



Universidade Federal da Bahia
Instituto de Matemática
Departamento de Ciência da Computação

Relatório Final do Projeto MEDIC/MEDISOM Software

Joselito Mota Júnior

Salvador, Agosto de 2016

Joselito Mota Júnior

Relatório Final do Projeto MEDIC/MEDISOM Software

Relatório Final de Iniciação Científica
apresentado para conclusão do projeto Medic/
Medisom com apoio da Fundação de Amparo
e Pesquisa do Estado da Bahia(FAPESB).

Universidade Federal da Bahia – UFBA

Instituto de Matemática

Departamento de Ciência da Computação

Orientador: Paul Dennis Etienne Regnier

Salvador

Agosto de 2016

Joselito Mota Júnior

Relatório Final do Projeto MEDIC/MEDISOM Software/ Joselito Mota Júnior.
– Salvador, Agosto de 2016-
Orientador: Paul Dennis Etienne Regnier

Relatório Final de Iniciação Científica – Universidade Federal da Bahia – UFBA
Instituto de Matemática
Departamento de Ciência da Computação, Agosto de 2016.

1. Software. 2. Medição. I. Paul Dennis Etienne Regnier. II. Universidade
Federal da Bahia. III. Departamento de Ciência da Computação. IV. Projeto
Medic/Medisom Software

Resumo

O objetivo geral do projeto é desenvolver uma plataforma mecatrônica livre de medição e visualização da radiação ultrassom[2] em 3D dentro de dutos. Utilizando sensores piezoelétricos, submersos dentro dos dutos, para a aquisição das ondas ultrassônicas emitidas por um cinturão de transdutores de ultrassom[2]. Microcontroladores são utilizados para as aquisições dos dados recebidos desses sensores, e também do controle dos motores de passo, que fazem o trabalho de movimentação dos sensores nas direções dos eixos X,Y e Z a partir das coordenadas enviadas do software final. Uma aplicação intermediária trata os dados recebidos do microcontrolador para que a aplicação final possa ler e gerar os mapas em 2D e 3D do duto.

Três segmentos serão responsáveis pelas atividades descritas anteriores. Serão as bases do desenvolvimento do projeto. São eles: o hardware, o firmware e o software.

Hardware: as atividades encarregadas para o hardware são de programação, montagem e integração do Microcontrolador com os motores de passo e sensores. Toda a movimentação e qualidade dos dados recebidos pelos sensores serão observadas e descritas. Também é de responsabilidade do hardware a confiabilidade e a mobilidade dos motores de passo que integra o sistema.

Firmware: será responsável pela conversão, adaptação, aperfeiçoamento e tratamento dos dados e da comunicação do hardware para o software. A conversão de dados recebidos pelos Microcontroladores que estão presentes no sistema físico e o seu tratamento de sinais será a desafiadora e fundamental tarefa de integração do sistema com o seu software final de tratamento e geração de mapas 2D e 3D.

Software: desenvolvimento de um software com a para armazenamento e visualização de mapas em 2D e 3D dos dados obtidos sobre o comportamento das ondas de ultrassom[2] no duto imerso de líquido. O software proporciona uma visualização rápida e amigável dos dados utilizando a poderosa linguagem Python[1] e suas bibliotecas Matplotlib[4], para a geração de mapas; e a biblioteca TKinter, para a criação da interface gráfica do usuário. Coordenadas de posições que serão enviadas a partir do software para o Microcontrolador para fazer a codificação e envio dos sinais de movimentações dos motores de passo. O sistema mecatrônico de mesa XY será responsável por posicionar o sensor de maneira correta para mapear as áreas selecionadas.

Os três segmentos serão oferecidos para a comunidade sob licença de software livre e código aberto GNU/GPL, permitindo assim as suas apropriações pelas instituições públicas

brasileiras.

Assim os três segmentos serão responsáveis pelo produto final que é a visualização com qualidade dos resultados dos experimentos. Cada segmento é de fundamental importância para o desenvolvimento do projeto. Por isso, existe uma forte motivação ao trabalho em grupo e colaboração de todas as pessoas envolvidas nos três diferentes segmentos base do projeto.

Palavras-chaves: ultrassom. plataforma. livre. medição. piezoelétricos hardware. firmware. software. python. matplotlib. graficos.

Lista de ilustrações

Figura 1 – Mapa tridimensional gerado pela plataforma através da biblioteca Matplotlib.	2
Figura 2 – Janela de carregamento de pontos.	5
Figura 3 – Menu principal para escolha	5
Figura 4 – Gráficos do disco 0 de 7 em 2D e 3D plotados no Medisom Software . .	6
Figura 5 – Gráfico do disco 0 salvo em 2D	6
Figura 6 – Gráfico do disco 0 salvo em 3D	7
Figura 7 – Exemplo de múltiplos gráficos em execução	7
Figura 8 – Todas as medições em um gráfico 3D	8
Figura 9 – Zoom em um gráfico 3D	9
Figura 10 – Salvamento de um gráfico 2D	10

Sumário

	Introdução	1
1	PREPARAÇÃO DA PESQUISA	3
1.1	Materiais e Métodos	3
1.2	Causas e Procedimentos para Superá-las	3
2	RESULTADOS E DISCUSSÃO	5
2.1	Resultados	5
2.2	Discussões	10
2.3	Atividades Realizadas no Período	10
3	CONCLUSÃO	13
4	REFERÊNCIAS	15

Introdução

O projeto consiste no desenvolvimento de uma plataforma mecatrônica livre para medição e visualização da radiação ultrassom[2] em 3D dentro de dutos com líquido (inicialmente dutos de petróleo a fins de limpeza dos mesmos), com integração entre hardware, firmware e software. Em busca de uma maior disponibilidade do produto final, todo o projeto visa a utilização de licença de software livre e de código aberto GNU/GPL.

Com o início do projeto, foi firmado o compromisso inicial de pesquisas e estudos aprofundados em sua respectiva área por parte de todos os integrantes (estão participando do projeto dois alunos de graduação em Ciência da Computação, um aluno de graduação em Engenharia da Computação e uma aluna de mestrado em Mecatrônica), o compartilhamento de informações é de grande necessidade para a integração consistente entre hardware, firmware e software, assim a comunicação é de suma importância, além do desenvolvimento em paralelo.

Dentre os sensores estudados, o sensor piezoelétrico será o mais usado para medição das ondas de ultrassom[2] (ondas que estão a uma frequência superior àquela que o ouvido humano pode perceber, que é em torno de 20.000 Hz. Dispositivos ultrassônicos operam acima de 20 kHz) geradas por um cinturão em volta ao duto que possui um conjunto de transdutores ultrassônicos que também utiliza o efeito piezoelétrico no seu funcionamento. O sensor piezoelétrico mede pressão, tensão, força ou aceleração, convertendo isso em um sinal elétrico. Pela lei de Ohm a corrente interna do sensor piezoelétrico pode ser calculada, pois a tensão é semelhante à resistência, assim sendo utilizado para monitoramento das ondas de ultrassom[2] emitidas no líquido. Já os transdutores convertem um sinal elétrico em ondas de ultrassom[2], ou seja, trabalha de maneira contrária ao sensor gerando as ondas.

Para as medições a plataforma mecatrônica está sendo desenvolvida sobre uma base de alumínio retangular e se desloca nos eixos X, Y e Z graças aos dois motores de passo (motor elétrico usado quando algo tem que ser posicionado de maneira muito precisa) que recebem orientações de uma placa controladora. Por sua vez, recebe valores de localização gerados pelo software em um computador pessoal que são transmitidas através de comunicação USB e firmware desenvolvido para este propósito. Realizada a medição, o resultado retorna por USB ao software que trata os dados e os fornece em uma interface clara a quem manipula o sistema. Uma das maneiras de visualização destes dados é por meio de mapas 3D que facilita uma melhor interpretação por conta do usuário.

Para visualizações dos resultados dos sinais com qualidade e precisão, um aperfeiçoamento dos conhecimentos sobre a linguagem de programação Python[1] e suas bibliotecas

Matplotlib[4] e TKinter está sendo de extrema necessidade, pois o programa final permitirá ao usuário interação precisa com a plataforma mecatrônica, além de visualização consistente dos dados para posterior análise. Para a integração com cada parte desenvolvida por todos os integrantes, já é necessário domínio dos assuntos envolvidos, assim reuniões periódicas para apresentação do que foi feito foram sendo realizadas e todos os integrantes desenvolveram suas respectivas áreas em paralelo. Foram disponibilizados pela instituição salas, laboratórios e alguns outros recursos, como computadores, microcontroladores e todas as ferramentas necessárias para desenvolvimento do projeto, além de consultas com outros professores que se disponibilizam a ajudar agregando muito mais conhecimento e informação.

A Figura 1 ilustra o...

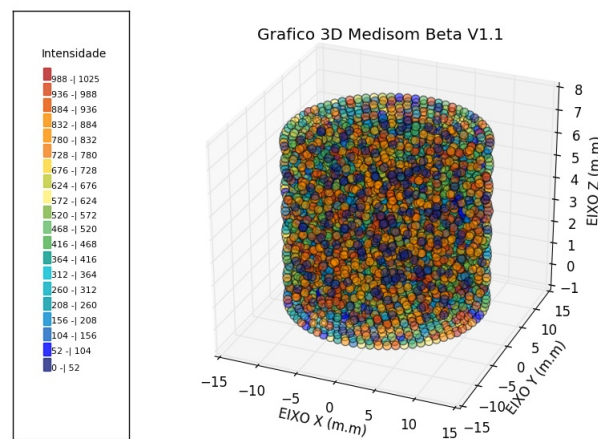


Figura 1 – Mapa tridimensional gerado pela plataforma através da biblioteca Matplotlib.

Iniciando com estudos mais aprofundados sobre as linguagens de programação C e Python[1] e suas bibliotecas e ferramentas existentes a fim de desenvolver um ambiente que fornece, através de uma interface limpa e clara, medições e visualizações de sinais com qualidade e precisão para quem a manipula (software). Para a integração com cada parte desenvolvida por todos os integrantes, é necessário domínio dos assuntos envolvidos, como cavitação e sensores que serão utilizados no andamento do projeto. Um protocolo de comunicação via USB (firmware), que foi desenvolvido em código aberto, será utilizado para permitir a aquisição de dados entre a plataforma mecatrônica criada (hardware) e a interface disponibilizada para o usuário (software). Para tais estudos, testes e aprofundamentos, estão sendo disponibilizados pela instituição salas, laboratórios e alguns outros recursos, como computadores, microcontroladores entre outros.

1 Preparação da Pesquisa

1.1 Materiais e Métodos

Os materiais utilizados para a produção deste trabalho foram os artigos científicos, textos, livros diversos sobre as linguagens de programação, manuais sobre as bibliotecas e sistemas operacionais, documentações de linguagens e bibliotecas e código fonte[3] de software com a maior parte disponível em repositórios online de software livres para consulta e reprodução.

Os múltiplos testes foram realizados em microcomputador portátil, sistemas embarcados, sensores, motores que a plataforma de medição possui.

O registro das informações coletadas foram feitos utilizando computadores, software de processamento e edição científica, repositórios online para publicação e exibição do código produzido, fotos e vídeos mostrando o seu funcionamento e comportamento.

1.2 Causas e Procedimentos para Superá-las

A busca incessante pela resposta dos questionamentos sobre a linguagem e a biblioteca e suas limitações em certos pontos do projeto fez com que a pesquisa fosse de muita importância. Por tanto, os fóruns de tecnologia, livros e tutoriais foram de fundamental importância, mesmo sendo escassos. A experimentação também foi importante no processo de obtenção dos resultados do software.

2 Resultados e Discussão

2.1 Resultados

Os resultados obtidos no software foram de grande relevância para o projeto. A aplicação foi desenvolvida com a interface do usuário bem amigável e simples.

Por ter uma proposta livre, este ambiente de geração de mapas 2D e 3D só funciona corretamente em ambientes baseados em Linux. Exige a instalação do ambiente Idle 2.7.9, Python[1] 2.7 e do pacote de gráficos Matplotlib[4].

A seguir o programa do Medic/ Medisom em funcionamento, bem como as suas janelas de interface e o resultado das gerações dos mapas.

A primeira janela de execução o software é o carregamento de pontos armazenados em arquivos em formato na extensão .txt gerados pela aquisição de dados feitas pelo firmware e pelo hardware.

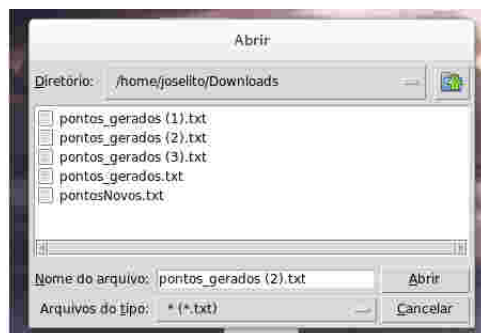


Figura 2 – Janela de carregamento de pontos.

Neste arquivo estão disponíveis o diâmetro do duto estudado, as coordenadas X, Y e Z e a intensidade medida da radiação ultrassom[2] pelo sensor.

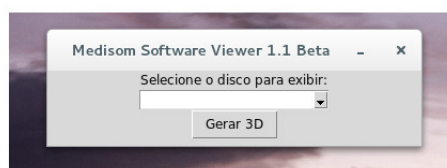


Figura 3 – Menu principal para escolha .

Na janela da Figura 3 é possível escolher entre os mapas 2D ou 3D disponíveis no arquivo de configuração. O programa separa cada eixo Z como um novo disco para visualização separada em 2D ou 3D(Figura 4).

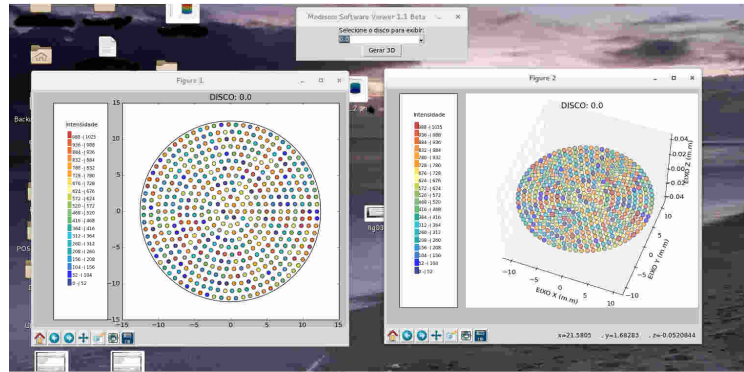


Figura 4 – Gráficos do disco 0 de 7 em 2D e 3D plotados no Medisom Software

Com a ferramenta também é possível o salvamento desses gráficos em diferentes formatos.

O gráfico que foi mostrado acima...

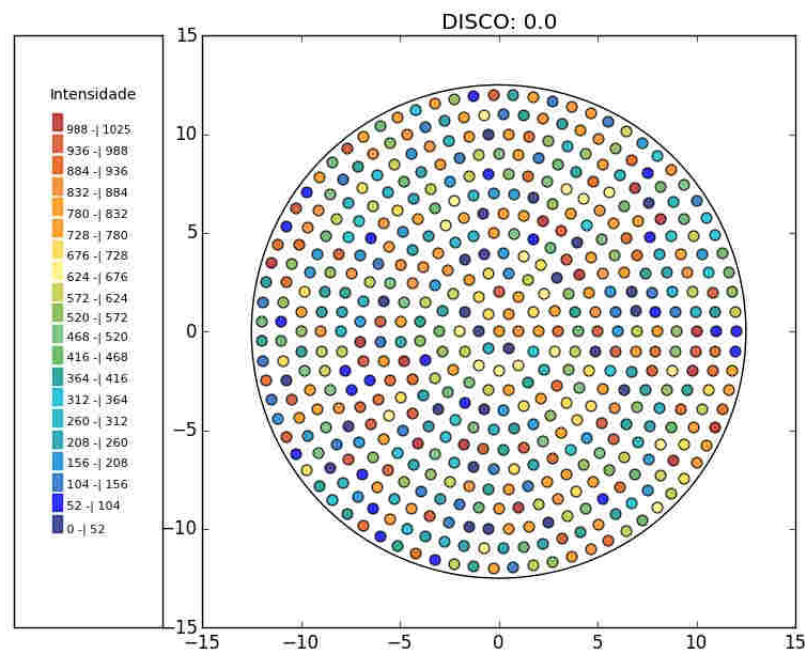


Figura 5 – Gráfico do disco 0 salvo em 2D

Projeção do mesmo gráfico acima em 3D...

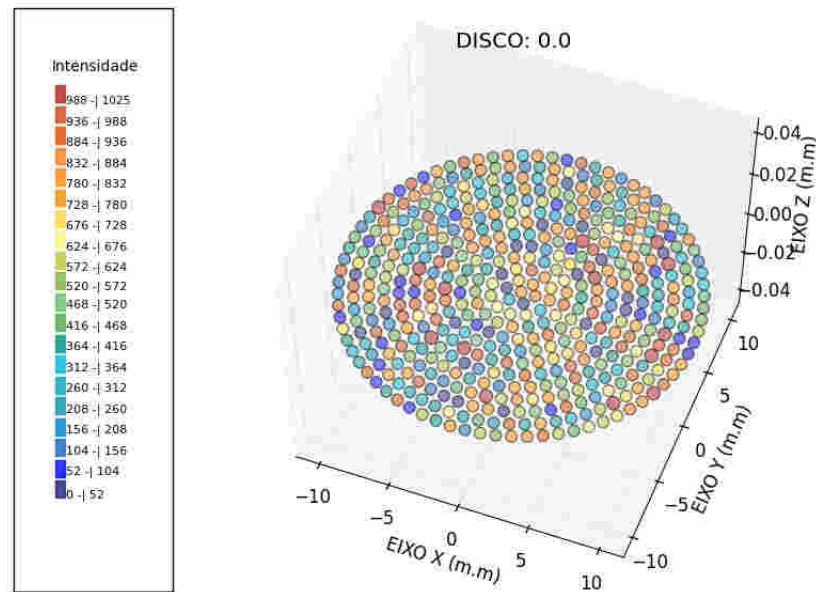


Figura 6 – Gráfico do disco 0 salvo em 3D

Para análise de vários discos ao mesmo tempo é possível a abertura de múltiplos gráficos. A implementação desta funcionalidade foi de um tanto trabalhosa, pois cada gráfico é um objeto diferente na biblioteca. Se um gráfico 2D está aberto e ele está nomeado na função, todos os gráficos vão ser plotados em cima do gráfico utilizado, não sendo viável a abertura de múltiplas janelas. Todos os futuros gráficos serão criados no nomeado por serem do mesmo tipo, ou seja, em 2D ou 3D. Houve uma pesquisa muito grande para que o software possa ter a funcionalidade de múltiplas janelas. A documentação da biblioteca está em outra língua e as fontes são escassas. Este recurso é de muita importância para que o usuário possa comparar fenômenos diferentes em diversos mapas gerados.

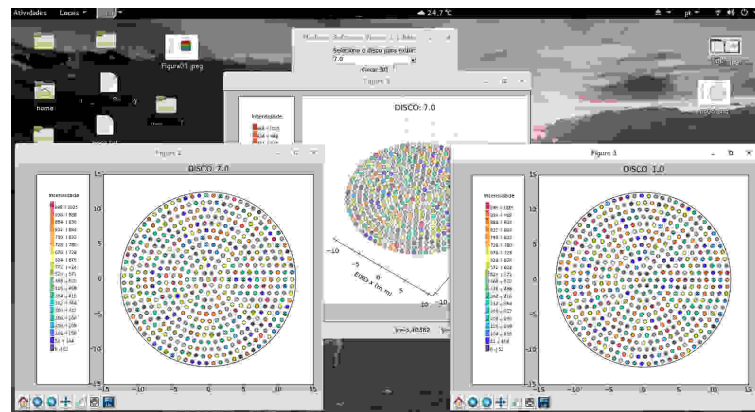


Figura 7 – Exemplo de múltiplos gráficos em execução

A visualização de todos os discos em 3D também está disponível. Assim o usuário

pode analisar de uma maneira mais geral os efeitos dentro do duto e tirar conclusões a partir dos dados coletados. Esta visão geral reúne todas as medições passadas pelo arquivo .txt. Todos os setores medidos, coordenadas e intensidades são mostrados de uma maneira geral no mapa total em 3D.

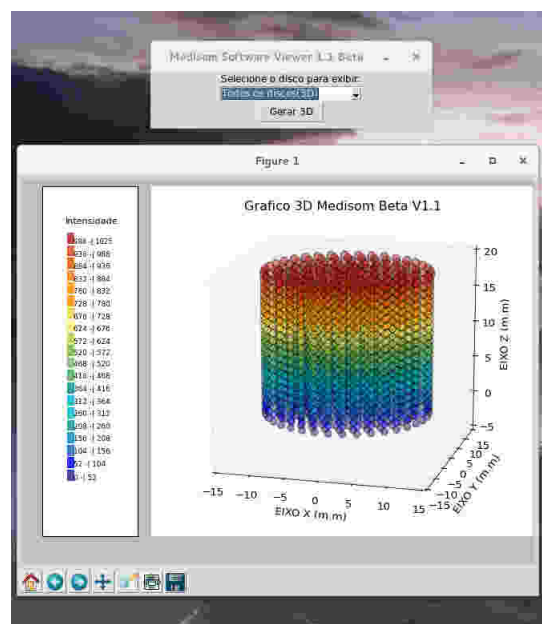


Figura 8 – Todas as medições em um gráfico 3D

É possível utilizar o zoom em figuras plotadas, como mostra a figura abaixo...

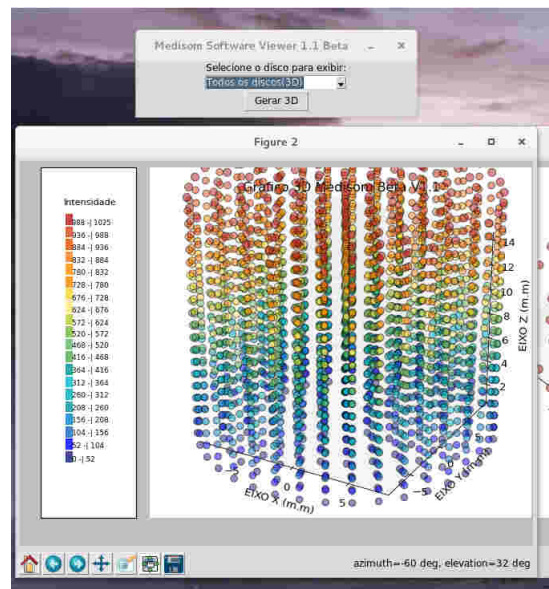


Figura 9 – Zoom em um gráfico 3D

O software permite o salvamento dos gráficos para análise ou anexo em trabalhos acadêmicos.

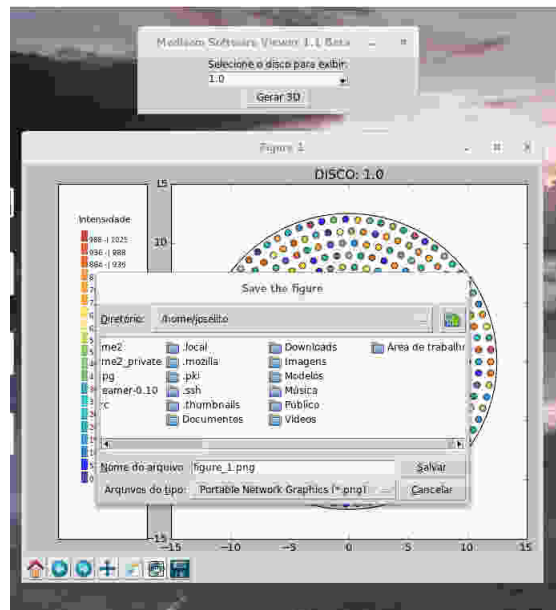


Figura 10 – Salvamento de um gráfico 2D

Observação.: Os gráficos e os pontos mostrados foram gerados aleatoriamente utilizando o programa de Firmware. O software ainda não foi testado com medições reais.

2.2 Discussões

Com a análise feita pelo hardware e os dados sendo tratados pelo firmware, o software recebe as informações em um arquivo e faz o carregamento dos pontos e da tradução das intensidades na escala de cores apresentada. Esperávamos uma certa demora na condensação e tratamento dos dados pelo software, já que ele trata muitos pontos de uma só vez. O efeito foi justamente o contrário, mostrando uma eficiência muito acima da esperada. Os pontos são tratados e os gráficos gerados em questão de segundos. Isto mostra que a biblioteca, a linguagem Python e a implementação utilizada foi na medida para que possa atingir a finalidade da plataforma de forma eficiente.

2.3 Atividades Realizadas no Período

1. Reuniões periódicas para definição e acompanhamento das etapas do projeto.
2. Reuniões em laboratório para construção e testes de protótipos;
3. Realização de levantamento bibliográfico sobre os seguintes temas:

- Sistemas Operacionais de tempo real (Xenomai) – Para possível aproveitamento futuro;

- Sensores (cristal Piezoelétrico);
 - Python (aprofundamento da linguagem de programação e a biblioteca Matplotlib para geração de mapas 2D e 3D dos sinais recebidos através do sensor de cristal Piezoelétrico);
 - C (aprofundamento da linguagem de programação para integrações com python e maior eficiência no processamento dos sinais, além da comunicação direta com microcontroladores utilizados no hardware);
 - Estudo da biblioteca Tkinter para criar interface de usuário amigável para a utilização do software de geração de mapas;
 - Estudo da biblioteca PyQt
 - Estudos das particularidades e possíveis deficiências da linguagem e da biblioteca;
 - Estudo sobre escalas de cores;
4. Início da construção da interface gráfica utilizando a biblioteca Tkinter da linguagem de programação Python;
 5. Testes na interface gráfica de usuário
 6. Construção de gráficos utilizando a biblioteca Matplotlib;
 7. Teste de gráfico 2D;
 8. Teste do gráfico 3D;
 9. Teste nas estruturas de dados de entrada dos pontos;
 10. Integração da interface gráfica de usuário, criada pela biblioteca Tkinter, com o módulo de geração de mapas;
 11. Criação da escala de cores por html;
 12. Teste exaustivo de erros e bugs no software;
 13. Possíveis otimizações no código fonte;
 14. Publicação em repositório Github;
 15. Organização para uma possível participação em congresso ou publicação no futuro;

3 Conclusão

A plataforma de visualização de mapas 2D e 3D está disponível para a plataforma Linux. Portanto, o software Medic/Medisom é um software livre para utilização da comunidade. A geração de mapas utilizando a biblioteca Matplotlib[4] do Python[1] se torna uma poderosa ferramenta para o estudo do fenômeno da radiação ultrassom[2] em dutos com líquidos. Ressaltando também a qualidade do material que a biblioteca possui, sendo uma excelente escolha para o problema proposto. Bem como o Python[1]. A escolha dela foi de fundamental importância para chegar aos resultados obtidos.

Sendo um software inicial, Medic/ Medisom Software está aberto para aprimoramentos e modificações no GitHub(<https://github.com/joselitojunior94/MEDISOM-SOFTWARE>) pela comunidade. O objetivo é atender ao projeto de iniciação científica e também para a comunidade de software livre.

A divisão dos módulos Hardware, Firmware e Software foi de fundamental importância para o andamento do projeto, pois cada módulo foi trabalhado independente com a colaboração das pessoas envolvidas no processo. Cada módulo também possui uma dependência que foi igualmente trabalhada de modo que o projeto fosse integrado com sucesso.

O Medic/ Medisom Hardware, Firmware e Software é uma plataforma inovadora criada por estudantes de graduação da Universidade Federal da Bahia para atender também a comunidade externa. Fruto de muito trabalho, pesquisa e empenho geral da equipe juntamente com o Professor Orientador Paul Dennis Etienne Regnier.

4 Referências

3D Graphing and Maps for Excel, R, Python, and MATLAB: Gender and Jobs, a 3D Gaussian, Alcohol, and Random

Advances in Ultrasound Technology for Environmental Remediation, Wu, T.Y.; Guo, N.; Teh, C.Y.; Hay, J.X.W

[1] 2013, VI, 120 p. 11 illus., 2 illus. In color., Softcover. EDUARDO BORGES, LUIZ – Python para Desenvolvedores 2a Edição, 2010 .

E.KLOEPPEL, JAMES - Temperature inside collapsing bubble four times that of sun

[2] Geração de ultra-som - <http://www.forp.usp.br/restauradora/us01.htm>Gera

GUI Programming Python - <https://wiki.python.org/moin/GuiProgramming>

Installing Xenomai 3.x - <https://xenomai.org/installing-xenomai-3-x/>

Kenneth S. Suslickfrom, THE CHEMISTRY OF ULTRASOUND, The Yearbook of Science and the Future 1994;

Encyclopaedia Britannica: Chicago, 1994; pp 138-155

[3]Matplotlib Examples - <http://matplotlib.org/examples/index.html>

Matplotlib License - <http://matplotlib.org/users/license.html>

Matplotlib mplot3d toolkit - <http://matplotlib.org/mpltoolkits/mplot3d/index.html>

[4] Matplotlib Docs - <http://matplotlib.org/contents.html>

Python Brasil, Comparação de GUI - <http://wiki.python.org.br/ComparacaoDeGUIs>

S. SUSLICK, KENNETH – The Chemistry of Ultrasound

Stack Overflow Questions - <http://stackoverflow.com/questions/17410400/pythonscattered-chart-legend>

Statistical Analysis Handbook – c 2015 Dr M J de Smith WAN, MINGXI; FENG, YI; TER HAAR, GAIL – Cavitation in Biomedicine, Springer Science+Business Media Dordrecht 2015.

WILDI, Theodore. Electrical Machines, drives and power systems. 2ed. Prentice Hall: 1991. 727p.

Walks-<http://blog.plot.ly/post/122970698612/3d-graphing-maps-for-excel-r-pythonmatlab>

Xenomai 3 Programmer's Reference Manual - <http://www.xenomai.org/documentation/xenomai>

3/pdf/xeno3prm.pdf