { -0.23, -0.45, -0.86, -1.67, - 2.31, -2.8, -3.1, -3.5, 0, 0, 0, 0 };*

*double[] k\_ajust = { 2.103 / (3.1 + k\_ajust1[0]), 2.103 / (3 + k\_ajust1[1]), 2.103 / (2.9 + k\_ajust1[2]), 2.103 / (3.1 + k\_ajust1[3]), 2.103 / (3.5+ k\_ajust1[4]), 2.103 / (3.8 + k\_ajust1[5]), 2.103 / (4.0 + k\_ajust1[6]), 2.103 / (4.11 + k\_ajust1[7]),0,0,0 };*

*double[] k\_ajust2 = { 2.45 / 2.5, 2.45 / 0.82, 2.45 / 2.45, 2.45 / 0.5, 2.45 / 1.77, 2.45 / 0.82, 2.45 / 1.77, 2.45 / 2.54, 2.45 / 2.83, 2.45 / 0.81, 2.45 / 1.76, 2.45 / 2.54 };*

*double[] d1 = new double[k.Length];*

*double[] d2 = new double[k.Length];*

*double[] y = new double[k.Length];*

*int rfinal = 2700;*

*double[,] Fourier = new double[k.Length, rfinal];*

*double[,] Fourier\_imp = new double[k.Length, rfinal];*

*double[,] Goertzelfft = new double[k.Length, 20];*

*int[] counts = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*double[] Dados1 = new double[65536];*

*double[] Dados2 = new double[65536];*

*double[] Dados3 = new double[65536];*

*double[] SumX = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }; //int Count = BytesReadWrite/2;*

*double[] SumX2 = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*double[] SumY = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*double[] SumXY = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*chart3.Series["25kHz"].Points.Clear();*

*chart3.Series["50kHz"].Points.Clear();*

*chart3.Series["100kHz"].Points.Clear();*

*chart3.Series["150kHz"].Points.Clear();*

*chart3.Series["200kHz"].Points.Clear();*

*chart3.Series["250kHz"].Points.Clear();*

*chart3.Series["300kHz"].Points.Clear();*

*chart3.Series["350kHz"].Points.Clear();*

*chart3.Series["400kHz"].Points.Clear();*

*chart3.Series["450kHz"].Points.Clear();*

*chart3.Series["500kHz"].Points.Clear();*

*chart3.Series["550kHz"].Points.Clear();*

*chart3.Series["600kHz"].Points.Clear();*

*chart3.Series["650kHz"].Points.Clear();*

*chart3.Series["700kHz"].Points.Clear();*

*chart3.Series["750kHz"].Points.Clear();*

*chart3.Series["800kHz"].Points.Clear();*

*chart3.Series["850kHz"].Points.Clear();*

*chart3.Series["900kHz"].Points.Clear();*

*chart3.Titles.Clear();*

*System.IO.File.Delete(@"C:\Users\Admin\Desktop\Resultado.txt");*

*for (int r = 0; r < rfinal && !\_stopLoop; r++)*

*{*

*label2.Text = "Em operação: interação " + Convert.ToString(r) + " de " + Convert.ToString(rfinal);*

*ErrorCode ec = ErrorCode.None;*

*//DrawIt();*

*try*

*{*

*// Find and open the usb device.*

*MyUsbDevice = UsbDevice.OpenUsbDevice(MyUsbFinder);*

*// If the device is open and ready*

*if (MyUsbDevice == null) throw new Exception("Device Not Found.");*

*//If not (WinUSB) the*

*// variable will be null indicating this is an interface of a*

*// device.*

*IUsbDevice wholeUsbDevice = MyUsbDevice as IUsbDevice;*

*if (!ReferenceEquals(wholeUsbDevice, null))*

*{*

*// Select config #1*

*wholeUsbDevice.SetConfiguration(1);*

*// Claim interface #0.*

*wholeUsbDevice.ClaimInterface(0);*

*}*

*// open read endpoint 1.*

*UsbEndpointReader reader = MyUsbDevice.OpenEndpointReader(ReadEndpointID.Ep02);*

*// open write endpoint 1.*

*UsbEndpointWriter writer = MyUsbDevice.OpenEndpointWriter(WriteEndpointID.Ep01);*

*// Remove the exepath/startup filename text from the begining of the CommandLine.*

*string cmdLine = Regex.Replace(*

*Environment.CommandLine, "^\".+?\"^.\*? |^.\*? ", "", RegexOptions.Singleline);*

*//if (!String.IsNullOrEmpty(cmdLine))*

*//{*

*int BytesReadWrite = 1000;*

*// int Freqs = 20;*

*int bytesWritten;*

*byte[] bytesWritten1 = new byte[BytesReadWrite\*3];*

*for (int i = BytesReadWrite - 1; i >= 0; i--)*

*{*

*bytesWritten1[i] = 1;//(byte)i;*

*}*

*ec = writer.Write(bytesWritten1, 100, out bytesWritten);*

*byte[] readBuffer = new byte[BytesReadWrite];*

*int bytesRead;*

*double[] Dado1 = new double[BytesReadWrite/2];*

*double[] Dado2 = new double[BytesReadWrite/2];*

*double[] Dado3 = new double[BytesReadWrite/2];*

*float prec = 1;*

*chart1.Series["Fourier"].Points.Clear();*

*chart1.Series["Interpolação"].Points.Clear();*

*chart1.Titles.Clear();*

*chart2.Series["Abs"].Points.Clear();*

*chart2.Titles.Clear();*

*int[] count = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 }; //Teste Goertzel*

*int start\_time = DateTime.Now.Millisecond;*

*ec = reader.Read(readBuffer, 100, out bytesRead);*

*int elapsed\_time = DateTime.Now.Millisecond - start\_time; //Takes 80ms until the end of the FORs... Way down below*

*count[0] = 0;*

*for (int i = 0; i < BytesReadWrite-1; i = i + 2) //THE LIBRARY READS 32 DATA, BUT FPGA SENDS ONLY 16, SO HALF IS JUST ZEROES (-8 was 4)*

*{*

*chart1.Series["Fourier"].Points.Clear();*

*Dado1[count[0]] = (readBuffer[i + 1] \* 256 + readBuffer[i]) / prec;*

*Dados1[counts[0]] = (((readBuffer[i + 1] \* 256 + readBuffer[i + 0]) / prec) + Dados1[counts[0]] \* (counts[0])) / (counts[0] + 1);*

*count[0]++;*

*chart2.Series["Abs"].Points.AddXY(count[0], (readBuffer[i + 1] \* 256 + readBuffer[i]) / prec);*

*}*

*chart1.Titles.Add("Avaliação");*

*double[] imagW = new double[k.Length];*

*double[] realW = new double[k.Length];*

*double[] Goertzel = new double[k.Length];*

*double DataMean = Dado1.Sum() / (BytesReadWrite/8-4);*

*int loc;*

*double[] result = dft(Dado1);*

*int loc1;*

*for (int o = 1; o < BytesReadWrite/2-1; o++) //BytesReadWrite/8- 4*

*{*

*chart1.Series["Fourier"].Points.AddXY(o \*2\* 2.083e6 / BytesReadWrite, result[o]);*

*}*

*for (int jj = 0; jj < k.Length; jj++)*

*{*

*loc = (int)Math.Round(((k[jj] / (2.083e6) - Math.Floor(k[jj] / (2.083e6))) \* (BytesReadWrite / 2 - 1)));*

*loc1 = loc-3;*

*if(loc1<0)*

*{*

*loc1 = 0;*

*loc = 4;*

*}*

*for (int oh = 2; oh > -3; oh--) //BytesReadWrite/8- 4*

*{*

*if (result[loc - oh] > result[loc1])*

*{*

*loc1 = loc - oh;*

*}*

*}*

*loc = loc1;*

*if (r != 0)*

*{*

*Goertzelfft[jj, 18] = Goertzelfft[jj, 17];*

*Goertzelfft[jj, 17] = Goertzelfft[jj, 16];*

*Goertzelfft[jj, 16] = Goertzelfft[jj, 15];*

*Goertzelfft[jj, 15] = Goertzelfft[jj, 14];*

*Goertzelfft[jj, 14] = Goertzelfft[jj, 13];*

*Goertzelfft[jj, 13] = Goertzelfft[jj, 12];*

*Goertzelfft[jj, 12] = Goertzelfft[jj, 11];*

*Goertzelfft[jj, 11] = Goertzelfft[jj, 10];*

*Goertzelfft[jj, 10] = Goertzelfft[jj, 9];*

*Goertzelfft[jj, 9] = Goertzelfft[jj, 8];*

*Goertzelfft[jj, 8] = Goertzelfft[jj, 7];*

*Goertzelfft[jj, 7] = Goertzelfft[jj, 6];*

*Goertzelfft[jj, 6] = Goertzelfft[jj, 5];*

*Goertzelfft[jj, 5] = Goertzelfft[jj, 4];*

*Goertzelfft[jj, 4] = Goertzelfft[jj, 3];*

*Goertzelfft[jj,3] = Goertzelfft[jj,2];*

*Goertzelfft[jj,2] = Goertzelfft[jj,1];*

*Goertzelfft[jj,1] = Goertzelfft[jj,0];*

*Goertzelfft[jj,0] = result[loc];*

*}*

*else*

*{*

*Goertzelfft[jj, 18] = result[loc];*

*Goertzelfft[jj, 17] = result[loc];*

*Goertzelfft[jj, 16] = result[loc];*

*Goertzelfft[jj, 15] = result[loc];*

*Goertzelfft[jj, 14] = result[loc];*

*Goertzelfft[jj, 13] = result[loc];*

*Goertzelfft[jj, 12] = result[loc];*

*Goertzelfft[jj, 11] = result[loc];*

*Goertzelfft[jj, 10] = result[loc];*

*Goertzelfft[jj, 9] = result[loc];*

*Goertzelfft[jj, 8] = result[loc];*

*Goertzelfft[jj, 7] = result[loc];*

*Goertzelfft[jj, 6] = result[loc];*

*Goertzelfft[jj, 5] = result[loc];*

*Goertzelfft[jj, 4] = result[loc];*

*Goertzelfft[jj,3] = result[loc];*

*Goertzelfft[jj,2] = result[loc];*

*Goertzelfft[jj,1] = result[loc];*

*Goertzelfft[jj,0] = result[loc];*

*}*

*Fourier[jj, r] = (Goertzelfft[jj, 18] + Goertzelfft[jj, 17] + Goertzelfft[jj, 16] + Goertzelfft[jj, 15] + Goertzelfft[jj, 14] + Goertzelfft[jj, 13] + Goertzelfft[jj, 12] + Goertzelfft[jj, 11] + Goertzelfft[jj, 10] + Goertzelfft[jj, 9] + Goertzelfft[jj, 8] + Goertzelfft[jj, 7] + Goertzelfft[jj, 6] + Goertzelfft[jj, 5] + Goertzelfft[jj, 4] + Goertzelfft[jj, 3] + Goertzelfft[jj, 2] + Goertzelfft[jj, 1] + Goertzelfft[jj, 0]) / 19;*

*chart1.Series["Interpolação"].Points.AddXY(k[jj], Fourier[jj, r]);*

*}*

*Fourier\_imp[0, r] = 1275 / Fourier[0, r] - 85;*

*Fourier\_imp[1, r] = 1285 / Fourier[1, r] - 104;*

*Fourier\_imp[2, r] = 1327 / Fourier[2, r] - 107;*

*Fourier\_imp[3, r] = 1344 / Fourier[3, r] - 105;*

*Fourier\_imp[4, r] = 1375 / Fourier[4, r] - 106;*

*Fourier\_imp[5, r] = 1384 / Fourier[5, r] - 104;*

*Fourier\_imp[6, r] = 1369 / Fourier[6, r] - 101;*

*Fourier\_imp[7, r] = 1333 / Fourier[7, r] - 93;*

*Fourier\_imp[8, r] = 1342 / Fourier[8, r] - 108;*

*Fourier\_imp[9, r] = 1245 / Fourier[9, r] - 98;*

*chart3.Series["50kHz"].Points.AddXY(r , Fourier\_imp[0, r]);*

*chart3.Series["100kHz"].Points.AddXY(r, Fourier\_imp[1, r]);*

*chart3.Series["200kHz"].Points.AddXY(r, Fourier\_imp[2, r]);*

*chart3.Series["300kHz"].Points.AddXY(r, Fourier\_imp[3, r]);*

*chart3.Series["400kHz"].Points.AddXY(r, Fourier\_imp[4, r]);*

*chart3.Series["500kHz"].Points.AddXY(r, Fourier\_imp[5, r]);*

*chart3.Series["600kHz"].Points.AddXY(r, Fourier\_imp[6, r]);*

*chart3.Series["700kHz"].Points.AddXY(r, Fourier\_imp[7, r]);*

*chart3.Series["800kHz"].Points.AddXY(r, Fourier\_imp[8, r]);*

*chart3.Series["900kHz"].Points.AddXY(r, Fourier\_imp[9, r]);*

*}*

*catch (Exception ex)*

*{*

*label2.Text = (ec != ErrorCode.None ? ec + ":" : String.Empty) + ex.Message;*

*}*

*finally*

*{*

*if (MyUsbDevice != null)*

*{*

*if (MyUsbDevice.IsOpen)*

*{*

*IUsbDevice wholeUsbDevice = MyUsbDevice as IUsbDevice;*

*if (!ReferenceEquals(wholeUsbDevice, null))*

*{*

*wholeUsbDevice.ReleaseInterface(0);*

*}*

*MyUsbDevice.Close();*

*}*

*MyUsbDevice = null;*

*UsbDevice.Exit();*

*}*

*}*

*if (counts[0] != 65535)*

*{*

*counts[0]++;*

*}*

*double[] XMean = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*double[] YMean = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*double[] Slope = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*double[] YInt = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };*

*if (r > 17)*

*{*

*int r\_18 = r - 17;*

*SumX[0] = SumX[0] + r\_18;*

*SumY[0] = SumY[0] + (Fourier\_imp[0, r]);*

*SumX2[0] = SumX2[0] + r\_18 \* r\_18;*

*SumXY[0] = SumXY[0] + r\_18 \* (Fourier\_imp[0, r]);*

*SumX[1] = SumX[1] + r\_18;*

*SumY[1] = SumY[1] + (Fourier\_imp[1, r]);*

*SumX2[1] = SumX2[1] + r\_18 \* r\_18;*

*SumXY[1] = SumXY[1] + r\_18 \* (Fourier\_imp[1, r]);*

*SumX[2] = SumX[2] + r\_18;*

*SumY[2] = SumY[2] + (Fourier\_imp[2, r]);*

*SumX2[2] = SumX2[2] + r\_18 \* r\_18;*

*SumXY[2] = SumXY[2] + r\_18 \* (Fourier\_imp[2, r]);*

*SumY[3] = SumY[3] + (Fourier\_imp[3, r]);*

*SumXY[3] = SumXY[3] + r\_18 \* (Fourier\_imp[3, r]);*

*SumY[4] = SumY[4] + (Fourier\_imp[4, r]);*

*SumXY[4] = SumXY[4] + r\_18 \* (Fourier\_imp[4, r]);*

*SumY[5] = SumY[5] + (Fourier\_imp[5, r]);*

*SumXY[5] = SumXY[5] + r\_18 \* (Fourier\_imp[5, r]);*

*SumY[6] = SumY[6] + (Fourier\_imp[6, r]);*

*SumXY[6] = SumXY[6] + r\_18 \* (Fourier\_imp[6, r]);*

*SumY[7] = SumY[7] + (Fourier\_imp[7, r]);*

*SumXY[7] = SumXY[7] + r\_18 \* (Fourier\_imp[7, r]);*

*SumY[8] = SumY[8] + (Fourier\_imp[8, r]);*

*SumXY[8] = SumXY[8] + r\_18 \* (Fourier\_imp[8, r]);*

*SumY[9] = SumY[9] + (Fourier\_imp[9, r]);*

*SumXY[9] = SumXY[9] + r\_18 \* (Fourier\_imp[9, r]);*

*XMean[0] = SumX[0] / r\_18;*

*YMean[0] = SumY[0] / r\_18;*

*Slope[0] = (SumXY[0] - SumX[0] \* YMean[0]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[0] = YMean[0] - Slope[0] \* XMean[0];*

*YMean[1] = SumY[1] / r\_18;*

*Slope[1] = (SumXY[1] - SumX[0] \* YMean[1]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[1] = YMean[1] - Slope[1] \* XMean[0];*

*YMean[2] = SumY[2] / r\_18;*

*Slope[2] = (SumXY[2] - SumX[0] \* YMean[2]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[2] = YMean[2] - Slope[2] \* XMean[0];*

*YMean[3] = SumY[3] / r\_18;*

*Slope[3] = (SumXY[3] - SumX[0] \* YMean[3]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[3] = YMean[3] - Slope[3] \* XMean[0];*

*YMean[4] = SumY[4] / r\_18;*

*Slope[4] = (SumXY[4] - SumX[0] \* YMean[4]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[4] = YMean[4] - Slope[4] \* XMean[0];*

*YMean[5] = SumY[5] / r\_18;*

*Slope[5] = (SumXY[5] - SumX[0] \* YMean[5]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[5] = YMean[5] - Slope[5] \* XMean[0];*

*YMean[6] = SumY[6] / r\_18;*

*Slope[6] = (SumXY[6] - SumX[0] \* YMean[6]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[6] = YMean[6] - Slope[6] \* XMean[0];*

*YMean[7] = SumY[7] / r\_18;*

*Slope[7] = (SumXY[7] - SumX[0] \* YMean[7]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[7] = YMean[7] - Slope[7] \* XMean[0];*

*YMean[8] = SumY[8] / r\_18;*

*Slope[8] = (SumXY[8] - SumX[0] \* YMean[8]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[8] = YMean[8] - Slope[8] \* XMean[0];*

*YMean[9] = SumY[9] / r\_18;*

*Slope[9] = (SumXY[9] - SumX[0] \* YMean[9]) / (SumX2[0] - SumX[0] \* XMean[0]);*

*YInt[9] = YMean[9] - Slope[9] \* XMean[0];*

*}*

*label5.Text = "IMP(50kHz) = " + Convert.ToString(Fourier\_imp[0, r]); //7.2 , 6.9, 6.92, 7.0, 7.15, 7.3, 7.35, 7.4, 7.08*

*label6.Text = "IMP(100kHz) = " + Convert.ToString(Fourier\_imp[1, r]); //0.6, 0.58, 0.58, 0.61, 0.64, 0.71, 0.76, 0.73, 0.62*

*label7.Text = "IMP(200kHz) = " + Convert.ToString(Fourier\_imp[2, r]);*

*label4.Text = "IMP(300kHz) = " + Convert.ToString(Fourier\_imp[3, r]);*

*label3.Text = "IMP(400kHz) = " + Convert.ToString(Fourier\_imp[4, r]);*

*label8.Text = "IMP(5000kHz) = " + Convert.ToString(Fourier\_imp[5, r]);*

*label18.Text = "IMP(600kHz) = " + Convert.ToString(Fourier\_imp[6, r]);*

*label17.Text = "IMP(700kHz) = " + Convert.ToString(Fourier\_imp[7, r]);*

*label16.Text = "IMP(800kHz) = " + Convert.ToString(Fourier\_imp[8, r]);*

*label30.Text = "IMP(900kHz) = " + Convert.ToString(Fourier\_imp[9, r]);*

*label15.Text = "Taxa de variação(50kHz) = " + Convert.ToString(Slope[0]);*

*label14.Text = "Taxa de variação(100kHz) = " + Convert.ToString(Slope[1]);*

*label13.Text = "Taxa de variação(200kHz) = " + Convert.ToString(Slope[2]);*

*label12.Text = "Taxa de variação(300kHz) = " + Convert.ToString(Slope[3]);*

*label11.Text = "Taxa de variação(400kHz) = " + Convert.ToString(Slope[4]);*

*label10.Text = "Taxa de variação(500kHz) = " + Convert.ToString(Slope[5]);*

*label9.Text = "Taxa de variação(600kHz) = " + Convert.ToString(Slope[6]);*

*label20.Text = "Taxa de variação(700kHz) = " + Convert.ToString(Slope[7]);*

*label19.Text = "Taxa de variação(800kHz) = " + Convert.ToString(Slope[8]);*

*label39.Text = "Taxa de variação(900kHz) = " + Convert.ToString(Slope[9]);*

*Application.DoEvents();*

*}*

*using (System.IO.StreamWriter file = new System.IO.StreamWriter(@"C:\Users\Admin\Desktop\Resultado.txt", true))*

*{*

*for (int kk = 18; kk < rfinal; kk++)*

*{*

*file.WriteLine(Convert.ToString(Fourier\_imp[0, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[1, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[2, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[3, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[4, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[5, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[6, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[7, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[8, kk]) + " "*

*+ Convert.ToString(Fourier\_imp[9, kk]));*

*}*

*file.WriteLine("\n");*

*}*

*this.button1.Enabled = true;*

*this.button2.Enabled = false;*

*}*

*private void exitToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

*Environment.Exit(0);*

*}*

*Bitmap memoryImage;*

*private void CaptureScreen()*

*{*

*Graphics myGraphics = this.CreateGraphics();*

*Size s = this.Size;*

*memoryImage = new Bitmap(s.Width, s.Height, myGraphics);*

*Graphics memoryGraphics = Graphics.FromImage(memoryImage);*

*memoryGraphics.CopyFromScreen(this.Location.X, this.Location.Y, 0, 0, s);*

*}*

*private void printToolStripMenuItem\_Click(object sender, EventArgs e)*

*{*

*CaptureScreen();*

*printDocument1.Print();*

*}*

*private void printDocument2\_PrintPage(object sender, System.Drawing.Printing.PrintPageEventArgs e)*

*{*

*e.Graphics.DrawImage(memoryImage, 0, 0);*

*}*

*private void button2\_Click\_1(object sender, EventArgs e)*

*{*

*this.button1.Enabled = true;*

*this.button2.Enabled = false;*

*\_stopLoop = true;*

*}*

*private double[] dft(double[] data)*

*{*

*int n = data.Length;*

*int m = n;*

*double[] real = new double[n];*

*double[] imag = new double[n];*

*double[] result = new double[m];*

*double pi\_div = 2.0 \* Math.PI / n;*

*for (int w = 0; w < m; w++)*

*{*

*double a = w \* pi\_div;*

*for (int t = 0; t < n; t++)*

*{*

*real[w] += data[t] \* Math.Cos(a \* t);*

*imag[w] += data[t] \* Math.Sin(a \* t);*

*}*

*result[w] = Math.Sqrt(real[w] \* real[w] + imag[w] \* imag[w]) / n;*

*}*

*return result;*

*}*

*}*

*}*

1. comite de ética - TCLE

Termo de consentimento livre e esclarecido para medição em pacientes com diabetes e neuropatia no Hospital Universitário da UFSC

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**UNIVERSIDADE FEDERAL DE SANTA CATARINA HOSPITAL UNIVERSITÁRIO PROFESSOR**

**POLYDORO ERNANI DE SÃO THIAGO**

**TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO**

O(A) Sr(a). está sendo convidado(a) a fazer parte de um estudo de pesquisa, chamado

“**APLICAÇÃO E AVALIAÇÃO CLÍNICA DE METODOLOGIA PARA DIAGNÓSTICO PRECOCE DA**

**NEUROPATIA AUTONÔMICA EM INDIVÍDUOS COM DIABETES**”. Este termo de

consentimento tem o objetivo de fornecer informações sobre o estudo que está sendo

proposto.

Leia este termo de consentimento atentamente e, se tiver alguma dúvida, peça

explicações a alguém da equipe do estudo. Se concordar em participar do estudo e

autorizar o uso e divulgação das informações obtidas durante o estudo, assine a última

página deste termo de consentimento e faça uma rubrica nas outras páginas.

A Neuropatia Autonômica é uma complicação grave e comum do Diabetes. Ela pode trazer

várias consequências, principalmente para a saúde do coração. Se for descoberta no início,

quando o paciente ainda não sente nada, o tratamento adequado pode ajudar a prevenir

estas alterações. Atualmente, não existe um teste simples que seja usado de rotina no

acompanhamento dos pacientes diabéticos para saber se a neuropatia autonômica está

presente. Ou seja, ela só é descoberta muito tarde, quando já existem sintomas. Um teste

simples baseado na análise do eletrocardiograma pode ser utilizado para avaliar estas

alterações, sendo realizado facilmente em ambulatórios. Uma forma adicional, ainda

pouco utilizada, é a avaliação da capacidade de sudorese, uma vez que há correlação de

problemas de produção de suor com neuropatia autonômica.

**Objetivo do Estudo e Justificativa**

O objetivo deste estudo é avaliar a utilidade de um teste simples baseado na análise do

eletrocardiograma para a detecção precoce da neuropatia autonômica em pacientes

diabéticos.

Você poderá ser convidado a fazer uma medida adicional para análise de deficiências de

sudorese também para detecção de neuropatia autonômica.

**Procedimentos do Estudo**

Serão feitas algumas perguntas sobre a sua saúde, e será aplicado um questionário sobre

os seus sintomas. Serão verificados: peso, altura, medida da cintura, pressão arterial,

frequência cardíaca e um exame completo dos seus pés. Os exames laboratoriais (de

sangue e urina) serão os mesmos que você faz durante o acompanhamento de rotina, no

laboratório do HU.

Os dados de exame clínico e exames laboratoriais poderão ser obtidos diretamente

durante a consulta ou dos registros no seu prontuário médico.

Serão avaliadas a sua pressão arterial e a sua frequência cardíaca enquanto você executa

manobras simples, como respirar profundamente ou mudar de posição (deitado em pé).

Por fim o eletrocardiograma em repouso será registrado durante 5-8 minutos.

Caso você faça o teste adicional para detecção de deficiências de sudorese, será orientado

a lavar/limpar suas mãos com água e sabão ou álcool gel e ficar sentado com elas sobre

eletrodos, que estimularão as glândulas que produzem o suor, existentes em grande

quantidade nas palmas das mãos e pés, com uma diferença de potencial elétrico muito

pequena de 4.0 Volts (como se estivesse segurando uma pilha) durante 5 minutos, sendo

avisado ao termino do procedimento. Em seguida será pedido para que limpe seus pés e

os coloque, também, sobre dois eletrodos durante 5 minutos, sendo avisado ao termino

do teste.

Pela sua participação, você não receberá qualquer valor em dinheiro, da mesma forma

que não terá qualquer despesa com o que for necessário para o estudo.

**Possíveis Efeitos Indesejáveis**

Você poderá sentir um leve desconforto ao realizar os exames, por precisar ficar na

mesma posição por alguns minutos.

**Benefício Esperado**

O benefício do estudo é o maior conhecimento sobre a neuropatia autonômica diabética,

com possibilidade de se detectar antes que os sintomas apareçam e atuar precocemente

prevenindo futuras complicações.

**Confidencialidade e Privacidade dos seus Dados e Identificação**

A equipe do estudo obterá informações suas como nome, endereço, telefone de contato,

data de nascimento, dados sobre sua saúde e história médica, além das informações

verificadas durante o estudo. Ao apresentar os resultados da pesquisa em congressos ou

publicações, a equipe nunca citará seu nome ou dados pessoais.

**Posso me Recusar a Participar do Estudo?**

Sim. A sua participação neste estudo é voluntária. A qualquer momento você tem o direito

de se retirar da pesquisa. Se decidir por não mais participar, não haverá, sob hipótese

alguma, nenhum prejuízo do seu atendimento e tratamento médico aos quais tem direito.

**Como Poderei Tirar Dúvidas a Respeito do Estudo?**

Em caso de dúvida entre em contato com os pesquisadores que estão conduzindo o

estudo: Prof. Jefferson L B Marques, no Instituto de Engenharia Biomédica da UFSC,

telefone 3721-2592, ou Enfª Ana Paula Pacheco, Dr. Alexandre Hohl, Drª Cristina Oliveira,

Dr. Marcelo Ronsoni, Drª Marisa Coral e Drª Simone Lee, no ambulatório de

endocrinologia do HU (área B), telefone (48) 3721-9134. Se você tiver alguma dúvida sobre

a ética da pesquisa, pode entrar em contato com o Comitê de Ética em Pesquisa (CEP) –

Biblioteca Universitária Central – Setor de Periódicos (térreo), fone: (48) 3721-9206.

Você receberá uma cópia assinada deste termo de consentimento livre e esclarecido.

**Eu li e compreendi este termo de consentimento. Confirmo que fui devidamente**

**esclarecido sobre os objetivos e, as finalidades do estudo e os termos de minha**

**participação na pesquisa. Sou voluntário para participar deste estudo.**

**Nome do voluntário: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data\_\_\_/\_\_\_/\_\_\_\_\_**

**Nome do pesquisador responsável (ou delegado):\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Assinatura: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Data\_\_\_/\_\_\_/\_\_\_\_\_**

Rubrica Voluntário: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Rubrica Pesquisador: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_