

**HEITOR DE MORAES SANTOS
PEDRO HENRIQUE DE A. AMORIM
RODRIGO MIDEA COELHO**

**Dispositivo de detecção de fissuras estruturais via impedância
elétrica para smartphones**

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Bacharel em Engenharia

São Paulo
2015

**HEITOR DE MORAES SANTOS
PEDRO HENRIQUE DE A. AMORIM
RODRIGO MIDEA COELHO**

**Dispositivo de detecção de fissuras estruturais via impedância
elétrica para smartphones**

Trabalho de Conclusão de Curso
apresentado à Escola Politécnica da
Universidade de São Paulo para obtenção
do título de Bacharel em Engenharia

Área de concentração:
Engenharia Elétrica com ênfase em
Sistemas Eletrônicos

Orientador:
Prof. Dr. Cássio Guimarães Lopes

São Paulo

2015

P-1
TF-2015
Su59 cl.



Escola Politécnica - EPEL



31500009793

Catálogo na publicação

m2015K

Santos, Heitor de Moraes

Dispositivo de detecção de fissuras estruturais via impedância elétrica para smartphones / H. M. Santos, P. H. A. Amorim, R. M. Coelho – São Paulo, 2015.

130 p.

Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Sistemas Eletrônicos.

1.Detecção de fissuras 2.Ensaio não destrutivo 3.Tomografia por impedância elétrica I.Universidade de São Paulo. Escola Politécnica, Departamento de Engenharia de Sistemas Eletrônicos II.I. III.Amorim, Pedro Henrique de Araújo IV.Coelho, Rodrigo Midea

[2155431]

AGRADECIMENTOS

Ao professor e orientador Cássio Guimarães Lopes, que tornou possível a execução e conclusão deste trabalho pela presença constante durante seu desenvolvimento e por todas as ideias que teve e nos inspirou a ter.

Aos nossos familiares e amigos, pelo carinho, apoio e compreensão constantes.

RESUMO

Neste trabalho, descreve-se um sistema eletrônico para detecção de fissuras em estruturas metálicas que tem por princípio de funcionamento a tomografia por impedância elétrica (*Electrical Impedance Tomography* ou EIT). Através de eletrodos dispostos ao redor da estrutura, aplica-se corrente num par de eletrodos adjacentes e medem-se as tensões de todos os outros pares, e esse processo repete-se até que a corrente tenha sido aplicada em todos os pares de eletrodos. Utiliza-se então o software EIDORS para construir o mapa de condutâncias da seção transversal da estrutura (cuja borda é definida pela presença dos eletrodos) com as tensões medidas, e as áreas com condutâncias baixas são identificadas como falhas na estrutura. O dispositivo descrito neste trabalho torna automático o processo de aplicação de corrente e medição de tensões, além de enviar os dados coletados no formato correto a uma estação de processamento, onde o mapa de condutividades será construído. Os testes de validação do dispositivo foram feitos com diversas placas quadradas de aço zincado e lados entre 20 e 25 cm, cada uma delas com fissuras de formatos e tamanhos diferentes (cuja menores dimensões estão em torno de 0,8 cm), e em todos os casos as fissuras com área maior que 0,5 cm² foram localizadas e identificadas. O tempo total entre medições, transmissão dos dados para a estação de processamento e construção do mapa de condutividades é menor que 11 segundos. Materiais metálicos foram o foco do trabalho, o sistema, porém, pode-se adaptar facilmente para ensaios com outros materiais. O sistema apresenta potencial para se tornar uma solução portátil e viável para situações de campo, utilizando-se um *smartphone* ou *tablet* como estação de processamento.

Palavras-chave: Detecção de fissuras. Ensaio não-destrutivo. Tomografia por impedância elétrica. EIDORS. Estruturas metálicas.

ABSTRACT

In this work, an electronic system which uses the operating principle of electrical impedance tomography (EIT) for crack detection in metal structures is described. Through electrodes positioned around the structure, continuous current is applied in a pair of adjacent electrodes and voltage is measured on all other pairs, and this process is repeated until the current has been applied to all pair of electrodes. Afterwards EIDORS software is used to build a conductance map of structure's cross section (which edge is defined by the presence of electrodes) with the measured voltages, and areas with low conductance are identified as structure failure. The device described in this work automatizes the current injection and voltage measurement process, besides sending the captured data in the right format to a processing station, where the conductance map is made. Validation tests were performed with different galvanized steel boards with sides measuring between 20 cm and 25 cm, each one of them with cracks in different formats and sizes (with smallest dimension of 0.8 cm), and in all cases cracks with area greater than 0.5 cm² were detected. The total time between measurements, data transmission to processing station e conductance map construction is less than 11 seconds. Metallic materials have been the focus of this work, but the system can be easily adapted for testing with other materials. The system has potential to become a portable and viable solution for field situations using a smartphone or tablet as processing station.

Keywords: Crack detection. Non-destructive testing. Electrical Impedance Tomography. EIDORS. Metal structures.

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application programming interface</i>
EIT	<i>Electrical impedance tomography</i>
NDI	<i>Non destructive inspection</i>
NDT	<i>Non destructive testing</i>

SUMÁRIO

1. INTRODUÇÃO	13
1.1 Pesquisa de Levantamento da Situação	14
1.1.1 Visão Geral.....	14
1.1.2 Estado da arte	16
1.1.2.1 Tomografia por impedância elétrica	16
1.1.2.2 Testes não destrutivos	16
2. REFERENCIAL TEÓRICO.....	19
2.1. Tomografia e EIT: Contextualização	19
2.2. Princípios físicos da EIT	20
2.3. Procedimentos de medição da EIT	21
2.4 Algoritmo de reconstrução e EIDORS	21
3. METODOLOGIA	24
3.1. Concepção	24
3.1.1 Especificação de requisitos de engenharia e de marketing.....	25
3.1.1.1 Requisitos de Engenharia	25
3.1.1.2 Requisitos de Marketing	26
3.1.1.3 Árvore de objetivos.....	27
3.1.1.4 Padrões Técnicos.....	28
3.1.2 Geração de conceito	29
3.1.2.1 Tabela de Conceito	29
3.1.2.2 Árvores de Conceitos	30
3.1.2.3 Avaliação Inicial e Análise de Forças e Fraquezas	30
3.1.3 Decomposição funcional	34
3.1.3.1 Análise de Acoplamento e Coesão	37
3.2 Design	38
3.2.1 Prova de Conceito	38
3.2.1.1 Prova de Conceito: Itens Críticos	38
3.2.1.2 Prova de Conceito: Método	38
3.2.1.3 Prova de Conceito: Resultados	39
3.2.1.4 Prova de Conceito: Conclusões	41
3.2.2 Simulações, testes e o projeto de Hardware	42
3.2.2.1 Fonte de corrente	42
3.2.2.1.1 Fonte de corrente DC	42
3.2.2.1.2 Fonte de corrente AC	44
3.2.2.2 Multiplexação de corrente	48

3.2.2.2.1 Conjunto de transistores NMOS para multiplexação de corrente	48
3.2.2.2.2 Bloco completo de multiplexação de corrente	50
3.2.2.3 Hardware de aquisição de tensões	52
3.2.2.4 Condicionamento de Sinal	55
3.2.2.5 Microcontrolador - Firmware e ADC	57
3.2.2.6 Protocolo de comunicação	60
3.2.2.7 Testes de implementação em corrente alternada	63
3.2.3 Esquemático e decomposição finais	66
3.2.4 Sistema de integração do detector de fissuras com smartphone	66
3.3. Implementação	70
3.3.1 Placa de circuito impresso – concepção, fabricação e montagem	70
3.3.2 O conjunto de eletrodos	75
4. RESULTADOS E ANÁLISE DOS RESULTADOS	77
4.1 Testes e prototipagem	77
4.1.1 Teste dos multiplexadores de corrente	77
4.1.2 Teste dos amplificadores de tensão	78
4.1.3 Teste do sistema integrado – Protótipo	78
4.1.3.1 Resultados	79
4.1.3.2 Especificações técnicas	85
4.1.3.3 Análises	86
4.1.3.4 Consideração sobre a implementação em smartphones	90
5. CONCLUSÕES	91
6. BIBLIOGRAFIA	93
7. ANEXOS	96
Anexo A - Esquemático final do hardware	96
Anexo B – Código do firmware para Tiva-C Launchpad	101
Anexo C – Scripts MATLAB do protocolo de comunicação e chamda a execução do algoritmo de reconstrução de imagem	120
Anexo D – Script MATLAB de automatização do processo	123
Anexo E – Script MATLAB do algoritmo de reconstrução de imagem por impedância elétrica – EIDORS	124
Anexo F – Gerenciamento de projeto	125
Anexo G – Análise de riscos	133
Anexo H – Custo do projeto	135

1. INTRODUÇÃO

Os custos de reparação de infraestruturas críticas, tais como instalações de armazenamento de resíduos nucleares, tanques de combustível, peças de aviões e navios, entre outros, podem comprometer consideravelmente o orçamento de empresas, sendo, por isso, alvos de estudos de minimização de custos. Além disso, os riscos de segurança que falhas em estruturas como essas trazem são enormes e insustentáveis. Esses, entre outros fatores, motivam crescentes estudos e desenvolvimentos na área de Inspeção Não-Destrutiva (em inglês *Non-Destructive Inspection*, NDI).

Especialistas dividem hoje a NDI em 3 principais tipos: Inspeção Visual, Testes Não-Destrutivos (NDT – *Non-Destructive Testing*) e Monitoramento de Saúde Estrutural (SHM – *Structural Health Monitoring*) [1]. Os NDT surgiram como alternativa para a inspeção visual através de ferramentas com detalhamento especial, para fissuras invisíveis a olho nu e para locais de difícil acesso na estrutura analisada. E, por mais que se desenvolvam aparelhos que visem aumentar a área de análise das estruturas, ferramentas portáteis de alta precisão são requeridas, dados os riscos que defeitos nessas estruturas oferecem.

Criar instrumentos que não mais se responsabilizam por todo o *pipeline* do processo - grosseiramente, aquisição e processamento - mas sim alternativas que fazem uso do recurso de processamento de outras máquinas como *desktops* locais, servidores ou ainda, *smartphones*, pode ser um caminho interessante no sentido de reduzir os custos de fabricação do instrumento e ainda aumentar a capacidade de processamento, podendo rodar algoritmos mais sofisticados.

Novamente sob a ótica dos dispositivos móveis, aliar essa instrumentação a smartphones tende não só a reduzir os preços das ferramentas como também a aumentar sua acessibilidade[2]. Desta forma, é possível construir uma plataforma modular de sensoriamento para smartphones (constituída de sensores, módulos de aquisição e envio de dados e o smartphone em si, conforme ilustrado na Figura 1) em

que se pode mudar de aplicação através da mudança dos sensores e do processamento dos dados no smartphone.

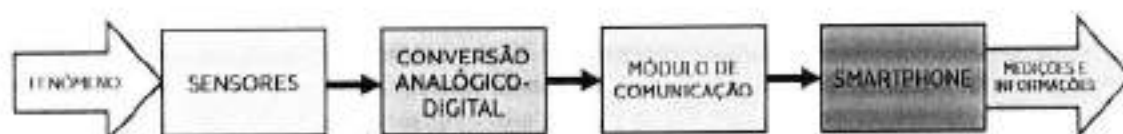


Figura 1 - Diagrama de blocos da plataforma modular de sensoriamento para smartphones. Fonte: Autor, 2015.

Partindo de uma plataforma modular de sensoriamento para smartphones, desenvolver dispositivos de testes não-destrutivos para estruturas que visem reduzir os custos desta instrumentação e aumentar sua acessibilidade a estruturas críticas, como as encontradas nas indústrias petroquímicas, aeronáuticas, na construção civil, entre outras, que nem sempre permitem fácil acesso e sugerem soluções portáteis. O dispositivo baseia-se na técnica de tomografia por impedância elétrica (*Electrical Impedance Tomography – EIT*) e gera imagens funcionais das estruturas através de um aplicativo dedicado, disponibilizando informações sobre o tamanho e localização das fissuras. Uma vez disponíveis no smartphone, o usuário tem a facilidade de armazenamento e envio destas imagens para anexo em relatórios das manutenções periódicas da estrutura.

1.1 Pesquisa de Levantamento da Situação

1.1.1 Visão Geral

Testes não-destrutivos (NDT) envolvem um amplo grupo de técnicas de análise usadas na ciência e na indústria para avaliar as propriedades de um material, componente ou sistema sem causar-lhes danos. Uma vez que não interferem permanentemente nos objetos a serem avaliados, a técnica de NDT é valiosa e permite economizar tempo e dinheiro, solucionar problemas e pesquisar novas tecnologias.

As técnicas de NDT são aplicadas nas mais diversas áreas atualmente, sendo bastante referenciadas nas indústrias químicas e petroquímicas, aeronáuticas e civil, onde desempenham um papel essencial de manutenção das infraestruturas críticas. As técnicas são as mais diversas, sendo que algumas delas são capazes de não só detectar fissuras, como também mapeá-las em imagens 2D (imageamento 3D em NDT é uma tecnologia ainda imatura [3]). Entre estas capazes de gerar imagens, a Tomografia por Impedância Elétrica (EIT) é uma técnica pertinente ao escopo deste projeto, já que a eletrônica de interface que ela exige é simples, comparada com outras técnicas de imageamento, contribuindo para o desenvolvimento de dispositivos de baixo custo.

A técnica de imagens médicas por EIT chamou inicialmente uma atenção maior do que para aplicações em NDI [3], pois se apresentava como uma solução muito mais barata que a tomografia de campo rígido (*hard-field tomography* – raios-X, raios- γ) e que não contém fontes de emissão radioativa. Assim, as primeiras patentes sobre tomografia via bioimpedância surgiram há cerca de 30 anos.

As diversas técnicas para instrumentação NDI, entre as quais Eddy Current, Ultrassom, Termografia Infravermelha, etc., fizeram com que diversas ferramentas fossem desenvolvidas e aplicadas industrialmente em diferentes áreas. A possibilidade, porém, de desenvolver instrumentos de baixo custo faz com que as técnicas de tomografia elétrica (EIT, ECT – *electrical capacitance tomography* – e MIT – *magnetic induction tomography*) sejam preferidas [2].

Tomografias elétricas são classificadas como tomografias de campo suave (*soft-field tomography*) e sua natureza é tal que a transmissão do campo não segue um padrão de linha reta e a distribuição do sinal no corpo sob teste depende do tipo de fonte de excitação. Essa natureza é muito mais complexa que a de campo rígidos, o que acarreta em grandes desafios para a computação dos algoritmos de reconstrução de imagens [2,3].

1.1.2 Estado da arte

1.1.2.1 Tomografia por impedância elétrica

A tomografia por impedância elétrica tem hoje seu uso bastante difundido em aplicações médicas e de detecção de fissuras em concreto.

Em aplicações médicas, monitoradores da atividade pulmonar e de corações têm sido usadas em UTIs de hospitais uma vez que, por não emitirem radiações prejudiciais à saúde humana, podem se manter funcionando ao corpo de paciente por até semanas [4].

Além disso, EIT tem ganhado ainda mais destaque no cenário biomédico atual para o monitoramento de outras atividades do corpo, com a ascensão do wearables no mercado [5].

Em testes não destrutivos, encontram-se muito mais nas literaturas atuais EIT associados a estruturas de concreto. Uma provável justificativa para isso não seria simplesmente uma questão de maior demanda de NDT para esse tipo de estrutura, mas sim, o fato de EIT apresentar maiores desafios técnicos em estruturas metálicas uma vez que a resistência dos metais é baixa, da ordem de magnitude da resistência dos condutores e contatos elétricos, o que gera maiores dificuldades para os sistemas de aquisição de dados.

1.1.2.2 Testes não destrutivos

O estado da arte de testes não destrutivos engloba diferentes técnicas baseadas em variados fenômenos físicos como ótica, tomografias de campo rígido, ultrassom, magnetismo, etc. As aplicações das técnicas variam de acordo com material, tipo de falha e resolução temporal e espacial pretendidas, ciclo de vida do material avaliado – se está em produção, manutenção, etc. -, entre outros. Para ilustrar melhor a situação atual de testes não destrutivos em estruturas metálicas, tomou-se como

exemplo aeronaves, uma vez que, hoje, pelo menos ainda 70% do peso global de aeronaves vem de materiais metálicos:

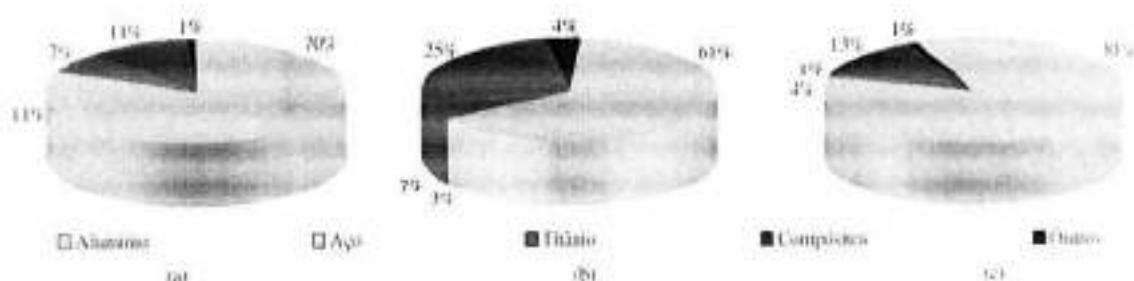


Figura 2 - Percentagem global de materiais utilizados na fabricação das aeronaves (Embraer). a) distribuição % global de peso do BOEING 777; b) distribuição % global de peso do AIRBUS A380; c) distribuição % global de peso do EMBRAER 170;

As tecnologias predominantes em NDT para os diferentes ciclos de vida das estruturas das aeronaves são:

- Radiografia de nêutrons
- Tomografia por raios-X
- Interferometria holográfica
- Termografia
- Laser ultrassom
- Emissão acústica
- Eddy current
- Etc.

Detecção de fissuras:

Fissuras em estruturas podem ser divididas em microfissuras, detectáveis por inspeção visual, ou microfissuras, as que só podem ser detectadas com microscópios ou testes não destrutivos. Outra distinção a ser feita é a de fissuras discretas, as quais são contadas e localizadas individualmente, e fissuras finas distribuídas, para as quais os cálculos de área são mais relevantes [6].

Os NDT, alvo deste projeto visam a detecção de fissuras tanto discretas, não detectáveis em inspeções visuais, quanto distribuídas no interior de estruturas metálicas.

2. REFERENCIAL TEÓRICO

2.1. Tomografia e EIT: Contextualização

A tomografia, técnica de geração de imagens de seções transversais de um volume que utiliza algum tipo de onda penetrante, vem sido proposta e desenvolvida em termos matemáticos desde o século 19 [7]. Grandes marcos em sua evolução foram, contudo, a introdução da transformada de Radon (1917, pelo matemático austriaco Johann Radon), que descreve como uma função poderia ser reconstruída a partir de um conjunto de projeções, e a construção do primeiro tomógrafo de radiação ionizante (1972, por Godfrey Hounsfield na Inglaterra) [8].

O processo de geração de imagens tomográficas começa com a colocação de uma fonte de energia em uma determinada posição no exterior do objeto, e diversas medidas são tomadas por sensores distribuídos na fronteira do objeto. Desse modo, identifica-se a projeção da energia ao sair do objeto, que depende da constituição interna do mesmo. Em seguida, a fonte é deslocada ao longo do contorno da seção transversal para dar origem a uma nova projeção, e esse processo é repetido até que se tenha aplicado energia em volta de toda a superfície do objeto. A partir do processamento dos dados das projeções, geram-se imagens das seções transversais que representam a distribuição de materiais no interior do objeto. A geração de imagens de tomografia, portanto, é um processo complexo devido às atividades que envolve: a energização de uma região de interesse, sistemas eletrônicos com múltiplos sensores, além da aquisição e processamento dos dados.

Por causa de sua natureza não-destrutiva e não-invasiva, técnicas de imagens tomográficas são muito úteis em contextos industriais, mostrando-se adequadas à avaliação de estruturas e peças, por exemplo. Graças à evolução na fabricação de computadores e sensores, aplicações de tomografia em processos industriais tornaram-se possíveis e viáveis. Na indústria, contudo, as exigências são diferentes das da medicina, sendo em geral necessário diminuir os custos e o tempo de geração de imagens e usar equipamentos menores cuja operação seja simples e segura. Assim, uma abordagem que tem sido bem-sucedida na tomografia para aplicações industriais é o uso de correntes e tensões elétricas no lugar da radiação ionizante de

alta frequência, numa técnica conhecida como tomografia por impedância elétrica (em inglês, Electrical Impedance Tomography ou EIT).

2.2. Princípios físicos da EIT

Os princípios físicos que explicam o funcionamento da tomografia por impedância elétrica têm origem nas leis de Maxwell [7]. Seja o objeto Ω um subconjunto fechado do espaço tridimensional, limitado pela curva fechada $\partial\Omega$. O objeto tem condutividade σ , que é função da posição. O potencial elétrico é ϕ e o vetor campo elétrico é dado pela relação: $\mathbf{E} = -\nabla\phi$, sendo $\nabla\phi$ a operação gradiente sobre o campo escalar de potenciais ϕ ; o vetor densidade de corrente é dado pela lei de Ohm: $\mathbf{J} = -\sigma\nabla\phi$. Não havendo fontes de corrente no interior do objeto, tem-se, pela lei de Kirchhoff: $\nabla(\sigma\nabla\phi) = 0$ (ou seja, a soma das correntes que entram no objeto Ω é igual à soma das correntes que saem dele).

A densidade de corrente na fronteira $\partial\Omega$ é: $\mathbf{j} = -\mathbf{J} \cdot \mathbf{n} = -\sigma\nabla\phi \cdot \mathbf{n}$, onde $\mathbf{n}$ é o vetor unitário normal a $\partial\Omega$, apontando para fora. Dada a distribuição de condutividades σ , a especificação do potencial elétrico $\phi|_{\partial\Omega}$ na fronteira é suficiente para determinar univocamente a solução para o potencial em todo o volume Ω (condição de contorno de Dirichlet). De modo similar, a especificação da densidade de corrente $\mathbf{j}$ em $\partial\Omega$ é suficiente para determinar uma solução para ϕ a menos de uma constante aditiva (condição de contorno de Neumann), o que equivale a escolher um ponto de terra.

A corrente aplicada na fronteira do objeto entra e sai do mesmo (não há acúmulo de carga). Portanto, o problema de reconstrução em EIT requer que se conheçam todos os pares $\phi|_{\partial\Omega}, \mathbf{j}$, ou seja, que se encontre o operador $\Lambda_\sigma : \phi|_{\partial\Omega} \mapsto \mathbf{j}$ correspondente à transcondutância elétrica do objeto Ω . O resultado da reconstrução da EIT é, enfim, o mapa de distribuição de condutâncias de Ω [7].

2.3. Procedimentos de medição da EIT

Para se medir a impedância elétrica, utiliza-se um conjunto de eletrodos dispostos em uma seção transversal ou em uma lateral de superfícies chapadas (assume-se que o corpo sob inspeção é plano e fino, ou que é fino e a zona de inspeção, pequena: a fuselagem de um avião, por exemplo, é aproximadamente cilíndrica, mas uma região pequena pode ser considerada plana). Após o posicionamento dos eletrodos, correntes são injetadas através de pares de eletrodos enquanto os outros captam as tensões na superfície da estrutura. O par de eletrodos em que se aplica a corrente é trocado e medem-se as tensões nos outros eletrodos, até que o procedimento se repita com todas as combinações diferentes de eletrodos de estímulos [9].

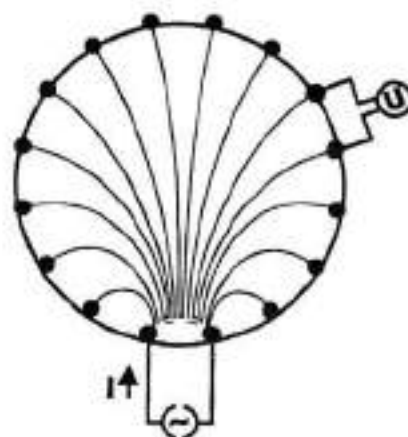


Figura 3 - Método de aplicação de correntes e medição de tensões em tomografia por impedância elétrica: a corrente (I) é aplicada sequencialmente nos pares de eletrodos adjacentes e as diferenças de potencial (U) são medidas nos pares de eletrodos restantes. A imagem é reconstruída ao longo das linhas equipotenciais (mostradas na figura) [7].

2.4 Algoritmo de reconstrução e EIDORS

O algoritmo executa a reconstrução das imagens de EIT em duas etapas: a resolução de um problema direto e, depois, de um problema inverso. No problema direto, a tensão é estimada para cada densidade de corrente no objeto, assumindo que o objeto tem condutância uniforme σ . Já o problema inverso consiste em calcular a condutância distribuída real $\sigma(x,y)$ a partir de todas as medições possíveis na fronteira $\phi|_{\partial\Omega}$. Pode-se dizer que o problema direto é do tipo "parâmetros do modelo $\rightarrow$ dados", enquanto o problema inverso é do tipo "dados $\rightarrow$ parâmetros do modelo". O problema direto é bem condicionado e se resolve pelo método dos elementos finitos, mas o

problema inverso é muito mal condicionado e não-linear; para resolver problemas desse tipo, é necessário conhecer a natureza da solução e utilizar essa informação para que se encontrem resultados razoáveis. Essa técnica é chamada de regularização.

A sequência de tarefas realizada pelo algoritmo de reconstrução de imagens de EIT é a seguinte [10]:

1. O objeto sob teste é discretizado num número finito de elementos com condutividade uniforme σ_0 ;
2. O problema direto é resolvido e a matriz das tensões de fronteira calculadas $[V_c]$ é computada para uma matriz conhecida de correntes aplicadas $[C]$ e uma estimativa inicial da matriz de condutividades $[\sigma_0]$;
3. A matriz de tensões medidas na fronteira $[V_m]$ é comparada com a matriz $[V_c]$ para calcular a matriz de diferenças $[\Delta V] = [V_m] - [V_c]$;
4. Avalia-se se as diferenças $[\Delta V]$ são menores que uma tolerância ϵ : em caso positivo, o algoritmo termina; caso contrário, o algoritmo vai para o próximo passo;
5. Calcula-se a matriz $[\Delta \sigma]$ pelo método de Gauss-Newton e utilizando-se a técnica de regularização;
6. Calcula-se a nova matriz de condutividades $[\sigma]_{k+1} = [\sigma]_k + [\Delta \sigma]_k$, onde k denota o número da iteração;
7. Volta-se para o passo 2 e o programa segue até que o critério $\Delta V < \epsilon$ seja satisfeito.

Neste projeto, essa rotina é implementada pelo software aberto EIDORS (Electrical Impedance and Diffuse Optical Tomography Reconstruction Software), cujo propósito é prover algoritmos de modelagem direta e inversa para reconstrução de imagens em problemas de tomografia por impedância elétrica, compartilhando dados e promovendo a colaboração entre grupos que trabalhem nesse tema. O software tem o formato de um toolkit (conjunto de funções) para Matlab (também GNU Octave, mas com limitações), e facilita a realização de testes e aplicações em EIT, tanto na área médica quanto no contexto industrial [11].

O software foi lançado em 1999 e reconstruía somente imagens bidimensionais, com algoritmo baseado em trabalhos escritos por Marko Vauhkonen e seu orientador Jari Kaipio, da Universidade de Kuopio [12]. Já os algoritmos de reconstrução tridimensional se baseiam no trabalho feito por Nick Polydorides e William Lionheart na Universidade de Manchester [13]. Devido ao crescimento do projeto e ao aumento de sua procura, Andy Adler e William Lionheart ampliaram as funcionalidades e alcance do EIDORS.

Além de ter sido consagrado por uma comunidade diversa e muito produtiva, o uso do EIDORS para reconstrução de imagens de tomografia por impedância elétrica evita a dificuldade da implementação de algoritmos para a resolução de problemas inversos. Algoritmos desse tipo estão muito sujeitos a falhas numéricas sutis, introduzidas inadvertidamente pelo desenvolvedor e difíceis de identificar e corrigir. Por ser amplamente usado, o EIDORS conta com uma documentação riquíssima, além de diversos exemplos prontos e dados compartilhados para testes e comparações, o que o torna adequado para os objetivos deste trabalho.

3. METODOLOGIA

A seção de metodologia é a mais densa desta documentação uma vez que relata o projeto desde a sua concepção a partir da problemática elucidada, detalhando como o produto se insere em um contexto técnico (requisitos de engenharia) e em um contexto de mercado (requisitos de marketing) e esclarecendo todas as etapas do design e implementação do mesmo.

3.1. Concepção

Tomografia por impedância elétrica é uma técnica que envolve grandes desafios nas áreas de projeto hardware e processamento de sinais. Isso por que os sinais tratados podem ser da ordem de magnitude de unidade e dezenas de microvolts e, a partir destes, surgem questões algorítmicas de resolução de um problema mal condicionado. Desta forma, o projeto se insere completamente no contexto do projeto do departamento de sistemas eletrônicos. Desafios em outras áreas como engenharia da computação estão também no escopo do projeto, uma vez proposta a criação de um app mobile comunicante com uma estação de controle e processamento do algoritmo (isto é, um servidor).

Assim, enumerar requisitos de engenharia desde a concepção do produto e após pesquisas em boas referências e estado da arte, criou um ponto de partida e a possibilidade de estimar com mais precisão pontos críticos do projeto, ganhando em gerenciamento e alocação de recursos humanos e de tempo para cada fase deste.

Já os requisitos de marketing contextualizam melhor o produto dentre as ferramentas de testes não-destrutivos.

3.1.1 Especificação de requisitos de engenharia e de marketing

Os requisitos e compromissos deste projeto foram determinados tendo-se em vista as funcionalidades que os clientes esperam do produto, as limitações trazidas pelas circunstâncias em que o produto será utilizado e as tecnologias que se têm à mão para construir esse sistema.

A seguir, estão listados os requisitos de engenharia e de marketing. A Tabela 1 mostra as relações entre eles.

3.1.1.1 Requisitos de Engenharia

- O dispositivo deve ser capaz de detectar fissuras de dimensões mínimas de 0,5 cm [14];
- O dispositivo deve detectar fissuras em estruturas de metal (notadamente aço e alumínio);
- Os eletrodos devem aderir fácil e ergonomicamente à estrutura;
- O ciclo de medições de tensão não deve levar mais de 30s [2];
- A fonte de corrente deve ser capaz de fornecer correntes de até 1000mA, a frequências de 0Hz (DC) até 100kHz [11];
- O ADC deve ter resolução de 10 bits;
- A estação de medição (eletrodos + ADC + módulo de comunicação) deve contar com bateria recarregável própria;
- A estação de medição deve ter dimensões máximas de 10 cm X 10 cm (mais os eletrodos, ligados por fios);
- O processo de medições e reconstrução de imagens deve ser iniciado por um comando enviado do smartphone;
- O aplicativo deve ter as opções: *Iniciar teste*, *Interromper teste*, *Mostrar imagem*, *Mostrar localização das fissuras* e *Sair*.

3.1.1.2 Requisitos de Marketing

- O dispositivo deve conseguir avaliar trechos críticos de estruturas grandes (paredes, vigas, elementos de pontes, fuselagens), dificilmente acessados com equipamentos maiores e não-portáteis;
- O smartphone deve mostrar imagens das estruturas com as fissuras em destaque [9, 11];
- A montagem do sistema deve ser simples;
- O sistema deve ser portátil, tanto por ser de fácil transporte quanto por ter plataforma para smartphones;
- O aplicativo deve ser fácil de usar;
- A bateria da estação de medição deve durar vários testes;
- O dispositivo deve ter preço atrativo em relação a outros produtos similares disponíveis.

		REQUISITOS DE ENGENHARIA									
		O dispositivo detecta fissuras com precisão de 0,5 cm (+)	O dispositivo avalia estruturas metálicas e de concreto (+)	Os estrados devem aderir à estrutura facilmente (+)	O ciclo de medições de tensão dura no máximo 30s (-)	A fonte de corrente fornece correntes de até 20mA @100kHz (+)	O ADC deve ter resolução de 10 bits (+)	A estação de medição deve ter bateria própria	A estação de medição deve medir no máximo 10 cm x 10 cm (-)	O processo se inicia após comando do smartphone	O aplicativo deve ter as funções Iniciar, Interromper, Mostrar Imagem E Fissuras e Sair
REQUISITOS DE MARKETING	Capacidade de avaliar estruturas grandes e de acesso difícil (+)	↑↑	↑↑	↑↑				↑	↑		
	Qualidade e precisão das imagens reconstruídas (+)	↑↑	↑↑				↑↑				
	Simplicidade de montagem (+)			↑↑					↑		
	Portabilidade do dispositivo (+)			↑↑				↑↑	↑↑	↑↑	↑
	Facilidade de uso do aplicativo (+)				↑↑					↑	↑↑
	Durabilidade da carga da bateria (+)				↓	↓		↑↑			
	Preço (-)			↓		↓	↓	↓	↑		

Tabela 1 - Matriz de compromissos entre os requisitos de engenharia e de marketing (legenda: (+) = aumentando melhora; (-) = diminuindo melhora; ↑ = influencia positivamente; ↑↑ = influencia muito positivamente; ↓ = influencia negativamente).

3.1.1.3 Árvore de objetivos

A seguir, na Figura 4, tem-se a árvore com os objetivos principais do sistema descrito neste trabalho. Há quatro características em especial que se desejam para o dispositivo de detecção de fissuras: funcionalidade, confiabilidade nos resultados, portabilidade e autonomia de bateria.

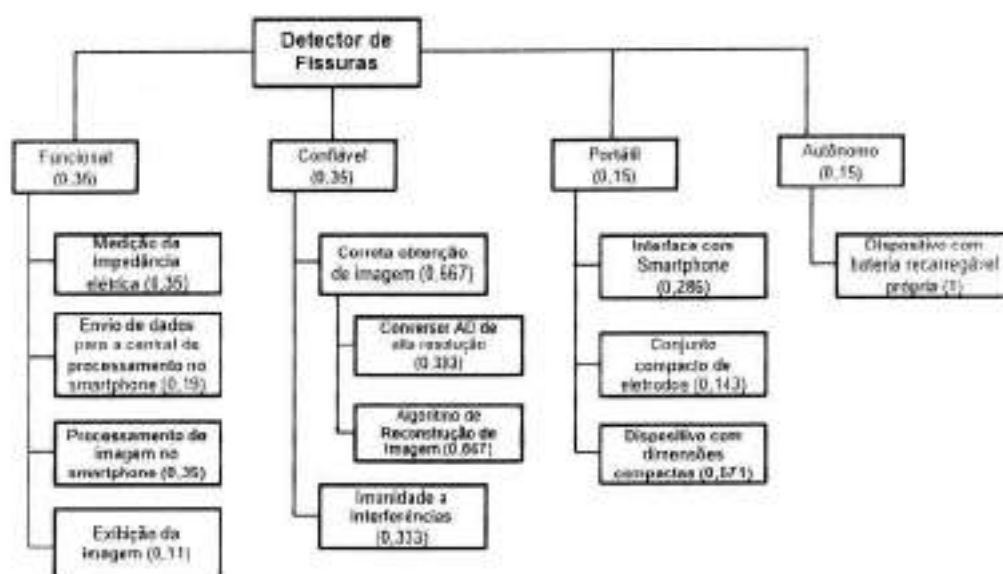


Figura 4 - Árvore de objetivos do detector de fissuras. Fonte: Autor, 2015.

3.1.1.4 Padrões Técnicos

Primeiramente, escolheu-se um padrão modular para a concepção do projeto – isto é, o mesmo é constituído de módulos desenvolvidos independentemente e acoplados ao final. A adoção de tal padrão confere versatilidade ao projeto, pois tem-se uma plataforma de sensoriamento para smartphone cuja aplicação pode transformar-se completamente ao se substituir um ou mais módulos.

Então, foram pesquisadas alternativas de um padrão técnico de comunicação para o módulo de comunicação do sistema, responsável pela transmissão de dados entre a estação de medição e o smartphone. As alternativas de padrões que o grupo levantou são as seguintes:

- *USB (Universal Serial Bus)*: esta é vista como uma primeira alternativa, por já estar presente tanto no smartphone quanto nas placas de desenvolvimento para microcontrolador (componente da estação de medição). Apresenta boa taxa de transferência de dados (a versão 3.0 tem taxa de transferência de até 5GBps [15]), mas exige ligação física (cabo) entre os dispositivos, o que pode limitar o posicionamento do dispositivo numa análise em campo;

- *WiFi*: apesar da comodidade de se escolher o padrão USB para a versão inicial do projeto, o WiFi é a alternativa que se deseja ter na versão final, pois dispensa ligação física entre os dispositivos (permitindo maior liberdade de movimentos e posicionamento dos eletrodos durante os testes), além de já está implementada nos smartphones e em módulos feitos para microcontroladores. O padrão WiFi IEEE 802.11g tem taxa de transferência máxima de 54Mbps e alcance máximo de 140 metros em áreas externas [16], sendo adequado para situações de teste em campo;
- *Bluetooth*: esta opção é considerada por permitir comunicação sem fio entre os dispositivos (este padrão já está implementado nos smartphones e em módulos prontos para microcontroladores). Contudo, tem alcance e taxa de transmissão menores que os do padrão WiFi (no máximo 100 metros e 24Mbps, respectivamente [17]).

3.1.2 Geração de conceito

3.1.2.1 Tabela de Conceito

Multiplexador Bidirecional	Condicionador de Sinal	Conversão Analógica Digital	ADC Dedicado	ADC Integrado	Interface Conversor / Smartphone	DAC
Cls de Chave Analógicas	Transformador	ADC Dedicado	FPGA	DSP	SPI - USB	Corrente
Relés	Amp Op	ADC Integrado	DSP	Microcontrolador	UART - USB	Tensão
MUX CMOS			Microcontrolador		SPI - Wifi	
			Buffer de saída		UART - Wifi	

Tabela 2 - Listagem de conceitos.

3.1.2.2 Árvores de Conceitos

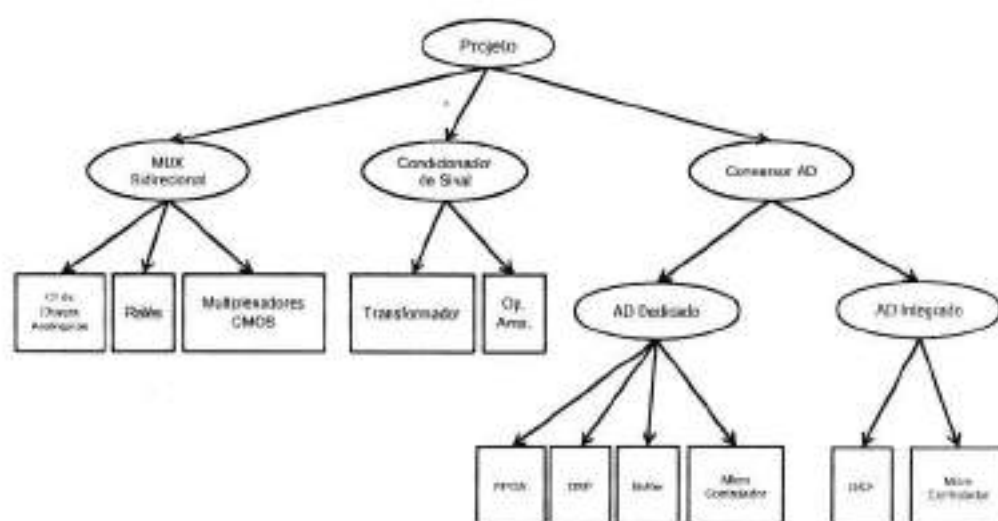


Figura 5 – Árvore de conceitos (opções) para o MUX bidirecional, o condicionador de sinal e o conversor AD. Fonte: Autor, 2015.

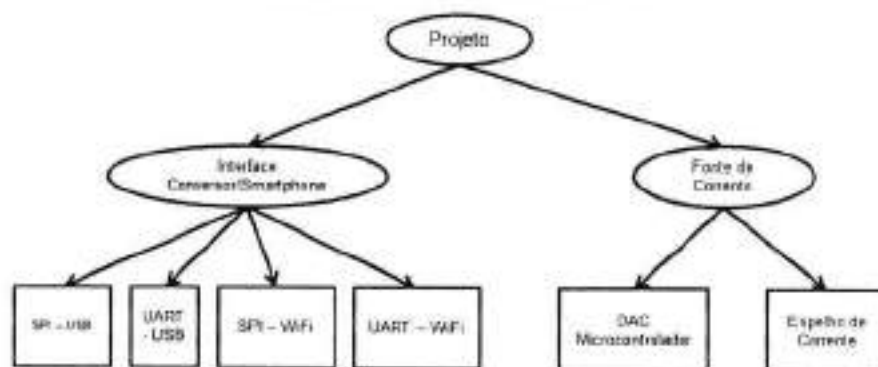


Figura 6 - Árvore de conceitos (opções) para a interface conversor/smartphone e a fonte de corrente. Fonte: Autor, 2015.

3.1.2.3 Avaliação Inicial e Análise de Forças e Fraquezas

As tabelas e árvores de conceitos reúnem diferentes opções tecnológicas às partes mais relevantes do projeto, ou seja, aquelas nas quais mudanças tecnológicas acarretam maiores efeitos de desempenho, custo, design, etc. no dispositivo. A partir destas, realiza-se uma avaliação inicial onde os conceitos que se mostrem de cara menos interessantes ao projeto possam ser descartados.

A lista abaixo reúne tecnologias descartadas nessa etapa inicial e a justificativa para tal:

- *MUX bidirecional → Relés*

MUX em chips integrados se mostram mais atrativos a um dispositivo portátil, uma vez que integram, por ex., 4 canais em um integrado de 14 pinos (DIP).

- *MUX bidirecional → Multiplexadores CMOS*

No caso de se utilizarem sinais digitais (que só apresentam 2 estados, alto e baixo), pode-se optar por implementar os multiplexadores bidirecionais utilizando multiplexadores digitais CMOS. Contudo, esse tipo de dispositivo pode distorcer sinais analógicos por ter um limite de tensão de saída, atuando como "clipper" para um sinal de entrada que exceda esse limite. Como haverá sinais analógicos de tensão na entrada do módulo MUX, a opção a se fazer são os circuitos integrados de chaves analógicas [18].

- *Condicionamento de sinal → Transformadores*

Comparando-se tamanho, facilidade de implementação em circuito impresso e custo unitário, um circuito integrado de amplificador operacional mostra-se muito mais adequado do que um transformador.

- *Fonte de Corrente → Espelho de corrente*

Implementar a fonte de corrente usando um circuito de espelho de corrente (isto é, projetar e construir o circuito usando componentes eletrônicos) é mais trabalhoso e traz maior probabilidade de imperfeições do que usar o próprio microcontrolador (seu módulo DAC) como fonte de corrente. Isto porque, no microcontrolador, bastam alguns comandos em software para se ter a corrente desejada em um de seus pinos.

- *Conversor AD → AD Dedicado*

Um AD dedicado tem como característica ter maior resolução e maior taxa de aquisição. Entretanto, quanto maior essa taxa de aquisição maior é a necessidade de

interfacear com um buffer de saída, como uma memória, além de muitas vezes ser necessário o desenvolvimento de hardware em FPGA para que não haja gargalos. Como a frequência máxima de análise deste projeto não deve passar dos 100kHz, acreditou-se não compensar usar um AD dedicado pois muitos microcontroladores possuem taxa de aquisição da ordem de 20 vezes superior a essa frequência, atendendo facilmente os requisitos do projeto e evitando o uso desnecessário de hardware.

- *Conversor AD → AD Integrado → DSP*

Há DSPs no mercado que possuem conversores analógico-digitais integrados no próprio chip. A vantagem de se usar um DSP é a de poder realizar um pré-processamento digital no sinal que será enviado ao smartphone, diminuindo a carga de processamento neste último. Entretanto, tipicamente DSPs não possuem memória, necessitando do interfaceamento com memórias externas. Outro ponto negativo é o de que programar o firmware de um DSP é mais complexo do que um microcontrolador, pois é necessário ter atenção com as questões limitantes de ponto flutuante, ponto fixo, interface de memória, etc.

• *Interface Conversor / Smartphone*

	Forças	Fraquezas
SPI - USB	- Adequado para dispositivos com fluxo contínuo de dados (como ADC) ↑↑↑ - Configuração física muito simples que UART ↑ - Máx. Taxa de transferência: USB 2.0 limita em 12 Mbps ↑↑↑	Necessita de ligação física ↓↓
UART - USB	Comumente integrado a microcontroladores ↑	- Necessita de ligação física ↓↓ - Configuração física complexa ↓ - Máx. Taxa de transferência: UART limita em até 2Mbps ↓↓↓
SPI - WiFi	- Adequado para dispositivos com fluxo contínuo de dados (como ADC); ↑↑↑ - Não necessita de ligação física ↑↑ - Configuração física muito simples que UART ↑ - Máx. Taxa de transferência: SPI limita em até 20 Mbps ↑↑↑	Configuração mais complexa que o SPI - USB ↓
UART - WiFi	Não necessita de ligação física ↑↑	- Configuração física complexa - Máx. Taxa de transferência: UART limita em até 2Mbps ↓↓↓

Tabela 3 - Análise de forças e fraquezas das opções para a interface entre conversor e smartphone.

Esta análise indica que a interface conversor / smartphone seria mais adequada ao nosso projeto se implementada por uma bridge SPI - WiFi.

3.1.3 Decomposição funcional

A Figura 7 mostra o diagrama de blocos correspondente ao sistema de detecção de fissuras de estruturas por meio da impedância elétrica [9]. Em seguida, tem-se a decomposição funcional de cada módulo da arquitetura do sistema (decomposição em nível 1).

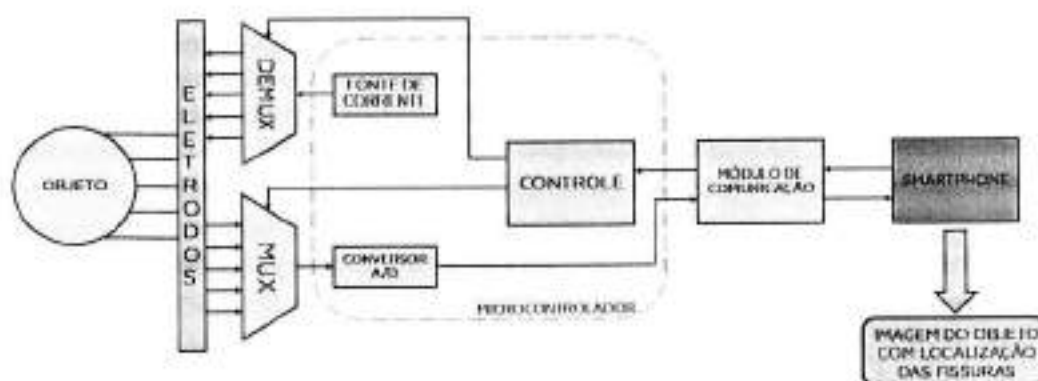


Figura 7 - Diagrama de blocos do sistema de detecção de fissuras por meio da impedância elétrica.

Fonte: Autor, 2015.

Módulo 1	Eletrodos
Entradas	Tensões medidas na superfície do objeto
Saídas	Corrente aplicada na superfície do objeto
Funcionalidade	Contatos elétricos que servem tanto para aplicar correntes à superfície do objeto como para medir as tensões de pontos da superfície desse objeto.

Módulo 2	Demultiplexador (DEMUX)
Entradas	Corrente (vinda da fonte de corrente)
Saídas	Corrente (através do par de eletrodos selecionado)
Funcionalidade	Selecionar o par de eletrodos através dos quais se aplicará corrente na superfície do objeto.

Módulo 3	Multiplexador (MUX)
Entradas	Tensões de todos os eletrodos
Saídas	Tensão medida entre o par de eletrodos selecionado
Funcionalidade	Selecionar o par de eletrodos em que se medirá a tensão.

Módulo 4	Fonte de corrente
Entradas	Comandos do microcontrolador (omitidos da Figura 7)
Saídas	Corrente
Funcionalidade	Fornece a corrente que se aplica à superfície do objeto para realizar o teste.

Módulo 5	Conversor analógico-digital (A/D)
Entradas	Sinal analógico de tensão de um par de eletrodos
Saídas	Sinal digital correspondente à tensão medida
Funcionalidade	Converte a tensão medida no par de eletrodos selecionados num sinal digital, a ser processado pelo smartphone.

Módulo 6	Controle
Entradas	Comandos do smartphone
Saídas	Sinal de seleção do MUX, sinal de seleção do DEMUX
Funcionalidade	Recebe comandos do smartphone para iniciar, pausar ou terminar as medições. Também envia os sinais de seleção para o DEMUX (seleciona os eletrodos em que se aplica a corrente) e para o MUX (seleciona os eletrodos em que se mede a tensão).

Módulo 7	Módulo de comunicação
Entradas	Dados digitais oriundos dos sensores (sem codificação) e comandos oriundos do smartphone (codificados)
Saídas	Dados digitais oriundos dos sensores (codificados) e comandos oriundos do smartphone (decodificados)
Funcionalidade	Serve como via de dados para realizar a comunicação entre a estação de medição e o smartphone (codifica, decodifica, transmite e recebe dados de acordo com seu padrão técnico).

Módulo 8	Smartphone
Entradas	Dados medidos pelo sensor, digitalizados e codificados
Saídas	Imagens reconstruídas do objeto com identificação das fissuras, comandos para a estação de medição
Funcionalidade	Processa os dados medidos, realizando a reconstrução das imagens do objeto com identificação das fissuras. Permite ao usuário controlar o teste (iniciá-lo, pausá-lo ou finalizá-lo).

3.1.3.1 Análise de Acoplamento e Coesão

Analisando-se o sistema ilustrado na Figura 7, nota-se que não há muitas conexões entre os módulos – de modo geral, cada módulo liga-se apenas ao módulo imediatamente anterior e ao imediatamente posterior. O único módulo que foge desse padrão é o de Controle, que se liga a três módulos (ainda assim, apresenta baixo número de conexões). Com isso, pode-se afirmar que o sistema tem acoplamento baixo e, portanto, um erro num determinado módulo é facilmente identificado por afetar poucos módulos diretamente.

Devido à concepção modular do sistema, tem-se que cada bloco representado na Figura 7 tem funções bem específicas – isto é, pode-se testar cada módulo separadamente e o controle necessário é localizado (somente para a seleção de sinais no multiplexador e no demultiplexador) e relativamente simples. Portanto, o sistema é altamente coeso.

3.2 Design

3.2.1 Prova de Conceito

3.2.1.1 Prova de Conceito: Itens Críticos

Conforme explicado na Seção 1.1, o dispositivo para detecção de fissuras estruturais proposto neste trabalho baseia-se numa plataforma modular de sensoriamento para smartphones (Figura 1).

Para se obter novas possibilidades de sensoriamento via smartphone a partir dessa plataforma, deve-se trocar o tipo de sensor (portanto, alterando o bloco "Sensores", primeiro da esquerda para a direita na Figura 1) e também alterar o algoritmo de processamento no celular (alterando o bloco "Smartphone", último da direita para a esquerda na Figura 1). Por conta disso, para a prova de conceito inicial, decidiu-se testar o princípio da tomografia por impedância elétrica realizando as etapas de sensoriamento, processamento dos sinais medidos e reconstrução das imagens. A seguir, estão descritos o método utilizado para a prova de conceito e seus resultados.

3.2.1.2 Prova de Conceito: Método

As medições foram realizadas na configuração de eletrodos adjacentes: a cada rodada, a corrente de excitação era aplicada à estrutura através de um par de eletrodos adjacentes e, então, mediam-se as tensões resultantes em todos os outros pares de eletrodos adjacentes (exceto no par em que se aplicava a corrente e nos seus dois pares vizinhos; a tensão nesses 3 pares era considerada nula) [9].

Cada rodada de medidas constitui a coluna de uma matriz de dados fornecida ao software EIDORS que, por sua vez, executa o algoritmo de reconstrução das imagens tomográficas das placas [11].

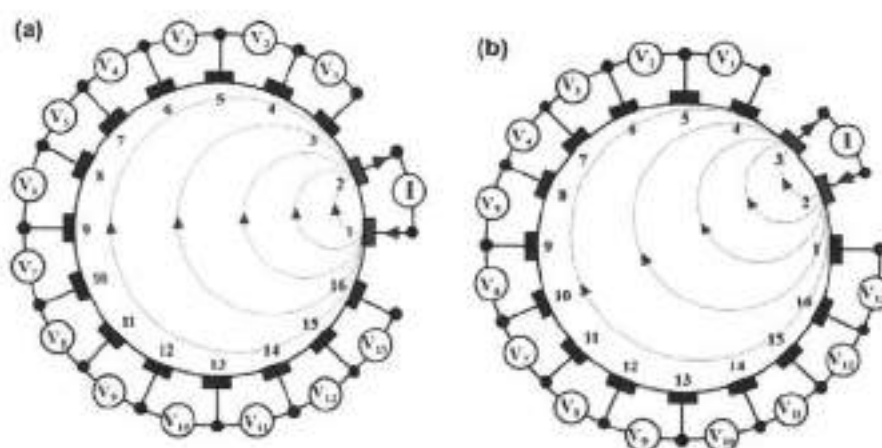


Figura 8 - Configuração de aplicação de corrente e medição de tensões nos pares adjacentes de eletrodos; em (a), a corrente é aplicada entre os eletrodos 1 e 2 e são medidas as tensões nos outros pares de eletrodos adjacentes (exceto os pares 1 e 2, 2 e 3 e 16 e 1); já (b) representa a rodada seguinte de medições, onde a corrente é aplicada entre os eletrodos 2 e 3 e são medidas as tensões nos outros pares de eletrodos adjacentes (exceto os pares 1 e 2, 2 e 3 e 3 e 4) [9].

3.2.1.3 Prova de Conceito: Resultados

A seguir, imagens das placas metálicas submetidas aos testes com suas respectivas imagens tomográficas reconstruídas com o software EIDORS. Em todos os testes, a corrente de excitação foi de 3 A; nos dois primeiros testes, utilizaram-se 8 eletrodos e, no terceiro, 16 eletrodos. Vale salientar a diferença de nitidez entre as duas imagens da placa de 4 furos retangulares, uma feita com 8 eletrodos (Figura 10, onde não se podem discernir os 4 furos, apesar de já apresentar algumas similaridades com a geometria original da placa) e a outra feita com 16 eletrodos (Figura 11, onde se distinguem bem as posições dos 4 furos no centro da placa).

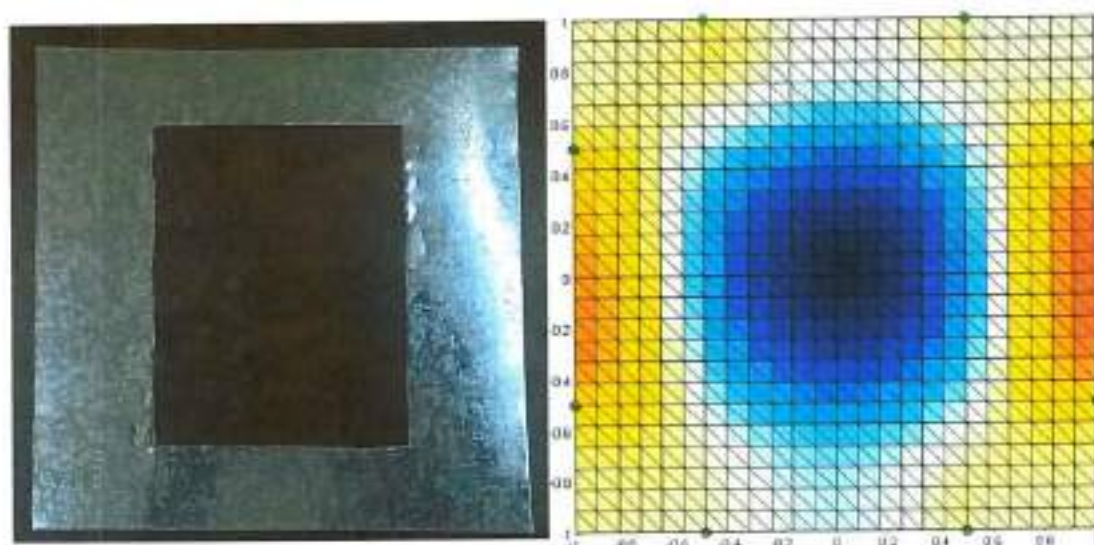


Figura 9 - À esquerda, placa metálica retangular de 25 cm x 25 cm com furo retangular no centro; à direita, imagem reconstruída da placa usando o programa EIDORS, com 8 eletrodos, configuração de eletrodos adjacentes e corrente de excitação de 3 A. Cores azuladas indicam condutividades relativamente mais baixas, e a localização dos eletrodos é representada pelos círculos verdes. Fonte: Autor, 2015.

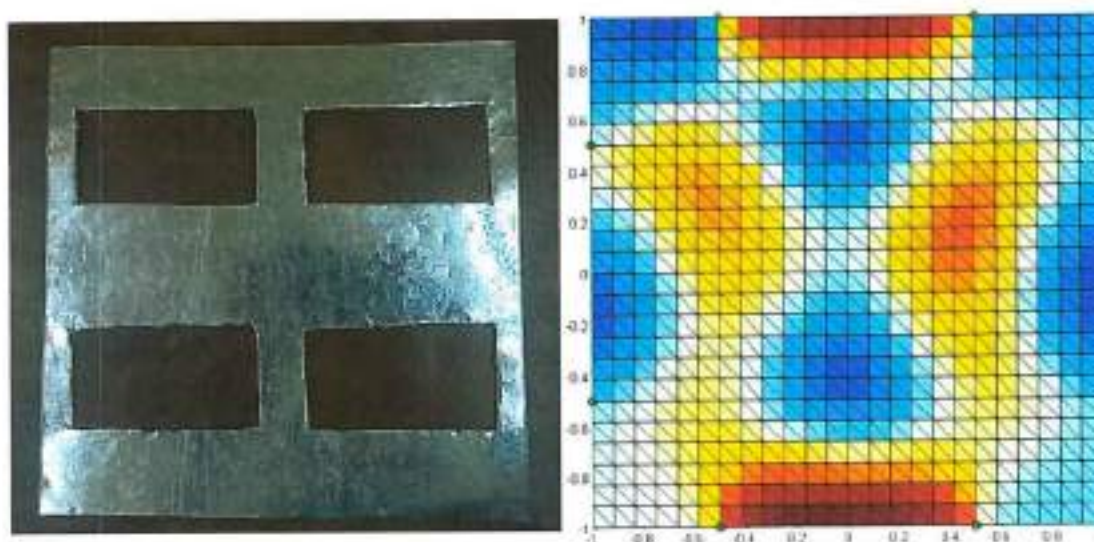


Figura 10 - À esquerda, placa metálica retangular de 25 cm x 25 cm com 4 furos retangulares; à direita, imagem reconstruída da placa usando o programa EIDORS, com 8 eletrodos, configuração de eletrodos adjacentes e corrente de excitação de 3 A. Cores azuladas indicam condutividades relativamente mais baixas, e a localização dos eletrodos é representada pelos círculos verdes. Fonte: Autor, 2015.

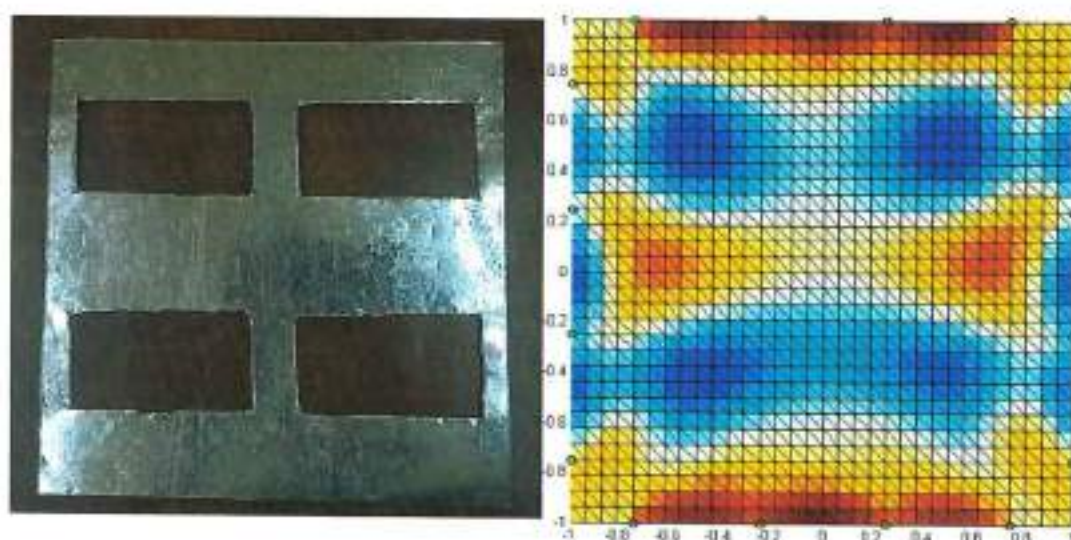


Figura 11 - À esquerda, placa metálica retangular de 25 cm x 25 cm com 4 furos retangulares; à direita, imagem reconstruída da placa usando o programa EIDORS, com 16 eletrodos, configuração de eletrodos adjacentes e corrente de excitação de 3 A. Cores azuladas indicam condutividades relativamente mais baixas, e a localização dos eletrodos é representada pelos círculos verdes. Fonte: Autor, 2015.

3.2.1.4 Prova de Conceito: Conclusões

Os resultados obtidos nos testes para prova de conceito foram bastante relevantes para o andamento do projeto. Deles uma série de informações puderam ser extraídas, entre as quais: uma estimativa do ganho em qualidade da imagem com o aumento do número de eletrodos de 8 para 16; o comportamento do algoritmo ante formas geométricas como retângulos; relações entre correntes injetadas e tensões medidas; influência nos testes de um bom e uniforme acoplamento entre eletrodos e estrutura-alvo; possíveis pontos do algoritmo a serem aperfeiçoados; além de direcionarem o projeto a outros testes onde novas e mais precisas conclusões possam ser tomadas.

Esses testes foram realizados em multímetros de bancada com fontes de tensão, com controle manual da corrente. O "acoplamento" dos aparelhos com as placas foi feito através de cabos banana-banana, pressionados manualmente na placa. Foram marcados nas placas metálicas pontos equidistantes nos contornos, representando a localização dos eletrodos. Os próximos testes, pretende-se que sejam automatizados, através de uma eletrônica de multiplexação dos eletrodos, o que traria eficiência a

eles e, assim, possibilitaria uma grande variedade de testes e maior domínio da metodologia.

Dessa forma, as expectativas são bastante positivas na obtenção de imagens mais precisas pois vê-se diversas possibilidades de melhorias, em hardware e software, em relação a esses testes iniciais de prova de conceito.

3.2.2 Simulações, testes e o projeto de Hardware

A prova de conceito é o "sinal verde" para o *Hands-on*. Tomando como base a decomposição funcional (7), inicia-se um período de pesquisa de tecnologias que se adequassem aos requisitos já citados, para cada a parte do projeto. Foram estes os módulos estudados: *Fonte de corrente constante DC*, *Multiplexador / Demultiplexador*, *Microcontrolador (Firmware e ADC)*, *Protocolo de comunicação com MATLAB e condicionamento de sinal*.

As seções seguintes descrevem cada módulo do sistema, apontando os resultados obtidos em testes e simulações, justificando as escolhas dos componentes e apontando o comportamento esperado de cada, simulado e testado em protoboard.

3.2.2.1 Fonte de corrente

3.2.2.1.1 Fonte de corrente DC

O circuito utilizado neste projeto para fornecer corrente ao sistema consiste num regulador de tensão LM317 configurado de forma a fornecer a tensão desejada, dependendo do valor do resistor colocado entre seus terminais 1 (ADJ) e 2 (VO), como mostrado na Figura 12.

A tensão entre os terminais 1 e 2 (ADJ e VO) do regulador LM317 mantém-se constante em 1,25V, uma vez que a tensão de alimentação no terminal 3 (VI) seja maior que 3V [19]. Nessas condições, pela Lei de Ohm, a corrente de saída I_{OUT} será:

$$I_{OUT} = \frac{1,25}{R} [19],$$

Onde R é o resistor colocado entre os terminais ADJ e VO do regulador (Figura 12). Assim, pode-se controlar a corrente de saída através do valor do resistor colocado no circuito. A corrente na saída do circuito, contudo, será limitada por 2 fatores:

- i) A tensão total entre o terminal VO e o terminal de terra não pode ser superior à tensão de alimentação E (Figura 12). Assim, para que a fonte forneça a corrente inicialmente projetada, a resistência de carga R_{LOAD} deverá respeitar a relação:

$$R_{LOAD} \leq \frac{E}{I_{OUT}} - R$$

- ii) A fonte de alimentação deve ser capaz de fornecer a corrente que se pretende ter à saída do circuito.

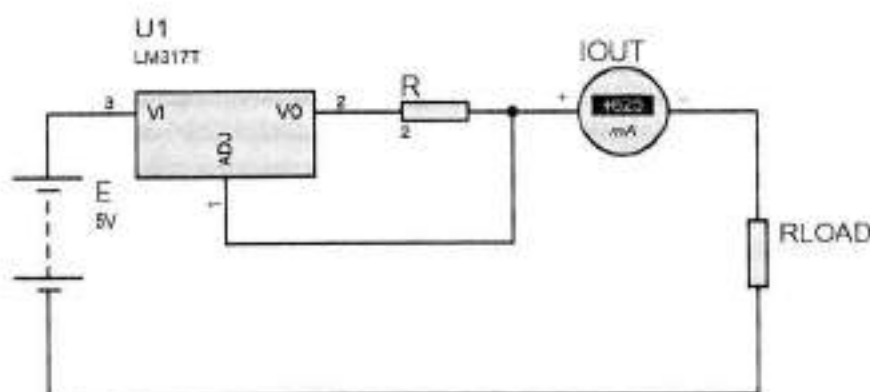


Figura 12 - Fonte de corrente utilizando o regulador de tensão LM317. Na configuração aqui apresentada, tem-se uma corrente de saída de 625mA, evidenciada pelo amperímetro IOU. [Fonte: Autor, 2015]

Na

Tabela 4 a seguir encontram-se os parâmetros utilizados ao implementar-se este circuito em laboratório e a corrente de saída obtida. O pequeno erro relativo de 1,3% deve-se à variação da resistência R em relação a seu valor nominal ($R = 2,03\Omega$, enquanto $R_{NOM} = 2\Omega$, isto é, a variação é de cerca de 1,5% e está dentro da tolerância).

Tensão de entrada (V)	Corrente de saída esperada (A)	Resistência colocada entre os terminais ADJ e VO (nominal, Ω)	Resistência colocada entre os terminais ADJ e VO (medida, Ω)	Corrente de saída (A)
5	0,625	2,00	2,03	0,617

Tabela 4 - Parâmetros utilizados em laboratório para o circuito da fonte de corrente com regulador LM317 e a corrente de saída obtida. [Fonte: Autor, 2015]

Com esses resultados, o circuito foi validado e mostrou-se apropriado para o projeto descrito neste documento.

3.2.2.1.2 Fonte de corrente AC

Para a geração de corrente alternada, tem-se um gerador de onda senoidal seguido de um bloco que converte a tensão de entrada numa corrente de saída que independe da resistência da carga. A seguir, trata-se de cada circuito separadamente.

O gerador de onda senoidal consiste num circuito com amplificador operacional que tem tanto realimentação positiva (rede resistiva, responsável pela oscilação em si) quanto negativa (um filtro *notch* do tipo *twin-T* que atenua sinais na frequência de oscilação). O circuito encontra-se na Figura 13. Assim, forma-se uma oscilação sustentável à frequência dada por: $f = \frac{1}{2\pi RC}$, onde R e C são componentes do filtro *notch* [20], conforme visto na Figura 13.

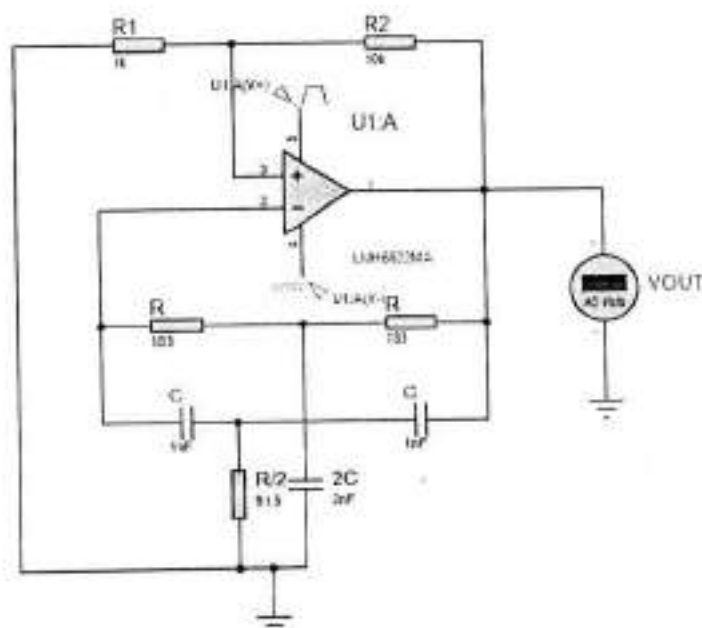


Figura 13 - Circuito do gerador de onda senoidal com filtro notch tipo twin-T, ajustado para que a tensão VOUT oscile a 870kHz. [Fonte: Autor, 2015]

De acordo com os resultados encontrados na Seção 3.2.2.7, a frequência de ressonância das placas está em aproximadamente 870kHz e, portanto, a fonte de corrente AC foi dimensionada para operar nessa frequência. Em laboratório, utilizou-se o amplificador operacional modelo LMH6622 por ter largura de banda suficientemente grande para a aplicação (160MHz) [21]. Na Figura 14, tem-se o resultado simulado para a tensão de saída do gerador de onda senoidal. Na simulação, a tensão de saída oscilou a 800kHz em vez de 870kHz (erro relativo de 8,05%), devido ao efeito de *slew rate* do amplificador operacional.

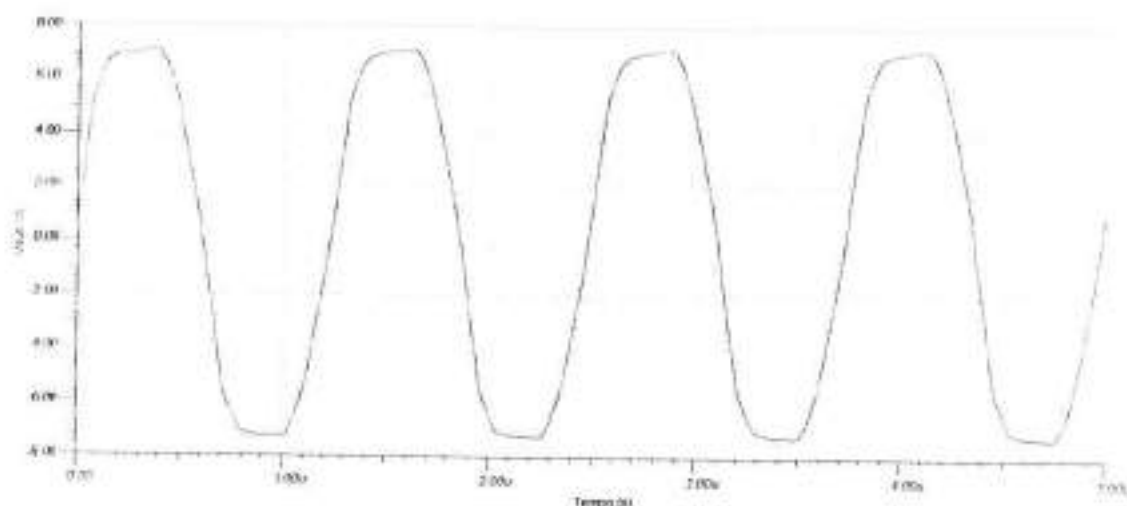


Figura 14 - Tensão de saída simulada para o circuito do gerador de onda senoidal com filtro notch twin-T, com frequência de 800kHz. [Fonte: Autor, 2015]

O bloco de conversão tensão – corrente posto na saída do gerador de tensão senoidal é um circuito simples com amplificador operacional, mostrado na Figura 15. Nessa configuração, o amplificador operacional fará com que sua entrada não-inversora (terminal 5) tenha tensão nula, devido ao efeito de curto-circuito virtual. Portanto, a tensão sobre o resistor R sempre será V_{IN} , e a corrente de saída I_{OUT} será sempre dada pela relação: $I_{OUT} = \frac{V_{IN}}{R}$, independentemente da resistência de R_{LOAD} . Na Figura 15, pode-se verificar que a corrente I_{OUT} é de 10mA_{RMS}, para tensão de entrada V_{IN} de 1V_{RMS} e resistor R de 100Ω.

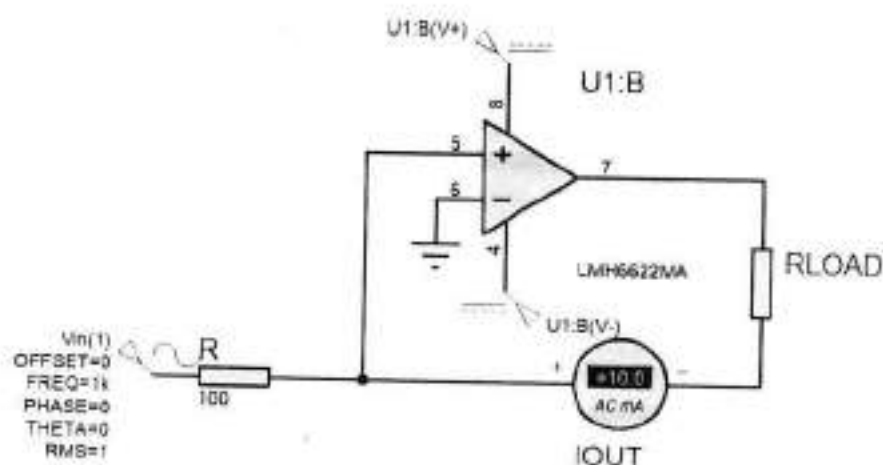


Figura 15 - Circuito do conversor tensão-corrente utilizando amplificador operacional. [Fonte: Autor, 2015]

O circuito é limitado, contudo, pelas tensões de alimentação do amplificador operacional. Em outras palavras, o produto $R_{LOAD} \cdot I_{OUT}$ não pode exceder a tensão V_+ nem ser menor que a tensão V_- , o que impõe as seguintes restrições para a resistência de carga R_{LOAD} :

$$R_{LOAD} \leq \frac{V_+}{I_{OUT}} \text{ e } R_{LOAD} \leq \frac{|V_-|}{|I_{OUT}|}$$

Na Figura 16 vê-se a associação do gerador de tensão senoidal com o conversor tensão-corrente, configurada para ter na saída uma corrente alternada que não depende da resistência da carga.

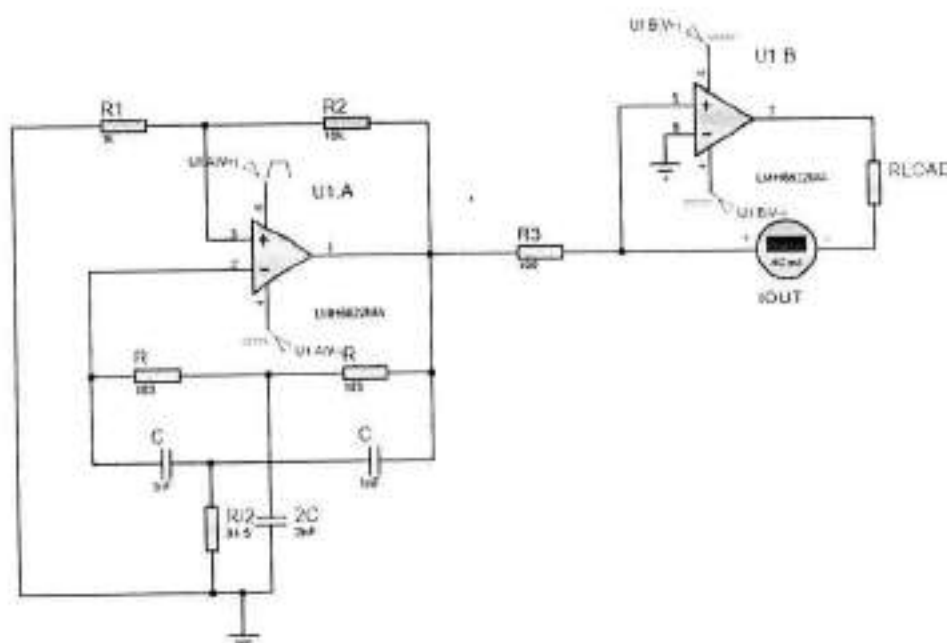


Figura 16 - Fonte de corrente AC, configurada para fornecer corrente de saída I_{OUT} à frequência de 870kHz. [Fonte: Autor, 2015]

A seguir, na Figura 17 vê-se a corrente de saída em função do tempo. Note-se que o circuito opera como esperado tanto em relação à frequência do sinal de saída (800kHz, como a tensão de saída do oscilador mostrada na Figura 14) quanto em relação à amplitude da corrente de saída, que corresponde à tensão de saída do oscilador dividida por R_3 (Figura 16).

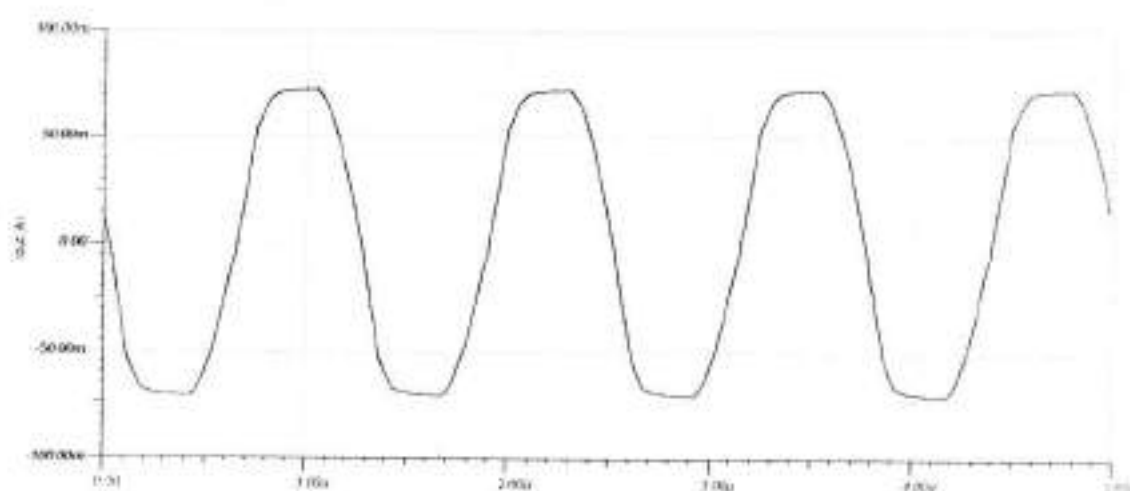


Figura 17 - Acima, a corrente de saída simulada para o circuito da fonte de corrente alternada, com frequência de 800kHz [Fonte: Autor, 2015]

Após essas simulações, o circuito da fonte de corrente alternada mostrou-se adequado para aplicação no projeto aqui descrito.

3.2.2.2 Multiplexação de corrente

3.2.2.2.1 Conjunto de transistores NMOS para multiplexação de corrente

A multiplexação da corrente entre os pares de eletrodos não pode ser realizada diretamente com o circuito integrado 4067, que é um multiplexador analógico de 16 canais. Isto porque a corrente máxima que pode fluir em seus canais é de 10mA [22] e, sendo a estrutura avaliada de material metálico, é necessário aplicar à mesma corrente da ordem de 1A para que se tenham tensões passíveis de ser lidas.

Desse modo, a multiplexação da corrente se dá através de um conjunto de transistores NMOS IRLML2402, associados aos pares e ligados aos terminais da fonte de corrente e a cada um dos eletrodos (na Figura 18 tem-se uma versão simplificada do conjunto, com apenas 3 pares de transistores). Os transistores NMOS foram escolhidos por suportarem uma corrente de até 2,2 A entre seus terminais de dreno e fonte (I_D) [23]. Os terminais de seleção Par 0, Par 1 e Par 2 são controlados pelo multiplexador analógico 4067. Se os terminais de seleção estiverem "flutuando" (isto é, não estiverem acionados), os transistores permanecem desativados devido aos resistores *pull-down* R0, R1 e R2. Se, por exemplo, o Par 0 for acionado (isto é, Par 0 em nível

de tensão alto e Par 1 e Par 2 em nível baixo), ativam-se os transistores Q1 e Q4. Com isso, o Eletrodo 0 é ligado ao terminal positivo da fonte de corrente e o Eletrodo 1 é ligado ao terminal negativo de corrente, fazendo com que a corrente flua entre esses 2 eletrodos.

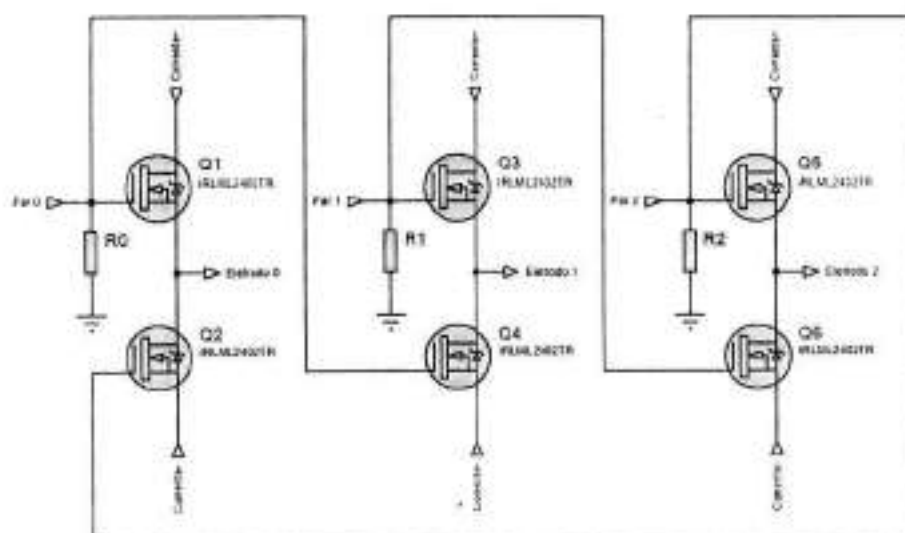


Figura 18 - Conjunto de transistores NMOS para multiplexação de corrente entre os eletrodos. Aqui estão mostrados apenas 3 pares de transistores que multiplexam 3 eletrodos. [Fonte: Autor, 2015]

Na Figura 19, tem-se uma visão geral do sistema apresentado nesta Seção. As entradas Sel0..Sel15, se ativadas (nível lógico 1) acionam os pares de eletrodos da seguinte maneira: Sel0 ativa o par 0 (eletrodos 0 e 1); Sel1 ativa o par 1 (eletrodos 1 e 2); e assim por diante, até Sel15, que ativa o par 15 (eletrodos 15 e 1).

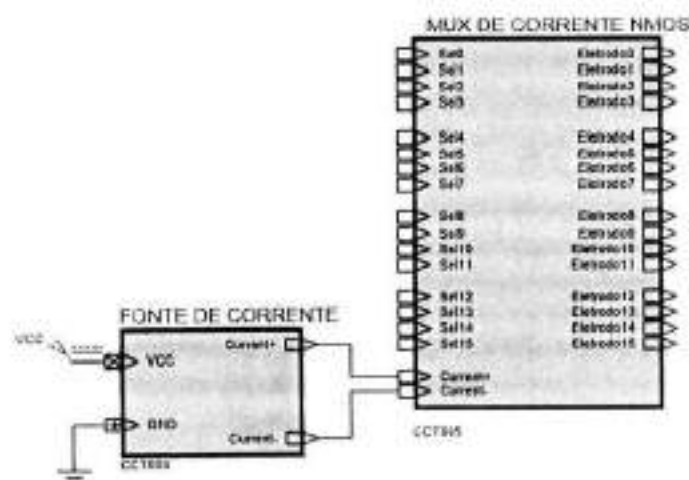


Figura 19 - Blocos representando a fonte de corrente associada ao conjunto de transistores NMOS para multiplexação de corrente. [Fonte: Autor, 2015]

3.2.2.2.2 Bloco completo de multiplexação de corrente

O bloco multiplexador de transistores NMOS descrito na Seção 3.2.2.2.1 é associado a um multiplexador analógico de 16 canais (circuito integrado 4067) para selecionar o par de eletrodos através do qual se aplica a corrente à estrutura avaliada. A Figura 20 mostra o sistema em questão.

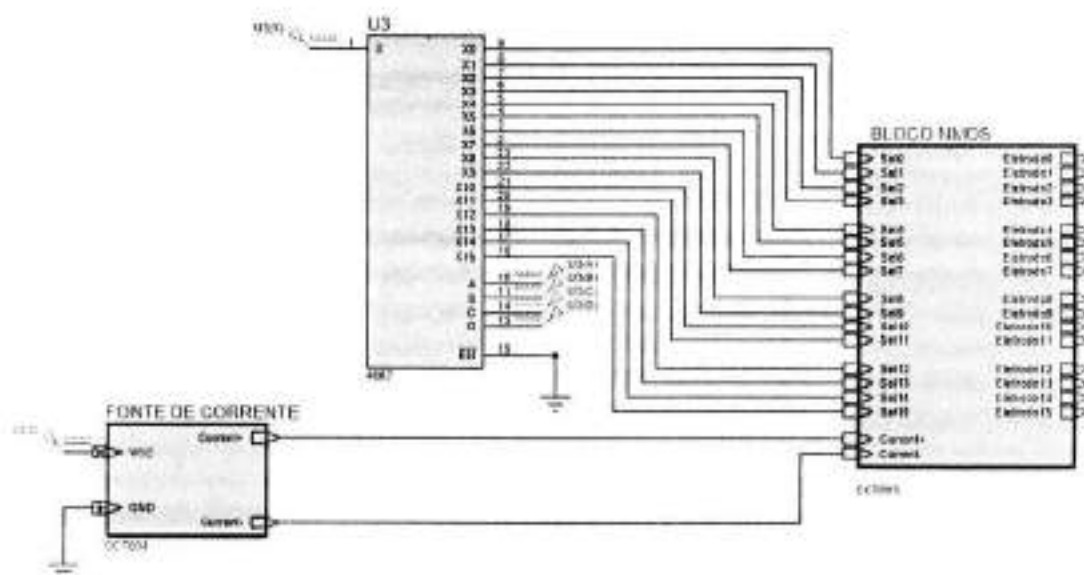


Figura 20 - Sistema de multiplexação de corrente, contando com a fonte de corrente, o multiplexador analógico 4067 e o conjunto multiplexador de transistores NMOS. [Fonte: Autor, 2015]

O bloco da fonte de corrente pode ser tanto o circuito descrito na Seção 3.2.2.1.1 (fonte de corrente contínua) quanto o descrito na Seção 3.2.2.1.2 (fonte de corrente alternada), a depender da aplicação desejada. Suas entradas são a tensão de alimentação (V_{CC}) e a referência de terra (GND), e por seus terminais de saída flui a corrente que será aplicada à estrutura avaliada.

O circuito integrado 4067 é um multiplexador analógico de 16 canais. Através de suas chaves de seleção DCBA, escolhe-se qual das saídas (X0..X15) será ligada à entrada X.

O bloco multiplexador de transistores NMOS é utilizado para transportar a corrente da fonte de corrente ao par de eletrodos selecionado. Diferentemente do caso da multiplexação do voltímetro (Seção 3.2.2.3), o sinal que se deseja aplicar aos

eletrodos pode ser muito mais intenso do que o circuito integrado 4067 pode suportar (o multiplexador analógico 4067 suporta corrente máxima de 10mA, sendo 1A a corrente da fonte DC e 100mA_{RMS} a corrente da fonte AC). Por isso, faz-se necessário o uso do bloco multiplexador de NMOS para ligar os eletrodos à fonte de corrente, sendo o circuito integrado 4067 utilizado para selecionar os transistores NMOS por onde fluirá a corrente.

Na Figura 21, têm-se os resultados de testes de funcionamento do circuito em simulação e em laboratório. Fazendo $DCBA = (0111)_2 = (7)_{10}$, seleciona-se o par de eletrodos 7 e a corrente da fonte flui entre os eletrodos 7 e 8. Tanto em simulação quanto em laboratório, verificou-se que quase a totalidade da corrente da fonte (cerca de 97,7% da corrente original) flui entre os eletrodos do par 7. Verificou-se, contudo, uma pequena corrente fluindo em sentido contrário nos pares que não foram selecionados; tal efeito se deve ao fato de os transistores desativados não constituírem circuitos abertos perfeitos (isto é, não terem resistências reais infinitas).

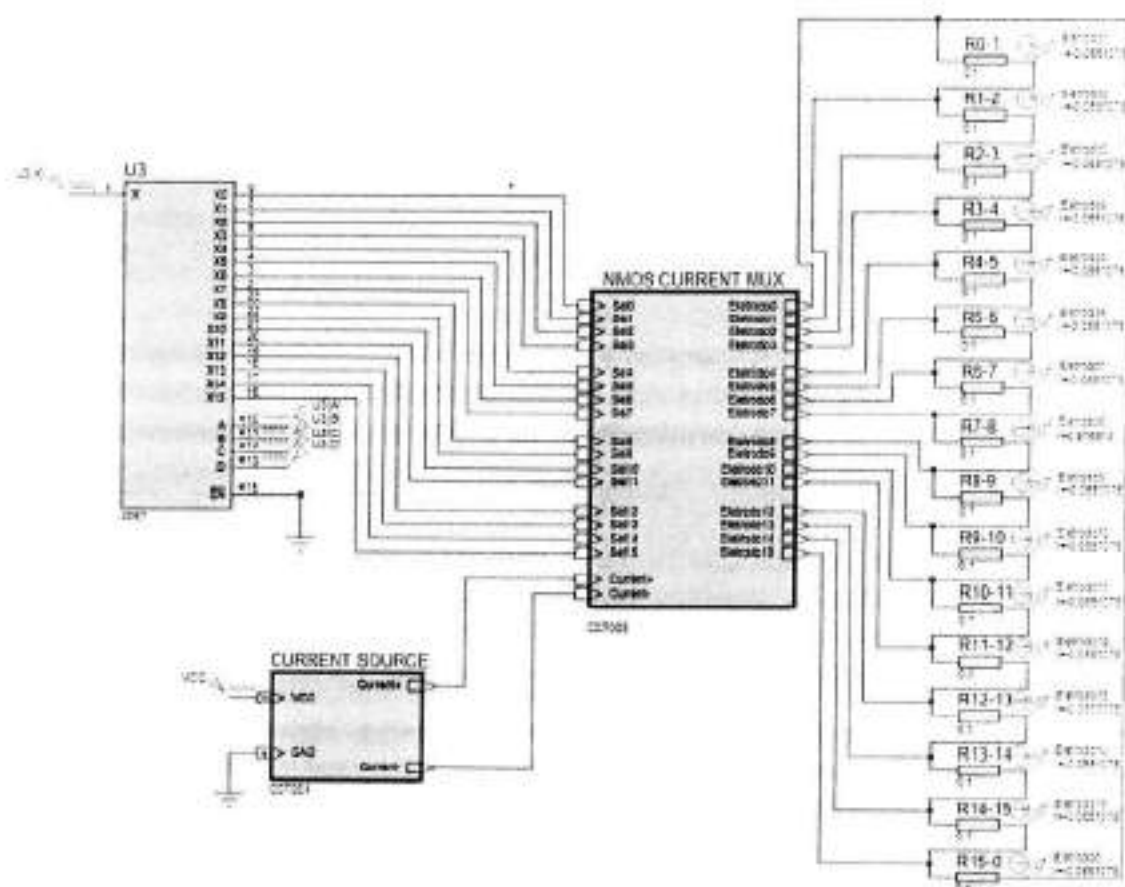


Figura 21 - Resultados dos testes para o bloco de multiplexação de corrente. Acima, a simulação mostrando a configuração das chaves de seleção ($DCBA = (0111)_2 = (7)_{10}$, onde pontos azuis representam 0V e pontos vermelhos representam VCC) e as correntes em cada eletrodo, evidenciando que praticamente toda a corrente da fonte (0.976614 A) flui no par de eletrodos 7, correspondente ao resistor R7-8. [Fonte: Autor, 2015]

Por fim, pode-se dizer que tais resultados validam o bloco de multiplexação de corrente e provam-no adequado para aplicação no projeto.

3.2.2.3 Hardware de aquisição de tensões

A parte do projeto de hardware responsável pela aquisição das tensões que servem de entrada para o algoritmo de reconstrução de imagem é representada a seguir (Figura 22):

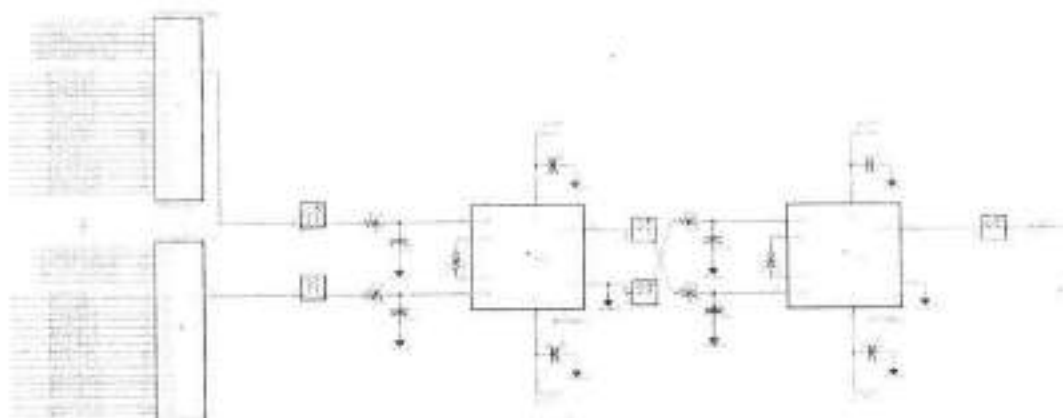


Figura 22 - Esquemático do hardware de aquisição das tensões nos eletrodos.
[Fonte: Autor, 2015]

Ela é composta por 2 principais módulos: o de multiplexação e o de condicionamento e do sinal.

Multiplexação

2 multiplexadores 4067, o mesmo circuito integrado de 16 canais utilizado na multiplexação de corrente, demultiplexam as tensões entre os eletrodos. Quatro saídas digitais do microcontrolador são responsáveis pela seleção de quais entradas são habilitadas em cada demux de modo que a cada medição, pares de eletrodos consecutivos sejam habilitados.

Condicionamento de Sinal

A tensão entre estes é então filtrada através de filtros RC passa-baixa (Low-pass filter - LPF). As especificações de R e C são 470Ω e 200nF de tal modo que sua frequência de corte seja:

$$f_c = \frac{1}{2\pi \cdot RC} = 1,7kHz$$

Essa filtragem melhora consideravelmente a conversão AD dos sinais pois retira ruídos introduzidos pelos CIs e pelo próprio ambiente. O sinal passa então por 2 estágios de amplificação através dos amplificadores de instrumentação INA188 (apresentados na Seção 3.2.2.4), da Texas Instruments. A resistência disposta entre os pinos R_{G1} e R_{G2} de cada AI é quem determina o ganho deles. Cada estágio está

configurado para um ganho de 1000 ($R_G = 50\Omega$) e 5 ($R_G = 12,5k\Omega$), respectivamente. Os AIs são alimentados simetricamente em +2,5V e -2,5V, entre eles existe um outro LPF e estes estão em configuração de amplificação diferencial, ou seja, o sinal amplificado é a diferença entre os sinais de cada par de eletrodos selecionados.

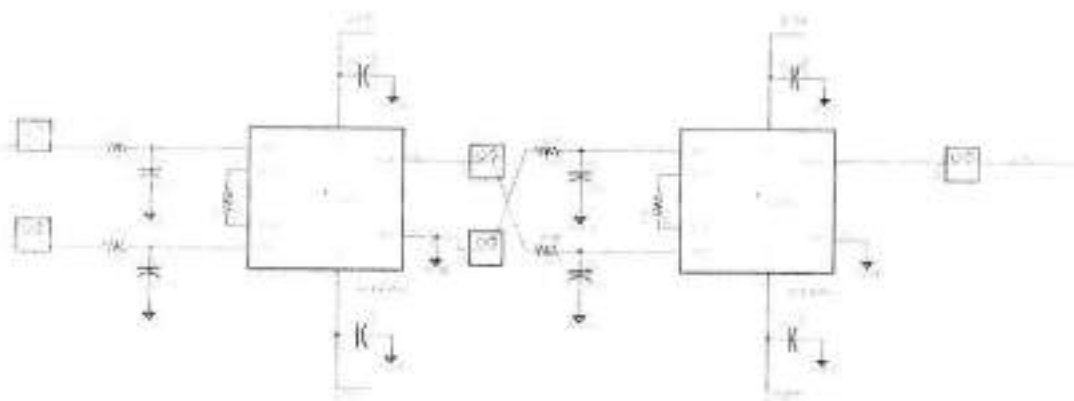


Figura 23 - Esquemático da parte de condicionamento de sinal, formada por 2 estágios de filtragem e amplificação. [Fonte: Autor, 2015]

Vale a pena ressaltar que, assim como em toda PCB, entre cada módulo foram inseridos jumpers com o objetivo de tornar o protótipo escalável e modular.

Testes

Alguns pontos foram considerados possíveis gargalos para o desenvolvimento desta parte e por isso foram alvos de maior detalhamento na simulações e testes em protoboard. São estes:

- *Se o 4067 (mux / demux) seria capaz de transmitir sinais de magnitude de dezenas de μV .*

Positivo. Foram injetados até $10\mu V$ nas entradas do 4067 e na saída foram detectados os mesmos $10\mu V$, sem qualquer perda ou introdução de ruído relevantes.

- *Se o AI seria capaz de amplificar sinais desta ordem de magnitude.*

Positivo. Esse ponto foi determinante na escolha do INA188 uma vez este possui especificações que atendiam.

- Se o sinal de interesse passaria sem perdas relevantes pelo filtro LPF, uma vez que este pode ter magnitude de $5\mu V$.

Positivo. Com uma impedância de entrada em modo diferencial de $100\text{ G}\Omega$, $1011/470 = 2 \cdot 10^8$ maior que R do LPF, toda a tensão fica na entrada do INA188 e a corrente nesta malha é desprezível de tal forma que não se observam perdas relevantes do sinal no LPF.

3.2.2.4 Condicionamento de Sinal

A leitura das tensões nas bordas da estrutura avaliada constitui parte central do princípio utilizado neste projeto. Deste modo, é necessária precisão e confiabilidade na medição dessas tensões.

As estruturas que se avaliam neste documento, contudo, são de material metálico e apresentam resistência total baixíssima (com um multímetro, mediram-se cerca de $100\text{ m}\Omega$). Assim, ainda que a corrente aplicada seja alta (da ordem de 1 A), as tensões lidas nas bordas das placas vão de $5\mu V$ a $400\mu V$, aproximadamente, e as entradas analógicas do microcontrolador não são capazes de detectar e medir sinais de tal magnitude.

Com isso, fez-se necessário o acréscimo de um bloco amplificador entre os eletrodos e as entradas analógicas do microcontrolador. Uma vez que a menor tensão a ser lida é de cerca de $5\mu V$, um bloco amplificador de ganho de 5000 apresentaria uma tensão de 25 mV às entradas do microcontrolador que, por sua vez, mediria o sinal sem problemas. Assim, estabeleceu-se o ganho de tensão desejado como 5000 e o bloco foi implementado com 2 circuitos integrados INA188, amplificadores de instrumentação com entradas diferenciais cujo ganho é configurado pela resistência de entrada R_G , de acordo com a seguinte relação:

$$G = 1 + \frac{50\text{ k}\Omega}{R_G} \quad [24]$$

O primeiro estágio tem ganho 1000 (configurado pelo resistor $R_G = 50\Omega$), enquanto o segundo estágio tem ganho 5 (configurado pelo resistor $R_G = 12,5\text{ k}\Omega$). A Figura 24 apresenta o circuito do bloco amplificador de ganho 5000, enquanto a Figura 25

apresenta a curva da tensão de saída do bloco em função das tensões aplicadas na entrada.

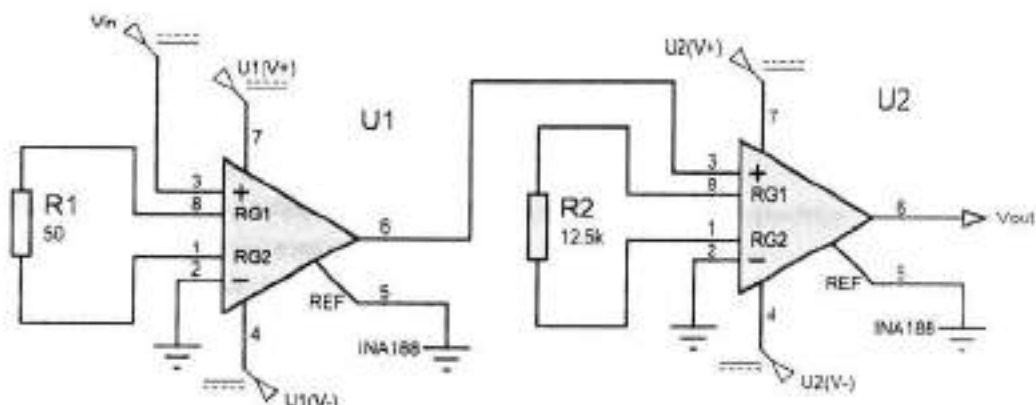


Figura 24 - Circuito do amplificador de ganho 5000, composto por 2 estágios com um amplificador INA188 cada. Os amplificadores de instrumentação são alimentados com $\pm 2,5V$. O primeiro estágio está configurado para ter ganho 1000, enquanto o segundo está configurado para ter ganho 5. [Fonte: Autor, 2015]

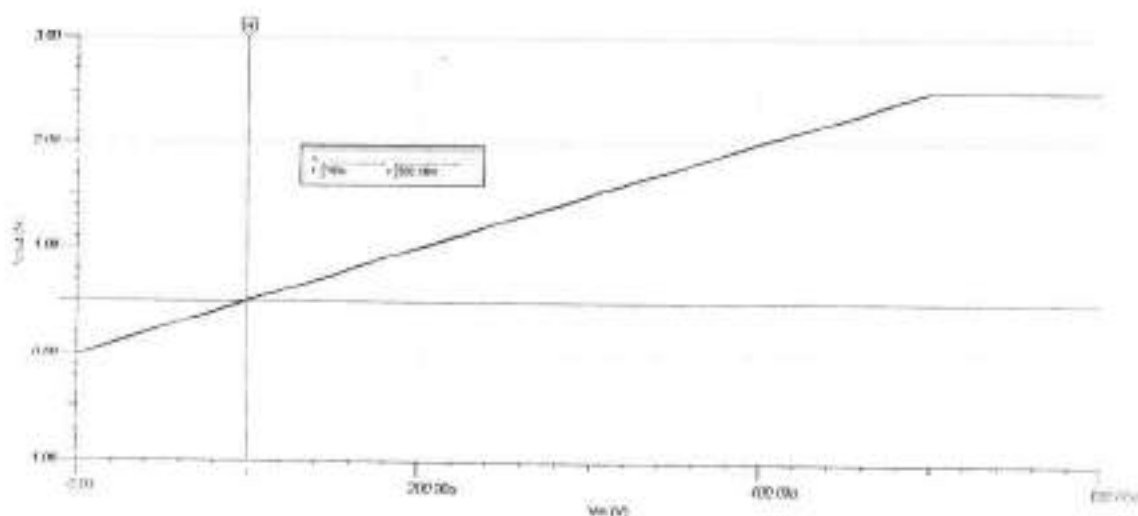


Figura 25 - Curva de transferência do amplificador de 2 estágios da Figura 24. Note-se que no ponto A o ganho é 5000 (tensão de saída de 500mV tensão de entrada de 100 μV), com tensão de offset introduzida pelos amplificadores de instrumentação INA188 de cerca de 0,18mV, resultando numa tensão de saída de 500mV. Note-se ainda a saturação para tensões de saída maiores que 2,5V, o que se deve às tensões de alimentação de $\pm 2,5V$. [Fonte: Autor, 2015]

3.2.2.5 Microcontrolador - Firmware e ADC

Foi decidido por utilizar o microcontrolador da série Tiva C da fabricante Texas Instruments (TI). Entre os fatores que motivaram o uso deste microcontrolador foi a disponibilidade no momento do desenvolvimento do projeto, precisão do conversor analógico digital interno, modo de baixa potência e quantidade de pinos de entrada e saída [25]. O baixo custo de obtenção da placa de desenvolvimento TIVA C Launchpad (cerca de doze dólares) foi fator primordial para a escolha.

O firmware foi desenvolvido em um sistema de desenvolvimento disponibilizado pela TI chamada Code Composer Studio (CCS), que possui recursos como depuração de memória em tempo real e programador integrado. A linguagem de programação utilizada foi o C, pois a biblioteca disponibilizada pela Texas Instruments é fornecida nesta linguagem.

De modo que modificações futuras fossem facilitadas, adotou-se o paradigma de programação de código limpo (*clean code*). A idéia central desta abordagem é permitir que código possa ser lido de forma fluída, como se fosse um texto, de forma que haja o maior desacoplamento possível entre os módulos constituintes do código. Apesar de adicionar complexidade, esta abordagem permite com que o código seja reutilizável e sua manutenção seja mais simples. Na figura abaixo é exibida a estrutura modular do firmware.

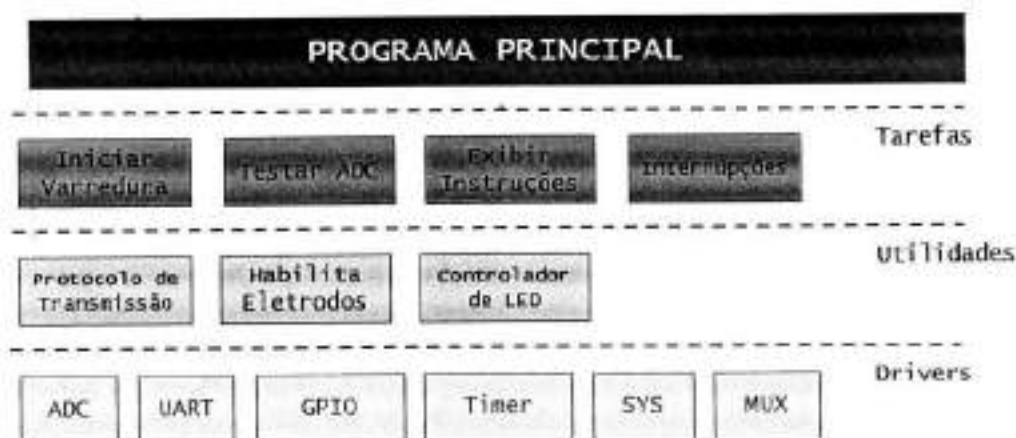


Figura 26 - Estrutura de Camadas do Firmware. [Fonte: Autor, 2015]

Todas instruções específicas do microcontrolador ficam na camada de drivers. A camada de utilidades é composta de módulos que fornecem meios de acesso aos drivers, criando uma camada de abstração da camada tarefas. A camada de tarefas são as atividades principais que o microcontrolador deve fazer, tais como iniciar varredura dos eletrodos, exibir instruções e opções de comando via canal de comunicação, entre outros.

Pelo fato das camadas de tarefas e utilidades não possuírem instruções específicas do microcontrolador utilizado, é possível portar inteiramente o código desenvolvido em C destas duas camadas para outros microcontroladores. Para isto, basta implementar as funcionalidades da camada de drivers utilizando as instruções específicas do outro microcontrolador.

Validação – Conversor Analógico Digital (ADC)

Para validar o conversor analógico digital, foi utilizado um modo aquisição ativado por software. O microcontrolador solicitava a leitura do ADC, aguardava a leitura ser concluída, e exibia via comunicação serial o valor lido, conforme figura abaixo.



Figura 27 - Fluxograma de validação do ADC. [Fonte: Autor, 2015]

Foram medidos dois extremos: valor de 0V (GND) e 3.3V (VCC). Por ter resolução de 12bits, os valores possíveis vão de 0 a 4095 ($2^{12}-1$). Desta forma, era de se esperar que com entrada de 0V no canal do ADC fosse lido 0, e que com entrada 3.3V fosse lido 4095. Abaixo são exibidos os resultados.

Número da Amostra	Tensão de Entrada [V]	Resultado Esperado da Conversão	Resultado Obtido
1	0	0	2
2	0	0	2
3	0	0	1
4	0	0	1
5	0	0	1
6	0	0	1
7	0	0	1
8	0	0	1
9	0	0	1
10	0	0	0
11	0	0	1
12	0	0	1
13	0	0	0
14	0	0	2
15	0	0	1
16	0	0	1
17	0	0	1
18	0	0	1

Tabela 5 - Validação do Conversor A/D aplicando entrada de 0V.

Número da Amostra	Tensão de Entrada [V]	Resultado Esperado da Conversão	Resultado Obtido
1	3.3V	4095	4078
2	3.3V	4095	4077
3	3.3V	4095	4077
4	3.3V	4095	4080
5	3.3V	4095	4077
6	3.3V	4095	4081
7	3.3V	4095	4079
8	3.3V	4095	4078
9	3.3V	4095	4079
10	3.3V	4095	4074
11	3.3V	4095	4077
12	3.3V	4095	4082
13	3.3V	4095	4082
14	3.3V	4095	4078
15	3.3V	4095	4080
16	3.3V	4095	4078
17	3.3V	4095	4082
18	3.3V	4095	4078

Tabela 6 - Validação do Conversor A/D aplicando entrada de 3.3V.

É observada uma leve variação entre os valores esperados na teoria e os observados na prática. Cada incremento no valor corresponde a $\frac{V_{CC}}{4095} = \frac{3.3}{4095} = 0.8\text{mV}$. Assim, em torno de 0V há uma variação de cerca de 1.6mV, e em torno de 3.3V há uma variação de no máximo 16,92mV. Na prática, esta variação seria reduzida com o uso de filtros passa baixa na entrada do conversor analógico digital.

3.2.2.6 Protocolo de comunicação

O protocolo implementado considera o microcontrolador como sendo o servidor das requisições. O microcontrolador será responsável por receber mensagens de requisição e por respondê-las de acordo com uma sequência pré-programada. O microcontrolador executa uma rotina de checagem de recebimento de mensagens a cada 1s, cuja indicação visual é feita pela mudança de estado do LED azul da placa da TIVA C. O microcontrolador está programado a responder as seguintes mensagens:

- *Mensagem em ASCII: '1' (0x39 em hexadecimal)*

Indica que o microcontrolador deve iniciar a varredura dos eletrodos, designada como Tarefa 1. Em caso de recebimento correto, o microcontrolador enviará uma string de confirmação, com o formato "OK_INICIAR\r\n". Em seguida serão enviados dados no formato descrito no item Tarefa 1 – Varredura dos Eletrodos.

- *Qualquer outra mensagem em ASCII*

O microcontrolador enviará uma mensagem ao solicitante com informações das opções disponíveis.

É possível criar mensagens específicas para futuras implementações, bastando definir a correspondência entre o caractere enviado e a tarefa executada.

Após enviar a string de confirmação, o microcontrolador responderá com uma mensagem contendo o frame de dados, com medições do ADC e informações sobre os eletrodos. A figura abaixo ilustra o formato dessa mensagem.

Quantidade de elementos na mensagem	Frame de dados	LRC do frame de dados
-------------------------------------	----------------	-----------------------

Figura 28 - Composição de uma mensagem da varredura de eletrodos [Fonte: Autor, 2015]

A quantidade de elementos na mensagem é a quantidade de bytes que serão enviados pelo microcontrolador, incluindo o LRC e o próprio byte de quantidade de elementos. Desta forma, o primeiro valor recebido após "OK_INICIAR\r\n" é um byte contendo o tamanho do frame de dados mais 2.

Dado um par de eletrodos de corrente injetando corrente na placa a ser testada, é feito um número X de medições de tensão. As informações das X medições de tensão para um único par de eletrodos de corrente constituem um frame de dados. A figura abaixo ilustra uma sequência de bytes a serem recebidos para no começo do frame de dados, representando uma medição de tensão. Os bytes de dados são separados por 0x20, ou o caractere espaço em ASCII.

1	2	3	4	5	6	7
0x20	Valor Lido pelo ADC	0x20	Ganho	0x20	Eletrodo Corrente -	0x20
8	9	10	11	12	13	14
Eletrodo Corrente +	0x20	Eletrodo Tensao +	0x20	Eletrodo Tensao -	0x20	Numero da Medida

Figura 29 - Composição do Frame de Dados para uma Medição de Tensão [Fonte: Autor, 2015]

O LRC (Longitudinal Redundancy Check) é um algoritmo de detecção de erro utilizado em telecomunicações. Apesar de não corrigir erros, ele é útil para identificar falhas na mensagem recebida. A implementação do LRC consiste em o microcontrolador calcular o LRC dos bytes do frame de dados, desconsiderando os bytes 0x20 de separação. O microcontrolador envia no final da mensagem o LRC conforme figura abaixo, e o receptor deve receber a mensagem, calcular o LRC e comparar com o LRC enviado pelo microcontrolador. Caso haja divergência, o receptor requisita o reenvio da mensagem.

...	0x20	Numero da Medida	0x20	LRC	V	Va
-----	------	------------------	------	-----	---	----

Figura 30 - Formato do final de mensagem, com LRC [Fonte: Autor, 2015]

O LRC é implementado através de uma série de XORs sobre os bytes de um buffer. Abaixo é mostrado o algoritmo utilizado pelo microcontrolador e pelo MATLAB.

```
uint32_t calculaLRCBuffer(uint32_t *txbuffer, uint32_t size){
    uint32_t LRC = 0;
    int i;
    for (i = 0; i < size; i++)
    {
        LRC ^= txbuffer[i];
    }
    return LRC;
}
```

Figura 31 - Algoritmo de LRC em C. [Fonte: Autor, 2015]

```
function y = calculaLRC(x)
LRC = 0;
size = numel(x);
for n=2:(size-1)
    LRC = bitxor(LRC,x(n));
end
if x(size) == LRC
    y = 1;
else
    y = 0;
end
```

Figura 32 - Algoritmo de LRC em Matlab Script. [Fonte: Autor, 2015]

Caso haja correspondência no receptor entre o LRC recebido e o calculado, o receptor deve responder o caractere 'k' (0x6B em hexadecimal) para receber a próxima mensagem. Havendo divergência, o receptor envia qualquer outro caractere para receber a mensagem novamente.

O microcontrolador enviará uma mensagem toda vez que as medidas de tensão de um par de corrente forem concluídas. Quando todas as medidas forem concluídas e a última mensagem for enviada e recebida com êxito, o microcontrolador enviará a string "SUCESSO\r\n", sinalizando fim de varredura.

3.2.2.7 Testes de implementação em corrente alternada

Uma implementação alternativa seria com a injeção de sinais AC na estrutura. Seguindo a mesma ideia para a implementação DC, neste caso a fonte de corrente constante seria substituída por um gerador de corrente senoidal, de amplitude e frequência constantes. Essa abordagem pode ser encontrada em artigos e livros [1, 2, 3, 7], principalmente naqueles onde se pretende medir condutividade de soluções, uma vez que a condutividade das soluções é afetada pela mobilidade dos íons e, caso uma tensão DC seja aplicada, estes íons tendem a se aproximar dos eletrodos, causando uma polarização na solução e, eventualmente, reduzindo sua condutividade.

Os testes realizados pela equipe tinham por objetivo observar a magnitude do sinal medido entre os eletrodos comparados à corrente injetada na estrutura ou, em outras palavras, a relação entre $V_{\text{eletrodos}}/I$. A importância desta medição é observar regiões "ótimas" para a estrutura, em que a corrente injetada possa ser minimizada e garantindo que a tensão ainda possa ser medida. De imediato, isso traria uma redução do consumo energético do dispositivo, ponto crucial em dispositivos portáteis.

Outra motivação para o teste foi o fato de a estrutura alvo do projeto, os metais, apresentam resistência na ordem de $100\text{m}\Omega$ entre 2 eletrodos, de tal forma que, para a análise DC, tensões entre os eletrodos variando de $5\mu\text{V}$ a $500\mu\text{V}$ foram obtidas com a injeção de 3A constantes na placa de testes. Assumiu-se então que para determinadas faixas de frequência a relação $V_{\text{eletrodos}}/I$ poderia subir relevantemente de forma que a implementação em AC pudesse se apresentar mais vantajosa para os problemas propostos.

Os testes visaram, primeiramente, reconhecer tais regiões de interesse, em que houvessem picos na resposta em frequência do material e, em seguida, fazer rodadas de medições para gerar imagens.

O teste aconteceu de maneira manual e iterativa: a frequência, variando de 0 (DC) a 1 MHz, era ajustada manualmente, garantindo que a corrente I injetada fosse

sempre 10 mA_{ef}. Na primeira iteração, o passo foi de 50 kHz, portanto variando em 0, 50kHz, 100kHz, etc. Em regiões que apresentassem melhor resposta, 2 outras iterações foram realizadas a passo de 10 e 2 kHz, respectivamente. As medições foram feitas em um modelo do multímetro. Os resultados para as placas-alvo foram:

Pico ressonante:		870 kHz
Placa: 4 furos simétricos		
I ₆ (mA _{ef})	V ₁₁ (mV _{ef})	V ₁₁ /I ₆ (Ohm)
10,03	0,15	0,015

Pico ressonante:		878 kHz
Placa: 1 furo central		
I ₆ (mA _{ef})	V ₁₁ (mV _{ef})	V ₁₁ /I ₆ (Ohm)
10,03	0,02	0,002

Tabela 7 - Resultado da resposta em frequência para 2 diferentes placas referência do projeto.

[Fonte: Autor, 2015]

A escolha dos eletrodos foi feita da seguinte maneira: foram testados 4 pares diferentes de eletrodos, tanto na injeção como na medição das senoides. Dentre esses, aquele que apresentou maior sinal de saída, foi selecionado (I₆, V₁₁).



Figura 33 - Foto feita durante teste AC, mostrando toda a infraestrutura utilizada para este. [Fonte: Autor, 2015]

A segunda parte dos testes usou o resultado de 870kHz do pico ressonante do material para gerar imagem a partir dos 16 eletrodos. A tabela a seguir apresenta a matriz de tensões medidas e as imagens geradas a partir desta:

4 furos - Senoidal 870 kHz - 100mA

	1ª Rodada	2ª Rodada	3ª Rodada	4ª Rodada	5ª Rodada	6ª Rodada	7ª Rodada	8ª Rodada	9ª Rodada	10ª Rodada	11ª Rodada	12ª Rodada	13ª Rodada	14ª Rodada	15ª Rodada	16ª Rodada
V1	0,00E+00	5,00E+00	1,00E+01	1,75E+01	0,99E+00	1,84E+01	6,55E+00	3,79E+00	4,91E+00	5,73E+00	7,93E+00	1,00E+00	6,29E+00	3,13E+00	1,78E+00	0,00E+00
V2	0,00E+00	5,00E+00	0,00E+00	1,00E+01	1,50E+01	5,41E+01	4,43E+00	5,89E+00	3,39E+00	4,94E+00	3,69E+00	5,00E+00	5,34E+00	7,24E+00	7,88E+00	2,93E+00
V3	1,55E+00	5,00E+00	0,00E+00	0,00E+00	0,00E+00	1,72E+01	3,89E+00	1,22E+01	3,50E+00	3,85E+01	2,49E+00	5,40E+00	1,25E+01	7,63E+00	3,68E+00	5,93E+00
V4	2,72E+00	5,00E+00	0,00E+00	0,00E+00	0,00E+00	1,51E+01	3,23E+01	1,78E+01	1,77E+01	2,03E+00	1,37E+00	1,15E+00	3,09E+00	5,10E+00	2,06E+01	1,61E+01
V5	1,28E+00	5,40E+00	5,40E+00	5,00E+00	0,00E+00	0,00E+00	3,37E+01	1,81E+00	5,04E+00	5,04E+00	8,79E+00	7,09E+01	3,03E+00	3,36E+01	1,24E+02	3,19E+00
V6	1,00E+00	5,11E+00	1,02E+00	4,79E+00	0,00E+00	0,00E+00	1,72E+00	3,98E+00	6,80E+00	1,13E+00	2,01E+00	1,38E+00	8,57E+00	4,53E+00	5,58E+00	
V7	1,43E+00	1,18E+00	1,64E+00	1,07E+00	6,22E+00	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,73E+00	8,30E+00	6,29E+00	5,09E+00	1,00E+01	3,38E+01	8,49E+01
V8	6,32E+00	1,03E+00	7,78E+00	3,89E+00	3,94E+00	1,56E+01	0,00E+00	0,00E+00	0,00E+00	3,10E+00	6,27E+00	3,68E+00	3,60E+01	1,13E+01	1,03E+00	0,00E+00
V9	4,01E+00	1,75E+00	7,99E+00	1,05E+00	8,78E+00	5,13E+00	1,78E+00	0,00E+00	0,00E+00	2,89E+01	1,75E+01	0,30E+00	1,68E+01	6,59E+00	1,78E+00	
V10	1,99E+00	2,03E+00	8,88E+00	5,89E+00	3,18E+00	7,76E+00	2,84E+00	2,02E+01	0,00E+00	0,00E+00	0,00E+00	0,00E+00	1,88E+01	1,39E+01	4,43E+00	5,58E+00
V11	1,93E+00	1,05E+00	2,48E+00	3,48E+00	1,73E+01	7,41E+00	2,93E+00	4,01E+00	3,55E+00	0,00E+00	0,00E+00	0,00E+00	6,88E+00	1,32E+01	1,86E+00	1,65E+01
V12	1,37E+01	2,05E+00	1,77E+00	3,35E+00	1,00E+00	5,10E+00	2,84E+01	1,81E+01	2,22E+00	3,03E+00	0,00E+00	0,00E+00	0,00E+00	1,11E+01	1,19E+01	1,78E+01
V13	5,91E+00	8,63E+01	1,78E+00	7,48E+01	3,03E+00	1,50E+01	1,14E+01	8,10E+00	3,89E+00	6,43E+00	1,99E+00	0,00E+00	0,00E+00	0,00E+00	5,31E+01	1,78E+01
V14	5,98E+00	4,80E+00	1,12E+00	2,61E+00	1,18E+00	8,57E+00	4,53E+00	5,50E+00	2,00E+00	4,11E+00	4,07E+00	4,79E+00	0,00E+00	0,00E+00	8,69E+00	1,27E+01
V15	1,33E+01	8,30E+01	5,20E+00	1,89E+00	1,09E+01	1,30E+01	2,01E+01	8,43E+01	1,41E+00	3,33E+00	3,80E+00	5,67E+00	6,22E+00	0,00E+00	0,00E+00	0,00E+00
V16	6,80E+00	1,10E+00	1,17E+00	3,67E+00	5,89E+01	1,19E+01	1,93E+00	6,95E+01	6,10E+00	2,10E+00	7,70E+00	8,19E+00	5,04E+00	1,34E+01	0,00E+00	0,00E+00

Tabela 8 - Tensões nos 16 eletrodos, para 16 rodadas de medição, com injeção de corrente alternada senoidal 870kHz. [Fonte: Autor, 2015]

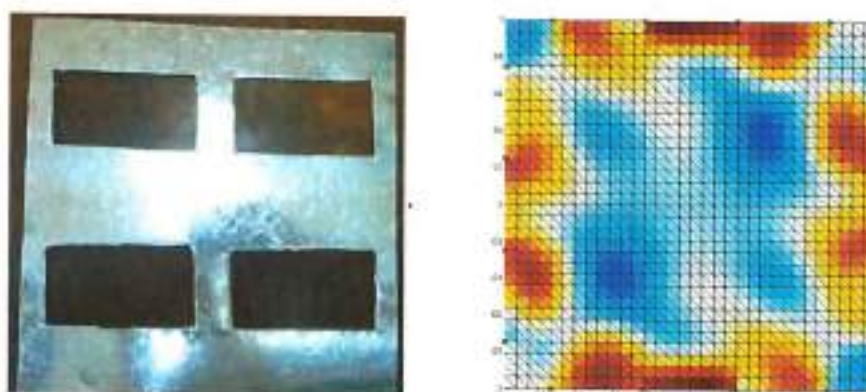


Figura 34 – Imagem da placa de teste e imagem (à esq.) gerada através do EIDORS (à dir.) durante teste AC, em 870kHz, com 128 e 256 números de elementos finitos. [Fonte: Autor, 2015]

As imagens tiveram resultados inferiores aos obtidos no teste anterior (DC – 16 eletrodos) o que, porém, não condena a implementação em AC. Existe uma razoável aproximação com a placa de quatro furos (representados em azul escuro na Figura 34).

É importante também salientar que alguns pontos a respeito do teste:

- O grande objetivo era validar essa outra abordagem (AC), observando em que novos limites de corrente e tensão o dispositivo teria condições de trabalhar;

- Com isso, algumas simplificações foram feitas como na escolha dos eletrodos referência para o teste, resposta em frequência sendo feita de maneira não automática e de baixa precisão;
- Durante o teste, observou-se instabilidade no sinal medido.

Em suma, foram citadas possíveis fontes de erro para o teste e “justificado” o não aprofundamento na solução dessas questões uma vez que, por uma decisão estratégica, a equipe não investiria muito tempo no teste assim que validada, inclusive “grosseiramente”, a implementação AC.

3.2.3 Esquemático e decomposição finais

Como resultado de todo esse processo de pesquisas, simulações e testes, um esquemático foi projetado e a partir deste momento o projeto caminharia para a etapa de criação da PCB. Foram ilustrados o esquemático final e uma decomposição funcional, criada a partir do esquemático. Este foi dividido em várias figuras de forma a facilitar a visualização do leitor e está disponível no Anexo A.

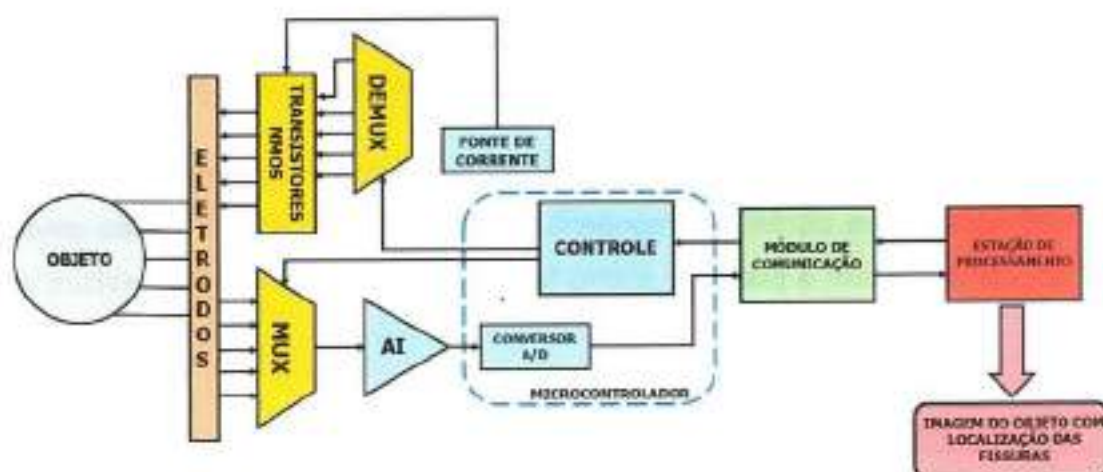


Figura 35 - Decomposição funcional final. [Fonte: Autor, 2015]

3.2.4 Sistema de integração do detector de fissuras com smartphone

Nesta seção é proposta uma arquitetura de sistema que integra o dispositivo de detecção de fissura a um aplicativo mobile, sendo este a interface homem-máquina

por onde o usuário controlaria o dispositivo e visualizaria a imagem resultado do ensaio não destrutivo.



Figura 36 – Arquitetura de sistema de integração do detector de fissura com app mobile e processamento do algoritmo EIT em nuvem. [Fonte: Autor, 2015]

O fluxo de funcionamento do sistema seria:

1. **Iniciar teste:** O app, através do método `iniciarVarredura()`, envia comando de início de do teste para o dispositivo. A comunicação proposta entre eles seria via *Bluetooth*, entre smartphone e microcontrolador (Tiva C) do dispositivo. Essa função hoje é executada ao chamar o script MATLAB seguida do input '1' que inicia a varredura no MATLAB.
2. **Transmissão de frames via protocolo autoral:** Seguindo o protocolo descrito na seção de design do *firmware*, os dados referentes a cada medição são enviados também via *Bluetooth* para o app. Na arquitetura atual implementada, essa comunicação é feita via serial, comunicando Tiva C com MATLAB no PC.
3. **Transmissão das tensões para processamento em nuvem:** A partir de então, a transmissão permanece sendo *wireless* porém em outro canal de comunicação, via *Wi-fi* através de chamadas *http*. O método `gerarImagem()` enviaria a matriz de tensões que serve de entrada para o algoritmo de

reconstrução de imagem e outros parâmetros de entrada do algoritmo: o número de eletrodos, a "precisão" através do número de elementos finitos e formato do modelo de elementos finitos (retangular, circular). A própria aplicação ficaria encarregada de manipular esses dados, transformando os frames recebidos em um objeto do tipo JSON que contenha os dados acima descritos. Exemplo de objeto JSON a ser enviado via http:

```

1 - {
2   "numElectrodes": 16,
3   "numFiniteElem": 256,
4   "format": "square",
5   "voltageMatrix":
6   [
7     0,0,0.12,0.55,0.65,0.55,0.65,...,0,
8     ...,
9     0.93,0.15,0.12,0.55,0.65,...,0,0,0
10  ]
11 }
```

Figura 37 - JSON enviado pelo app para processamento em nuvem. [Fonte: Autor, 2015]

4. **Transmissão da imagem e visualização no app:** Ao receber uma chamada *http*, a API do servidor se encarrega de 'inputar' os parâmetros recebidos do objeto no algoritmo e executá-lo. Como resultado do processamento, teria-se uma imagem de extensão .png que através do retorno da requisição *http* seria enviada à aplicação que a apresentaria na tela do smartphone, concluindo o fluxo de funcionamento do sistema.

Estrutura e tecnologias:

Backend:

Implementar em *Flask* (<http://flask.pocoo.org/>), um framework *python*, seria uma escolha adequada uma vez que nosso backend é basicamente constituído de uma API e um algoritmo de reconstrução de imagem e a implementação de API's em Flask é bastante simples.

Em relação ao algoritmo, este está implementado em MATLAB, ou seja, ou o backend precisa de alguma forma se comunicar com o MATLAB para executá-lo ou o algoritmo todo deveria ser implementado em *python* através de *libraries open source* como

numpy. A segunda abordagem é vantajosa no sentido de se ter um sistema 100% *open source*, não dependente da comunicação e licença comercial do MATLAB. A primeira abordagem, porém, pensando em prototipagem desta primeira versão do produto, é mais vantajosa por ser mais rápida de ser implementada. Entretanto, ela depende de que haja alguma forma de comunicação entre API (backend Flask) e algoritmo (MATLAB). Para isso, o backend faria uso de uma outra *library python open source* chamada *pymatlab* (<https://pypi.python.org/pypi/pymatlab>). Ela abre sessões em que python e MATLAB se comunicam.

Frontend:

Implementado o algoritmo em backend, o frontend ganha bastante em simplicidade de implementação. Suas funções passam a ser basicamente:

- Comunicar-se via *Bluetooth* com o detector de fissuras;
- Ajustar os dados recebidos via protocolo autoral para *object JSON* de forma que o algoritmo receba os dados sem necessidade de maior pré-processamento e que apenas os dados importantes ao algoritmo sejam enviados, diminuindo a taxa de transmissão demandada;
- Comunicar-se com o backend via REST API;

Vantagens e desvantagens da arquitetura proposta:

Vantagens:

- Executar o algoritmo no backend torna a implementação do *mobile app* simples, resumindo-se grosseiramente a uma API REST;
- Desta forma, o *app* pode ser facilmente criado para qualquer sistema operacional sem esbarrar em dificuldades impostas pelo algoritmo e peculiaridades de cada linguagem;
- Melhorias feitas no algoritmo não gerariam o inconveniente de atualizações do *app* para o usuário, dependendo apenas do *deploy* das versões de backend no servidor (tarefa dos desenvolvedores).

Desvantagens:

- Funcionamento apenas *online* e dependência da qualidade da rede no local de uso;

- O sistema ganha custos de manutenção do serviço em nuvem.

3.3. Implementação

A etapa de implementação engloba a concepção, fabricação e montagem da PCB e do conjunto de eletrodos. Em outras palavras, trata-se da implementação do hardware do detector de fissuras. Uma parte das implementações de software foram feitas concomitantemente aos testes de hardware, sendo estes o firmware da Tiva C e o protocolo de comunicação Tiva C ↔ MATLAB. A outra parte da implementação de software faz parte do último sprint do projeto e trata-se das camadas mais alto nível, sendo estas as camadas de back-end e front-end mobile da aplicação com algoritmo de reconstrução de imagem.

3.3.1 Placa de circuito impresso – concepção, fabricação e montagem

O desenvolvimento do layout requereu o uso de uma ferramenta de CAD. Dentre as diversas opções do mercado, a ferramenta escolhida foi o EAGLE 7.4.0, devido ao fácil acesso à ferramenta (licença gratuita para estudantes) e o amplo uso profissional no mercado.

O esquemático foi desenhado de acordo com o resultado da prototipagem do circuito, e notou-se a necessidade da criação de componentes customizados, pois não existia o componente INA188, um amplificador de instrumentação. A tarefa de criar um componente é essencial para o projeto, pois um erro no desenho do encapsulamento pode inviabilizar a placa final. Vide esquemático no Anexo A.

Com o esquemático feito foi realizada a captura do esquemático e geração de netlist, cujo processo é totalmente integrado com o software de desenho de layout no EAGLE. O netlist permite a exibição da conexão entre os diversos componentes, sendo de grande auxílio no roteamento.

A disposição dos componentes foi feita tendo em mente os blocos funcionais do circuito. Posteriormente notou-se que a disposição física dos componentes é crítica

para o sucesso do roteamento, tendo que ser feita algumas poucas modificações para que o roteamento manual ou automático fosse possível.

A maioria das interconexões foram realizadas manualmente, permitindo maior flexibilidade com largura de trilhas e simetria. Após 80% do roteamento ter sido feito manualmente, o auto-router foi executado para terminar o roteamento, juntamente com ajustes de posicionamento de componentes. Vide foto abaixo para o layout final.

Por fim, as Design Rules utilizadas, que são regras de fabricação que envolvem distâncias entre trilhas, tamanho do furo, entre outros, foram as do fabricante SeeedStudio, com distância trilha/trilha e trilha/pad de 6 mils, no mínimo, e largura mínima de trilha também de 6mils. Na tabela abaixo estão detalhadas as Design Rules do fabricante escolhido.

Item	Specs
	Unit: mil
Available Board Thickness	15.7(4 layer except), 23.6, 31.5, 39.4, 47.2, 63.0, 78.7
Board Dimension	—
Board Thickness Tolerance	± 10%
Dielectric Separation thickness	2.95 - 196.85
Minimum trace width	6
Minimum inner trace width (for 4 layer)	8
Minimum trace/vias/pads space	6
Minimum inner trace/vias/pads space (for 4 layer)	10
Minimum silkscreen width	6
Minimum silkscreen text size	40

Minimum distance between pads and silkscreen	7
Out Layer Copper Thickness (4 layers)	1.38-2.76
Inner Layer Copper Thickness (4 layers)	0.67
Drilling Hole Diameter (Mechanical)	11.81 - 250.00
Annular Ring	≥6
Diameter Tolerance (Mechanical)	± 3.148
SMT min Solder Mask Width	3.94
Min Solder Mask Clearance	5.12
Circuit to edge	≥15.76
Minimum distance between inner line/copper and drilling hole	≥11.82
Minimum milling slot	39.385*78.77

Tabela 9 - Design rules do fabricante escolhido

Realizada a checagem das Design Rules e não havendo questões a serem resolvidas, foi executado o CAM Processor para geração dos arquivos Gerber. A geração do Gerber foi checada através de visualizadores online de Gerber, comprovando a geração correta dos mesmos.

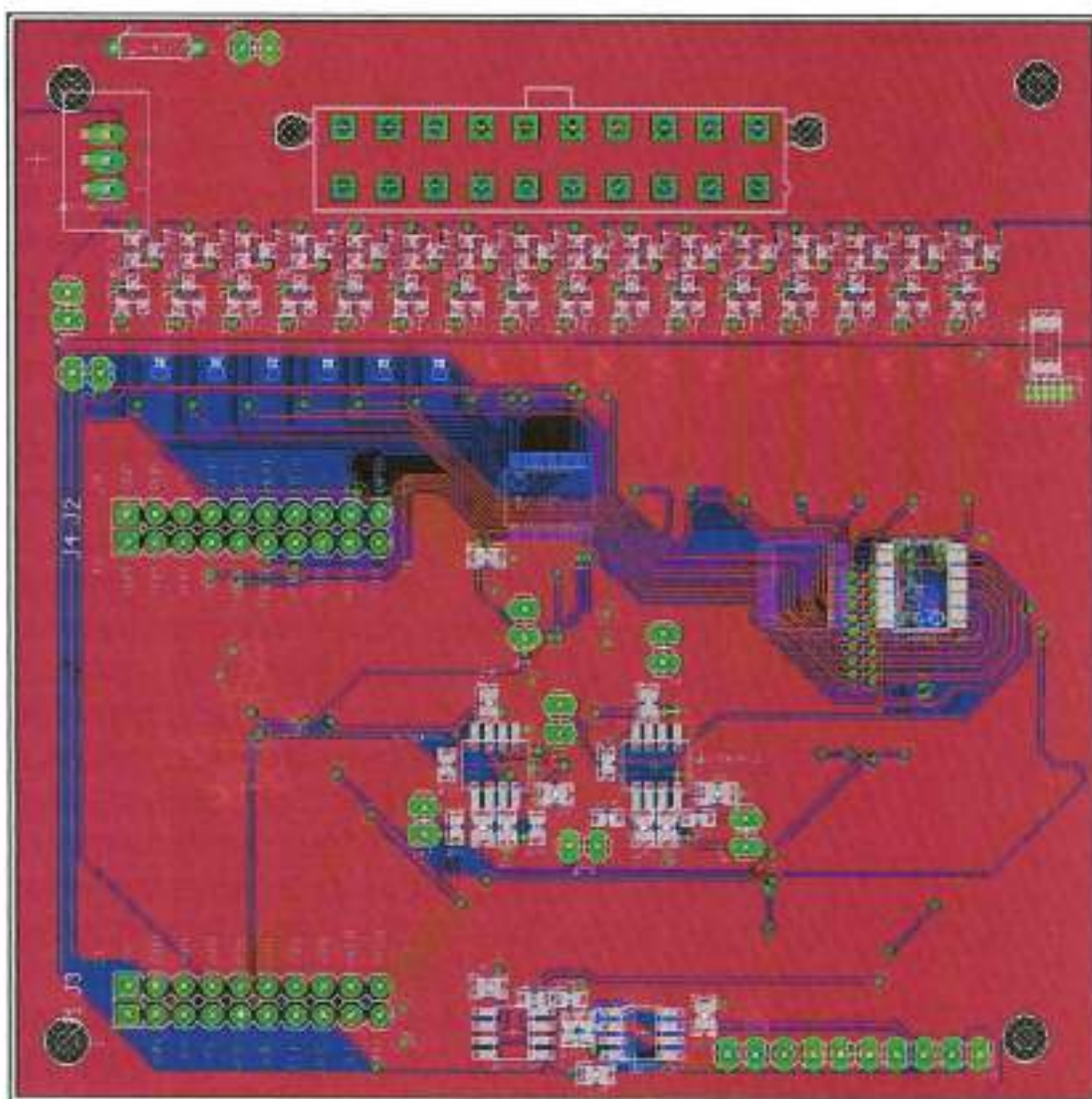


Figura 38 - Layout PCB. [Fonte: Autor, 2015]

Com os Gerbers gerados, foi feito o pedido de fabricação da placa pelo fabricante SeeedStudio. O custo foi de USD 21,90 para 10 placas, com frete de USD 39,00 para o Brasil. A placa chegou em 2 semanas, após o pagamento dos impostos de importação. Fotos da placa no Anexo 08.

A soldagem dos componentes SMD foi realizada utilizando pasta de solda e estação de retrabalho, em conjunto com fluxo aplicado nos terminais do componente. Os multiplexadores 4067 com encapsulamento SSOP-24 foram os componentes mais críticos da montagem, sendo necessário o uso de lupa para checagem da existência de curto circuitos decorrentes da solda.

Não houve dificuldades na soldagem dos MOSFETS FDN337N, porém posteriormente notou-se o mal funcionamento de um deles. A tentativa de troca de componente não foi bem-sucedida, pois a temperatura utilizada do ferro de solda era muito alta para a soldagem, fato comprovado pelo inchaço do componente decorrente da vaporização de componentes químicos. Corrigida a temperatura do ferro de solda, a troca foi realizada com sucesso. A placa com seus componentes soldados pode ser vista nas figuras abaixo.

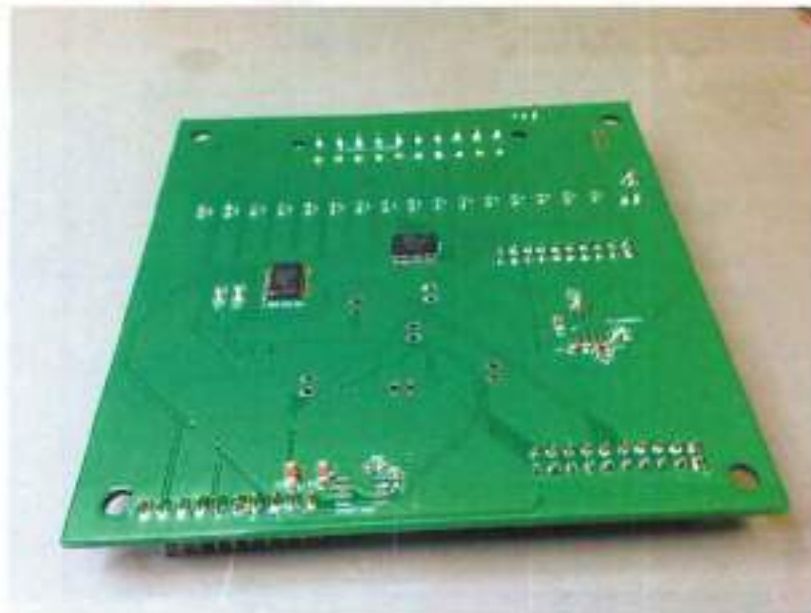


Figura 39 - Imagem inferior da placa finalizada. [Fonte: Autor, 2015]



Figura 40 - Imagem superior da placa finalizada. [Fonte: Autor, 2015]

3.3.2 O conjunto de eletrodos

A decisão estratégica de criar um detector de fissuras em metais fez do design dos eletrodos um dos pontos críticos do projeto. O maior desafio é detectar tensões em metais, considerando que estes têm resistências baixas, da ordem de $100\text{m}\Omega$ entre um par de eletrodos na placa teste e ainda que as resistências de fios e contato são da mesma ordem de magnitude.

Outro agravante neste desafio é que a resistência de contato é aleatória e variante no tempo, de modo que o esquemático da Figura 41 ilustra esta situação:

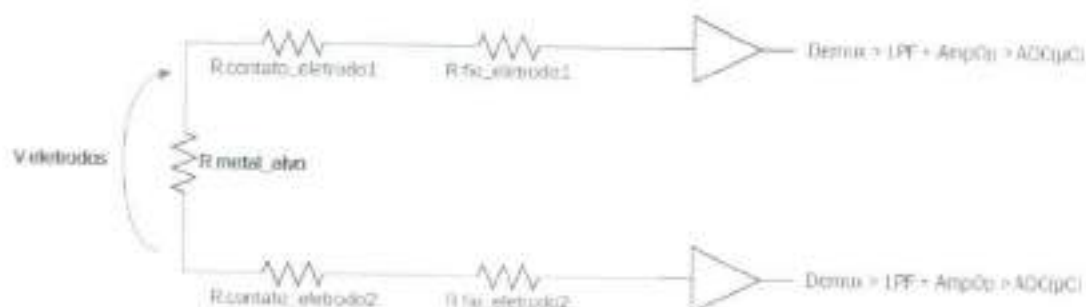


Figura 41 - Modelo esquemático da tensão entre os eletrodos que chega ao ADC do μC .

[Fonte: Autor, 2015]

Durante os testes de prova de conceito (POF – *proof of concept*), comprovou-se a interferência da resistência de fio e de contato na geração das imagens. Lá, testes feitos com pressões diferentes entre os conectores banana e a placa geraram boas diferenças entre as imagens.

Desta forma o design do conjunto de eletrodos visava:

- Minimização da resistência dos fios;
- Minimização da resistência de contatos;
- Tornar as resistências de contato o mais invariante no tempo o possível e;
- Reduzir as diferenças destas entre os eletrodos.

Uma primeira versão de prototipagem foi elaborada com um cabo extensor ATX (40cm – 20/24 pinos) e 16 conectores jacaré. A escolha do cabo baseou-se no fato de este ser usado em fontes de computador e que, por isso, tem baixa resistência (100m Ω – 40cm). Já os conectores jacarés são conectores que, assumiu-se, reduzir as resistências de contato através da pressão que exercem sobre a placa.

A Figura 42 mostra o cabo com um conector ATX macho de um lado e os jacarés na outra ponta, numerados de 1 a 16:



Figura 42 - Cabo do conjunto de 16 eletrodos que se fixam à estrutura alva.

4. RESULTADOS E ANÁLISE DOS RESULTADOS

4.1 Testes e prototipagem

Concluídas as montagens do hardware, já há recursos suficientes para se testar o primeiro protótipo do projeto. Através de scripts MATLAB foi possível criar uma interface automatizada dos dados recebidos pelo programa via serial com o algoritmo de reconstrução de imagem *open source* EIDORS.

O conceito de modularidade utilizado na concepção do hardware através dos jumpers entre as partes possibilitou que os testes do sistema fossem feitos sem complicações. Como descrito na seção de design, o projeto de hardware tinha alguns pontos críticos que foram inicialmente testados e, em seguida, foi feita a integração entre as partes do sistema.

4.1.1 Teste dos multiplexadores de corrente

Para validar o bloco de demultiplexação de corrente, foi utilizado parcialmente o código fonte desenvolvido para a varredura de eletrodos. Este teste exigiu a conexão do microcontrolador TIVA-C com a placa desenvolvida, de forma que o microcontrolador pudesse habilitar sequencialmente os MOSFETs, validando os circuitos integrados de multiplexação e transistores.

Conjuntamente, o amplificador de instrumentação responsável pela medição de corrente através da tensão de shunt foi responsável pela indicação se o par de MOSFET desejado estava ativado. Caso a corrente circulasse entre dois eletrodos, uma tensão proporcional seria gerada na saída do amplificador em questão.

Dessa forma, inicialmente foi feita uma varredura com todos os eletrodos desconectados. O valor esperado para esta configuração era de zero corrente circulante para todos os eletrodos. O resultado obtido foi o esperado, permitindo prosseguir com o teste conectando os eletrodos dois a dois, validando os pares de eletrodos. Para cada par conectado, foi feita uma varredura completa, e checado se a

corrente circulante estava condizente com o valor esperado. Por fim, foi observado o correto funcionamento de todos os eletrodos em relação à corrente, validando o bloco de injeção de corrente.

4.1.2 Teste dos amplificadores de tensão

Graças à construção modular da placa, foi possível testar separadamente o bloco de amplificação de tensão. Este bloco é constituído de dois amplificadores de tensão em série com ganho total teórico de 5000, e o componente utilizado foi o INA188, do fabricante Texas Instruments.

Para validar o bloco, foi utilizado um divisor resistivo em conjunto com uma fonte de tensão para obter 30uV de tensão de entrada. A entrada negativa foi conectada à terra, enquanto que a entrada positiva estava ligada aos 30uV. Na saída do amplificador de instrumentação, foi observado 180mV, valor 20% acima do esperado teoricamente. Esta discrepância se deveu à queda de tensão entre o terminal e o ponto de terra que foi conectado ao amplificador, além da falta de precisão do instrumento de bancada utilizado para a medição dos 30uV.

4.1.3 Teste do sistema integrado – Protótipo

Uma vez funcionando todos os módulos críticos separadamente, os testes passaram a contemplar a integração de todo o sistema. Dessa forma foram feitos os testes automatizados via MATLAB.

A ideia destes testes não era apenas validar o protótipo, mas também testar alguns dos limites do dispositivo, através da injeção de diferentes correntes DC, alterando a pressão sobre os contatos jacaré, alterando as placas de teste, alterando o tempo de multiplexação dos eletrodos, etc.

4.1.3.1 Resultados

Os resultados dos testes foram os seguintes:

1. *Teste da influência do tempo de multiplexação dos eletrodos. ($I=625\text{mA}$)*



Figura 43 – Placa Original

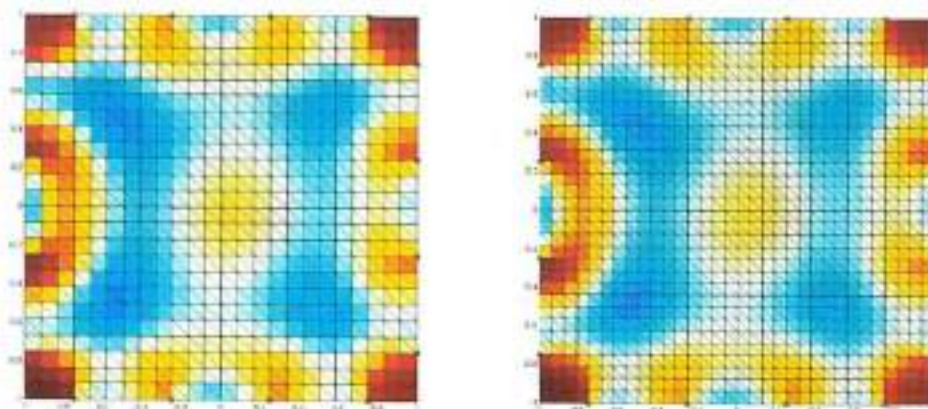


Figura 44 - Tempo de multiplexação de medições de tensão e corrente = 1ms.

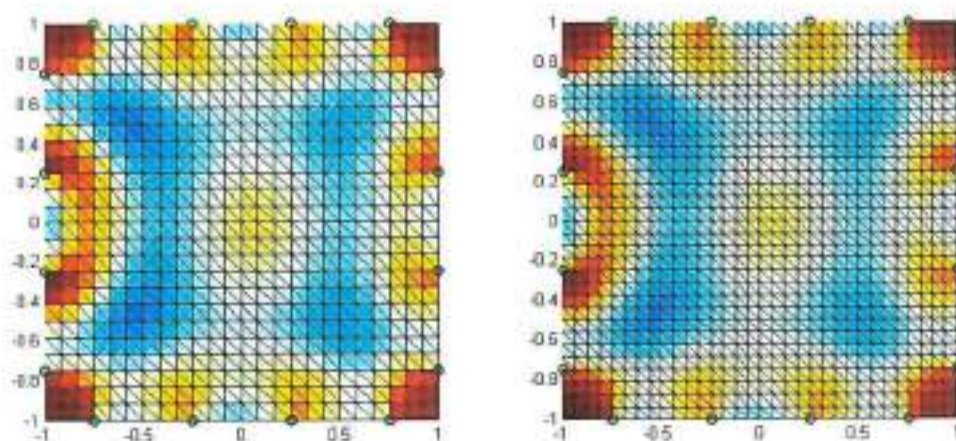


Figura 45 - Tempo de multiplexação de medições de tensão e corrente = 2,5ms.

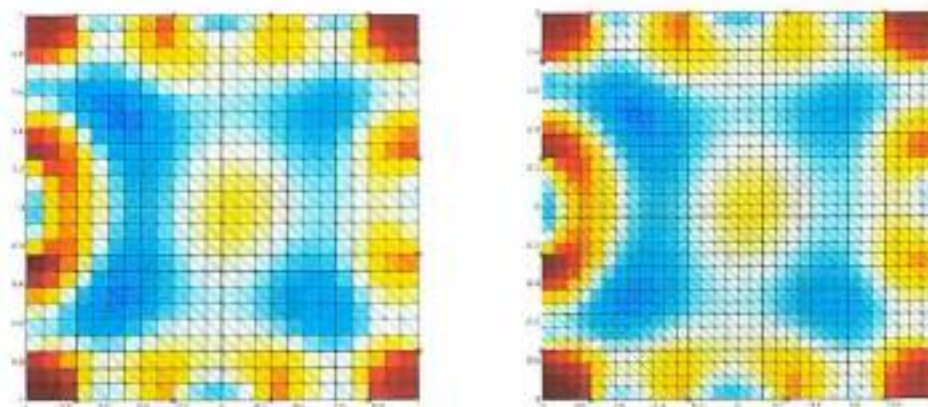


Figura 46 - Tempo de multiplexação de medições de tensão e corrente = 100ms.

2. *Teste da influência da corrente injetada entre os eletrodos. (Tempo de multiplexação de corrente e medições de tensões = 1ms)*



Figura 47 - Placa Original

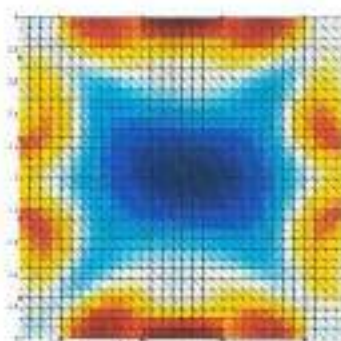


Figura 48 - Corrente = 1,25A

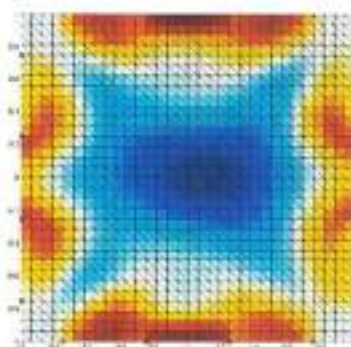


Figura 49 - Corrente = 0,625A

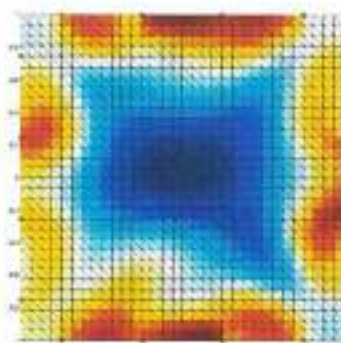


Figura 50 - Corrente = 0,415A

3. *Teste da influência da resistência de contato sobre a qualidade da imagem.*
(Corrente = 0,625A, tempo de multiplexação = 1 ms)

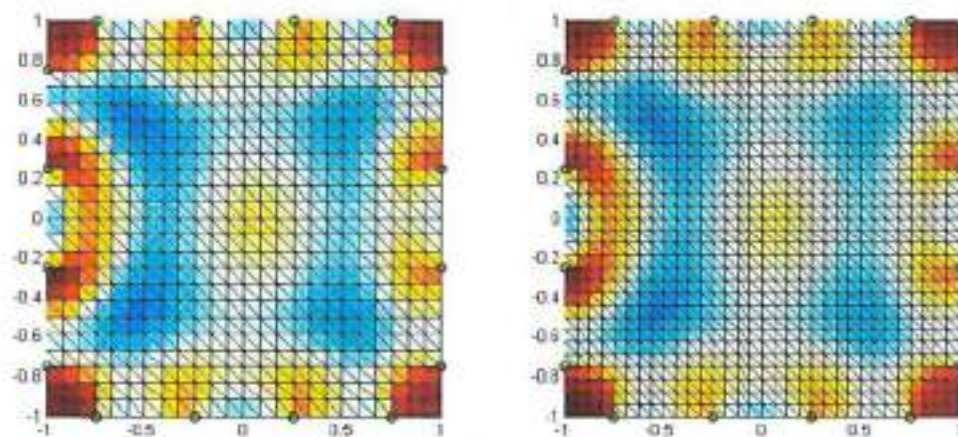


Figura 51 - Condição: Eletrodos jacaré fazendo contato com a pressão própria das molas

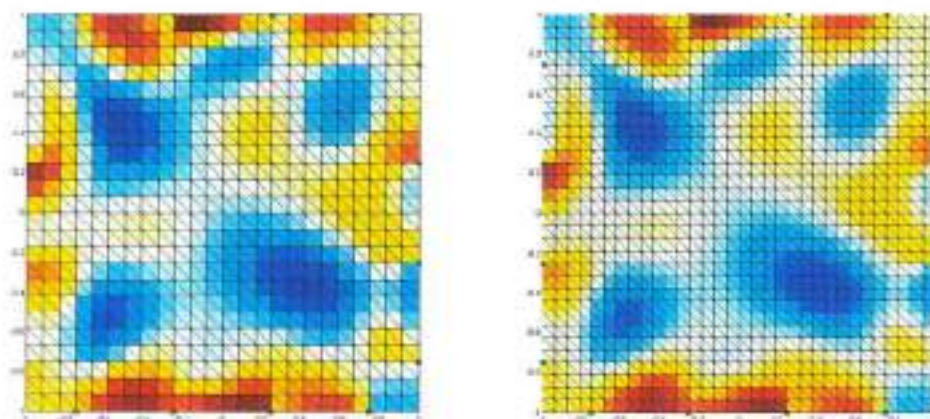


Figura 52 - Condição: 4 dos 16 eletrodos sendo pressionados manualmente.

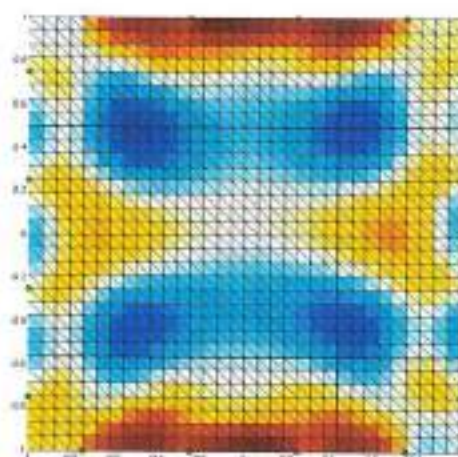


Figura 53 - Condição: Todos eletrodos com um reforço de fio de cobre em volta da mandíbula do jacaré.

4. *Teste qualitativo – placa com furo central e rasgo tangencial ($I = 1,25A$)*

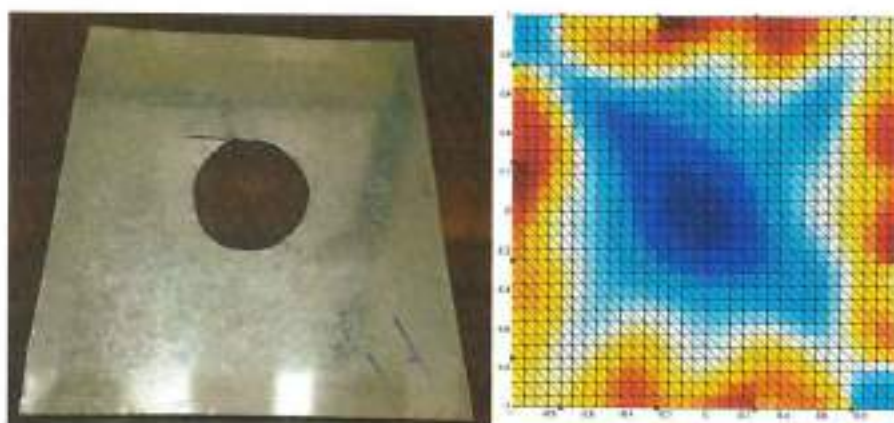


Figura 54 - Comparativo entre placa com furo central e rasgo tangencial com imagem gerada

5. *Teste qualitativo – placa com rasgos dispostos assimetricamente ($I = 1,25A$)*

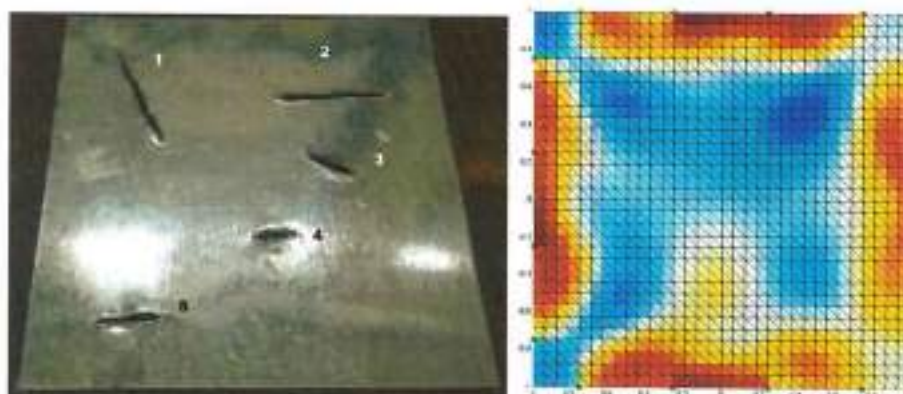


Figura 55 - Comparativo entre placa com rasgos assimétricos com imagem gerada

6. Teste qualitativo – placa com três rasgos ($I = 1,25A$)

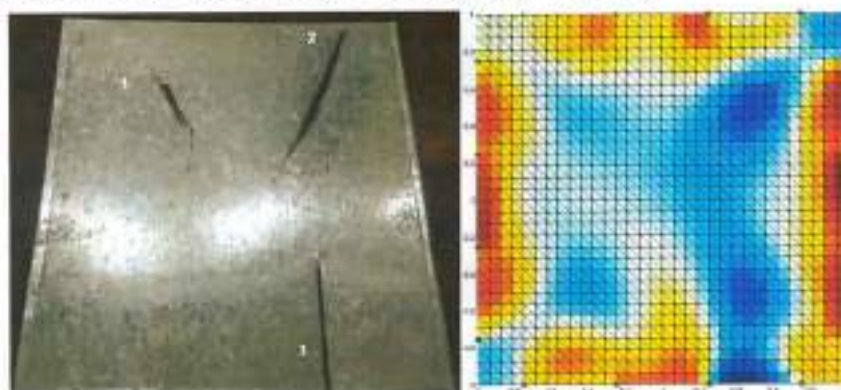


Figura 56 - Comparativo entre placa com três rasgos com imagem gerada

7. Teste qualitativo – placa com quatro furos de diversos diâmetros ($I=1,25A$)

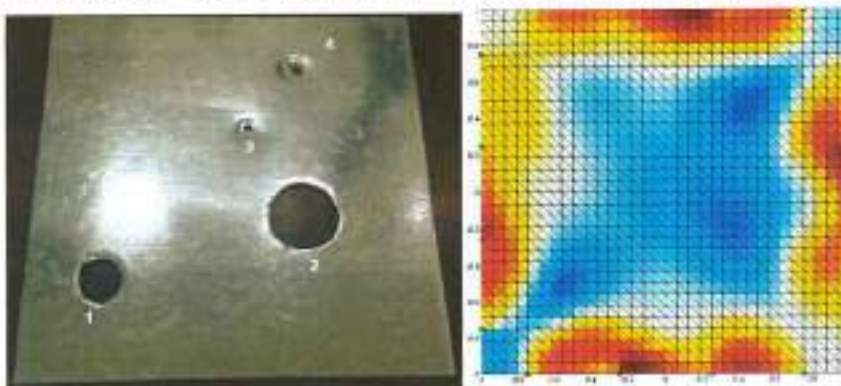


Figura 57 - Comparativo entre placa com quatro furos com imagem gerada

8. Teste qualitativo – placa com furos retangulares dispostos simetricamente ($I=1,25A$)

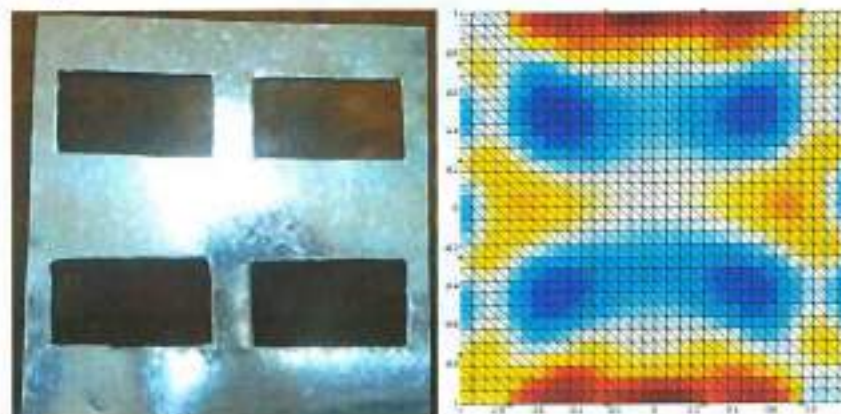


Figura 58 - Comparativo entre placa com furos retangulares simétricos com imagem gerada

9. Teste qualitativo – placa com furo retangular central ($I=1,25A$)

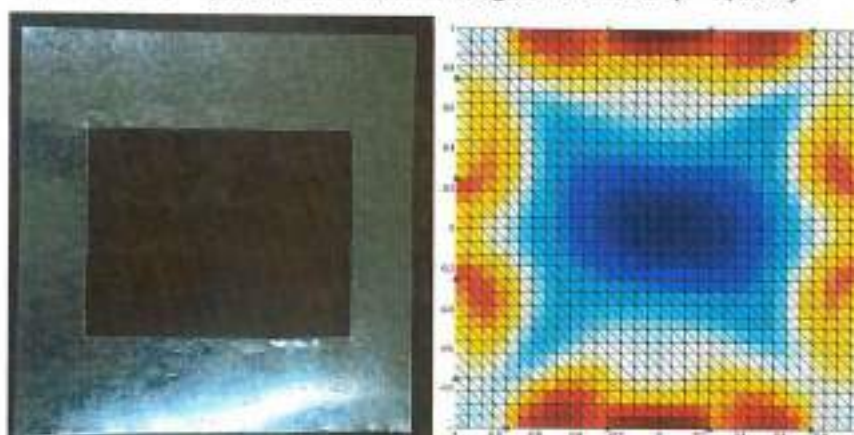


Figura 59 - Comparativo entre placa com furo retangular central com imagem gerada

10. Teste qualitativo – placa lisa, sem furos ($I=1,25A$)

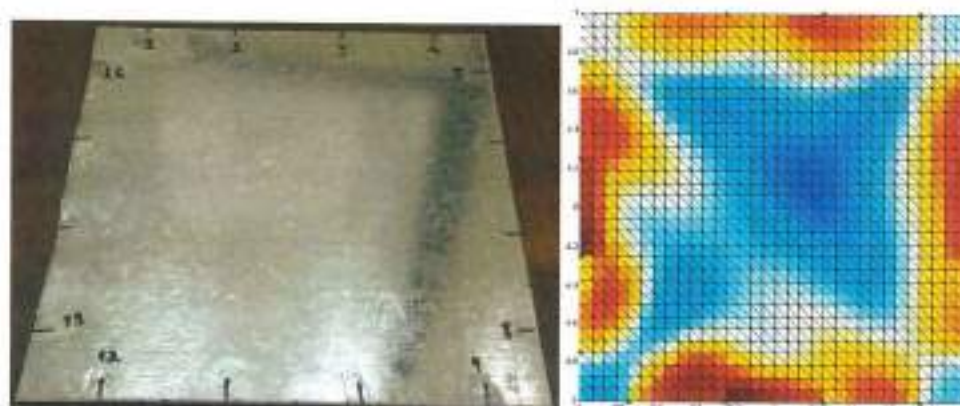


Figura 60 - Comparativo entre placa sem furos com imagem gerada

4.1.3.2 Especificações técnicas

As principais conclusões a que se chegou após os testes de caracterização do dispositivo encontram-se na tabela a seguir. O teste do qual foram obtidos tais resultados foi realizado com uma placa de aço zincado quadrada, com dimensões 20 cm X 20 cm, corrente de 1A e 16 eletrodos.

A menor fissura que foi detectada (marcada com o número 5, na Figura 55) ocupa 2% da área total da placa que foi avaliada; a duração total do teste foi de 11 segundos; a resistência mínima que pode ser colocada entre os terminais da fonte de corrente e ainda gerar um sinal detectável pelo voltímetro do microcontrolador é de $1\ \mu\Omega$; e, por fim, a máxima resistência que pode ser colocada entre os terminais da fonte de corrente é de $3,75\ \Omega$ (dada a condição de operação (i) da fonte de corrente especificada na Seção 3.2.2.1.1).

Placa de aço zincado de 20 cm X 20 cm, 1A, 16 eletrodos	
Área relativa da menor fissura detectada (% em relação à área total avaliada)	2%
Tempo médio de duração do teste	8 s
Resistência mínima entre os terminais da fonte de corrente	1 $\mu\Omega$
Resistência máxima entre os terminais da fonte de corrente	3,75 Ω

Tabela 11 – Principais especificações técnicas do dispositivo desenvolvido.

4.1.3.3 Análises

Os testes trouxeram resultados interessantes em relação ao protótipo e aos caminhos em que devem seguir as suas melhorias.

Corrente injetada:

Foi analisada a influência do valor da corrente utilizada na varredura. Na teoria, como as placas ao injetar corrente DC se comportam como resistências ôhmicas, quanto maior a corrente injetada, maior seria a diferença de potencial entre os eletrodos, facilitando a aquisição pelo microcontrolador e consequentemente o processamento pelo EIDORS. Foram feitas varreduras com correntes de 0,415A, 0,815A e 1,25A. Notou-se que a imagem gerada pelo EIDORS é mais fidedigna quanto maior a corrente, o que condiz com a previsão teórica.

Tempo de multiplexação de corrente e medição de tensões:

Para tempos superiores a 1 ms, não foi possível observar alterações significativas entre as imagens, o que também nos indica que podemos caminhar para um limite

inferior de tempo, visando minimizar o tempo de execução. Vale ressaltar que esse ponto não é crítico, uma vez que todo processo está sendo executado em tempos inferiores a 3 s, o que atende muito bem os requisitos do projeto.

Resistência de contato

Esse teste trouxe resultados altamente relevantes e também previstos anteriormente, desde a prova de conceito, o de que as resistências de contato têm forte influência na geração da imagem.

Uma vez que o algoritmo trabalha com as proporcionalidades entre as tensões medidas nos eletrodos para gerar a imagem, a resistência dos fios é facilmente contornada via software pois é uma resistência fixa invariante no tempo. Porém, as resistências de contato são variantes no tempo e da mesma ordem de magnitude (ou superior) da resistência do metal. Desta forma os testes mostraram que pressionando mais do que apenas as pressões das molas dos jacarês, é possível obter contornos muito mais contrastantes, imagens de maior qualidade. Um método eficiente para resolver essa questão foi enrolar fios de cobre em volta da peça metálica de cima do jacaré, de forma que maior área de contato e pressão fossem exercidas pelo jacaré do eletrodo.

Amostragem do Conversor Analógico Digital

Algumas otimizações foram necessárias no firmware do microcontrolador. Primeiramente, a conversão analógica-digital estava programada para definir como valor correto da conversão o valor médio de 8 amostras. Notou-se que entre realizações de varredura havia mudança significativa do padrão de imagem. Para contornar isso, foi o valor médio passou a ser feito entre 512 amostras, no decorrer de um segundo.

A análise qualitativa foi feita entre sete placas de teste, com diversas topologias de furos e rasgos. Abaixo segue análise das placas.

- Placa com furo central e rasgo tangencial (Figura 54)

Nota-se que o sistema foi capaz de identificar o furo e o rasgo. É possível identificar a baixa condutividade representada pelo tom azul escuro na diagonal superior como sendo influenciada pelo rasgo tangencial.

- Placa com rasgos dispostos assimetricamente (Figura 55)

A imagem gerada pelo sistema identificou uma área de menor condutividade no canto superior direito, correspondendo aos rasgos 2 e 3. Também há indicação de existência do rasgo 5. O rasgo 4, de menor comprimento, não foi identificado pelo sistema.

- Placa com três rasgos, sendo um na borda (Figura 56)

Na imagem gerada pelo sistema da placa da figura 53, observa-se a existência de uma área de baixa condutividade que começa na borda e segue na vertical, correspondendo ao rasgo 3. O rasgo 2 também foi identificado na imagem gerada, porém o sistema não identificou o rasgo 1.

- Placa com quatro furos de diversos diâmetros (Figura 57)

O furo 2, por ter um diâmetro maior, correspondeu à uma área de baixa condutividade maior na imagem gerada pelo sistema, enquanto que o furo 1, de dois centímetros de diâmetro, foi identificado pela área menor no canto inferior esquerdo. Não é possível determinar com precisão a localização dos furos 3 e 4 na imagem gerada, entretanto o conjunto dos dois pontos resultou em uma área de baixa condutividade.

- Placa com furos retangulares dispostos simetricamente (Figura 58)

Nesta placa, observou a identificação quatro áreas de baixa de condutividade através da imagem gerada pelo sistema. Nota-se que essas áreas geradas não correspondem à forma geométrica do furo real.

- Placa com furo retangular central (Figura 59)

É possível observar semelhança intensa entre a placa original e a imagem gerada. A forma retangular também foi preservada na imagem gerada pelo sistema.

Comparativamente com outras placas e resultados, esta imagem gerada foi a mais fidedigna com a placa real, considerando a forma e a área de baixa condutividade.

- Placa com lisa, sem furos (Figura 60)

Por fim, foi feita uma análise qualitativa da imagem gerada em uma placa sem furos, totalmente contínua. O resultado previsto seria a de uma imagem sem áreas de baixa condutância, o que na realidade não ocorreu. Entre diversas realizações de varredura, notou-se que sempre há áreas de baixa condutividade na imagem gerada. Entretanto, nota-se que a intensidade destas áreas é menor do que as áreas de baixa condutância vistas em placas com furos reais.

Os resultados analisados permitem avaliar o sistema como um todo. O dispositivo é eficaz em indicar fissuras e furos de tamanho maior que dois centímetros. Para agrupamentos de fissuras menores, conforme visto na Figura 57 e sua análise, o sistema identifica a região inteira como sendo de baixa condutância, não sendo possível precisar o tamanho das fissuras menores que 2 cm quando próximas entre si.

Adicionalmente, a forma geométrica dos rasgos dificilmente é preservada. Por exemplo, um rasgo horizontal pode ser representado por uma pequena região quadrada de baixa condutância, enquanto que a placa com furos retangulares teve suas fissuras representadas por círculos. Entretanto, vale ressaltar que o objetivo de verificar a existência ou não de fissuras ainda é atingido.

Um teste que trouxe resultados interessantes foi o da análise da placa sem fissuras (Figura 60). Um falso positivo neste projeto é definido como uma área de baixa condutância indicada pela imagem gerada pelo sistema em uma região sem furos ou fissuras. O cenário ideal para a análise da placa sem fissuras seria uma imagem sem falsos positivos, o que não ocorreu nos testes. Testes seguidos geraram falsos positivos consistentemente, porém de baixa condutividade maior comparada com as imagens geradas a partir de placas com furos. Na Figura 60 é possível observar que a área possui um tom mais claro do que as áreas das outras placas com fissuras, o

que abre espaço para futuras melhorias, na parte de tratamento de imagens, para identificação de falsos positivos.

4.1.3.4 Consideração sobre a implementação em smartphones

Um dos objetivos deste trabalho é criar uma ferramenta que pudesse ser facilmente integrada a smartphones. Dessa forma toda a arquitetura do sistema considerou essa possibilidade, mas o maior desafio estava centrado no problema do hardware e processamento dos sinais.

Durante o design do sistema de processamento em nuvem, buscou-se uma arquitetura que atendesse aos diversos requisitos de engenharia e marketing propostos no início do projeto. Não houve, porém, a implementação deste pois, mais uma vez, não era o foco do projeto, mas sim resolver os vários desafios de um hardware que lida com sinais da ordem de unidades de microvolts.

Vale lembrar que o trabalho até este momento criou um protótipo funcional que gera imagens através de EIT, de forma automática em desktops. O protótipo ainda permite uma série de evoluções em diversas áreas, entre as quais se enquadra a implementação da arquitetura de processamento em nuvem descrita na Seção 3.2.4.

5. CONCLUSÕES

Sob a ótica de um projeto de conclusão de curso de Engenharia Elétrica, a realização deste trabalho foi bastante satisfatória, uma vez que ele é bastante abrangente e exigiu competências em diferentes áreas técnicas. Foram também necessárias habilidades com questões de empreendedorismo na concepção do produto e de gerência de projeto.

A concepção do produto surgiu da visão da equipe de que a técnica de tomografia por impedância elétrica pode ser uma solução alternativa às tecnologias mais utilizadas para testes não destrutivos de estruturas metálicas. O processo de validação desta ideia baseou-se em pesquisas de estado da arte de NDT e de EIT e, principalmente, da necessidade de empresas que lidam com infraestruturas críticas de aumentar os níveis de segurança destas estruturas (p. ex. aeronaves) e tornar os processos de desenvolvimento e manutenção das máquinas mais eficientes economicamente através de novas soluções de NDT. O detector de fissuras desenvolvido gera imagens funcionais de estruturas de até 25 cm por 25 cm ou imagens locais de grandes estruturas, isto é, análise de pontos mais críticos e/ou que provavelmente apresentam fissuras.

Por ter como alvo avaliar materiais metálicos, o principal desafio do projeto de hardware é lidar com sinais da ordem de microvolts (na maioria dos casos encontrados na literatura, as aplicações são para concreto ou seres vivos, com sinais de ordens de grandeza maiores). Para atender a essa necessidade, foi preciso haver inicialmente uma pesquisa mais aprofundada de componentes. Com isso, grande parte do tempo foi investida em simulações de hardware, pois a equipe tinha consciência de que erros no projeto de hardware são bastante custosos em tempo e dinheiro.

Foram identificados alguns pontos no protótipo desenvolvido que podem ser aperfeiçoados em uma segunda iteração. Entre eles, a utilização de uma fonte de corrente de maior capacidade e o uso de um estágio de ganho programável para a medição de tensão permitiriam o uso do protótipo em uma gama mais ampla de materiais, pois a escala da tensão medida está relacionada com o material utilizado. O uso de algoritmos de tratamento de imagem permitiria uma melhor visualização das

fissuras e a identificação de falsos positivos. Uma modificação substancial no projeto seria utilizar uma fonte de corrente alternada em que fosse possível varrer uma escala de frequência. Isto simplificaria os circuitos de medição de tensão, já que a utilização de corrente alternada na frequência de ressonância resultaria em uma magnitude maior na tensão a ser amplificada, reduzindo a complexidade dos amplificadores de instrumentação.

Uma das diretrizes do trabalho foi a de elaborar o produto como uma versão inicial, adotando boas práticas de software e hardware que facilitassem sua manutenção, correção de erros e evolução para novas versões. Por isso a modularidade do hardware, o embasamento do software no paradigma de programação *Clean Code*, entre outras coisas. Seguir essa diretriz desde o começo conferiu robustez e maturidade ao projeto, e o mesmo apresenta potencial para se tornar ainda mais portátil e funcional, isto é, ainda mais adequado para testes de estruturas *in loco*.

6. BIBLIOGRAFIA

1. SPECKMANN, H. **Structural Health Monitoring**. Disponível em: <<http://easa.europa.eu/system/files/dfu/IMRBPB%20IMRBPB%20April%2008%20presentation.pdf>>. Acesso em: 08 de maio de 2015.
2. WEI, H.; SOLEIMANI, M. **Electromagnetic Tomography for Medical and Industrial Applications: Challenges and Opportunities [Point of View]**. Proceedings of the IEEE, vol.101, no.3. Março de 2013. Disponível em: <<http://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=6461963&isnumber=6461951>>. Acesso em: 08 de maio de 2015.
3. DENG, Y.; LIU, X. **Electromagnetic Imaging Methods for Nondestructive Evaluation Applications**. Sensors. Basel, Suíça. Disponível em: <<http://www.mdpi.com/1424-8220/11/12/11774/htm>>. Acesso em: 12 de maio de 2015.
4. TIMPEL. Disponível em: <<http://www.timpel.com.br/>>. Acesso em: 02 de dezembro de 2015.
5. PAPER, C.; UNIVERSIT, G. B.; UNIVERSIT, F. F. **Toolkit for wearable electrical impedance tomography (TWEIT): a software library for real- time image reconstruction on mobile devices** . n. JUNE 2012, 2015.
6. ABOU-ZEID, M.; FOWLER, D. W.; NAWY, E. G.; ALLEN, J. H. **Control of Cracking in Concrete Structures**. ACI 224R-01, Reported by ACI Committee 224, (October 2001), 1–8. <http://doi.org/0097-8515>
7. HOLDER, D. S. **Electrical Impedance Tomography: History, Theory and Applications**. IOP Publishing Ltd., Londres, 2005.
8. SHUNG, K. K.; SMITH, M. B.; TSUI, B. **Principles of medical imaging**. Taylor & Francis, Nova York, 2008.
9. HARIKUMAR, R.; PRABU, R.; RAGHAVAN, S. **Electrical Impedance Tomography (EIT) and Its Medical Applications: A Review**. International Journal of Soft Computing and Engineering (IJSCE), Volume-3, Issue-4, Setembro de 2013.
10. BERA, T. K. **Electrical Impedance Tomography**. Disponível em: <<http://web.yonsei.ac.kr/seoj/eit.htm>>. Acesso em: 30 de abril de 2015.
11. ADLER, A.; LIONHEART, W. R. B. **Uses and abuses of EIDORS: An extensible software base for EIT**. Physiol. Meas. 27:S25-S42, 2006.
12. VAUHKONEN, P. J.; VAUHKONEN, M.; SAVOLAINEN, T.; KAPIO, J. P. (1999). **Three-dimensional electrical impedance tomography based on the complete**

electrode model. *Biomedical Engineering*, IEEE Transactions on, 46(9), 1150–1160.

13. POLYDORIDES, N.; LIONHEART, W. R. B. **A Matlab toolkit for three-dimensional electrical impedance tomography: a contribution to the Electrical Impedance and Diffuse Optical Reconstruction Software project**, *Meas. Sci. Technol.* 13 (December 2002) 1871–1883.

14. CONCRETE FOUNDATIONS ASSOCIATION OF NORTH AMERICA. **Concrete Cracking**. Disponível em: <<http://www.cfawalls.org/foundations/cracking.htm>>. Acesso em: 08 de maio de 2015.

15. HEWLETT-PACKARD DEVELOPMENT COMPANY. **USB 3.0 Technology: Performance Advantage on HP Workstations**. Julho de 2012. Disponível em: <<http://www8.hp.com/h20195/v2/GetDocument.aspx?docname=4AA4-2724ENW>>. Acesso em: 14 de maio de 2015.

16. IEEE STANDARDS ASSOCIATION. **Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications – IEEE Std. 802.11 - 2012**. The Institute of Electrical and Electronics Engineers, Inc. Publicado nos Estados Unidos da América em 29 de março de 2012.

17. BLUETOOTH SIG INC. **A Look at the Basics of Bluetooth Technology**. Disponível em: <<http://www.bluetooth.com/Pages/basics.aspx>>. Acesso em: 14 de maio de 2015.

18. FAIRCHILD SEMICONDUCTOR. **Application Note: Basic Analog and Digital Multiplexer Design Comparison**. Copyright 2002 Fairchild Semiconductor Corporation. Disponível em: <application-notes.digchip.com/009/9-12462.pdf>. Acesso em 18 de maio de 2015.

19. TEXAS INSTRUMENTS. **LM317 3-Terminal Adjustable Regulator**. Copyright © 1997 – 2014, Texas Instruments Incorporated.

20. ECE PROJECTS. **Sine Wave Generator**. Disponível em: <<http://electronic-projects.50webs.com/sine-wave-generator-circuit.htm>>. Acesso em: 30 de julho de 2015.

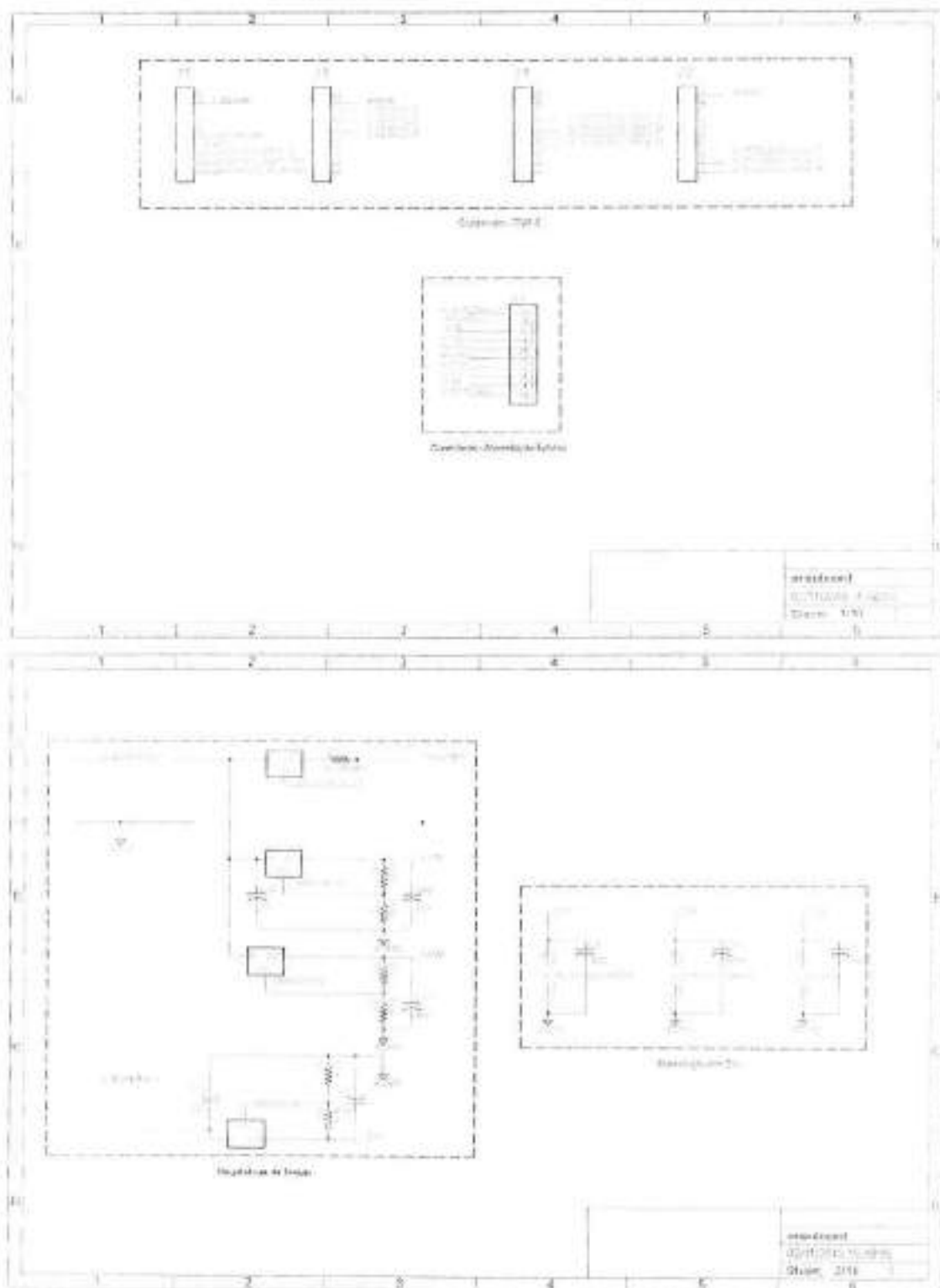
21. TEXAS INSTRUMENTS. **LMH6622 Dual Wideband, Low Noise, 160 MHz, Operational Amplifiers**. Copyright © 2001 – 2014, Texas Instruments Incorporated.

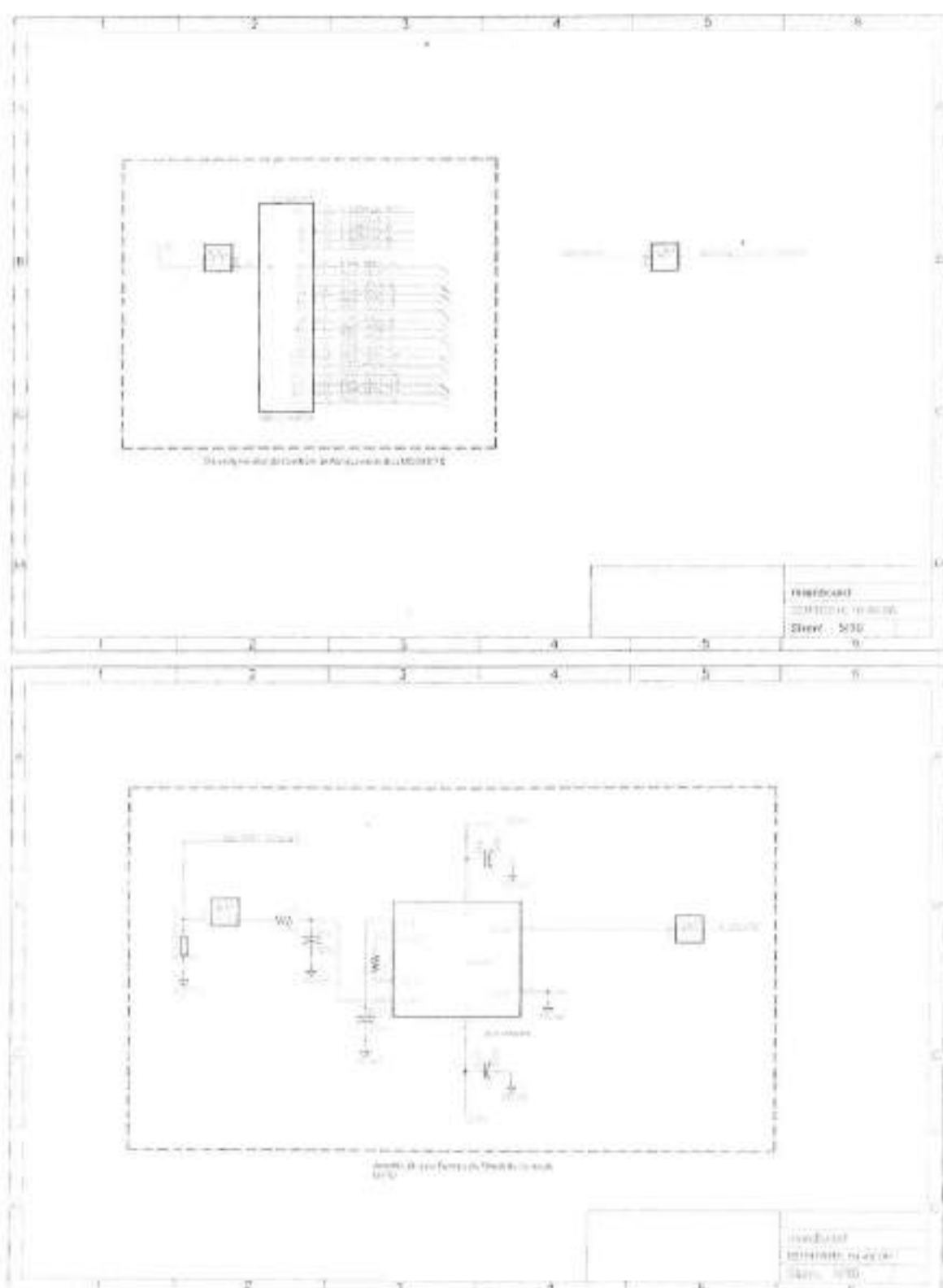
22. TEXAS INSTRUMENTS. **High-Speed CMOS Logic 16-Channel Analog Multiplexer/Demultiplexer**. Copyright © 2003, Texas Instruments Incorporated.

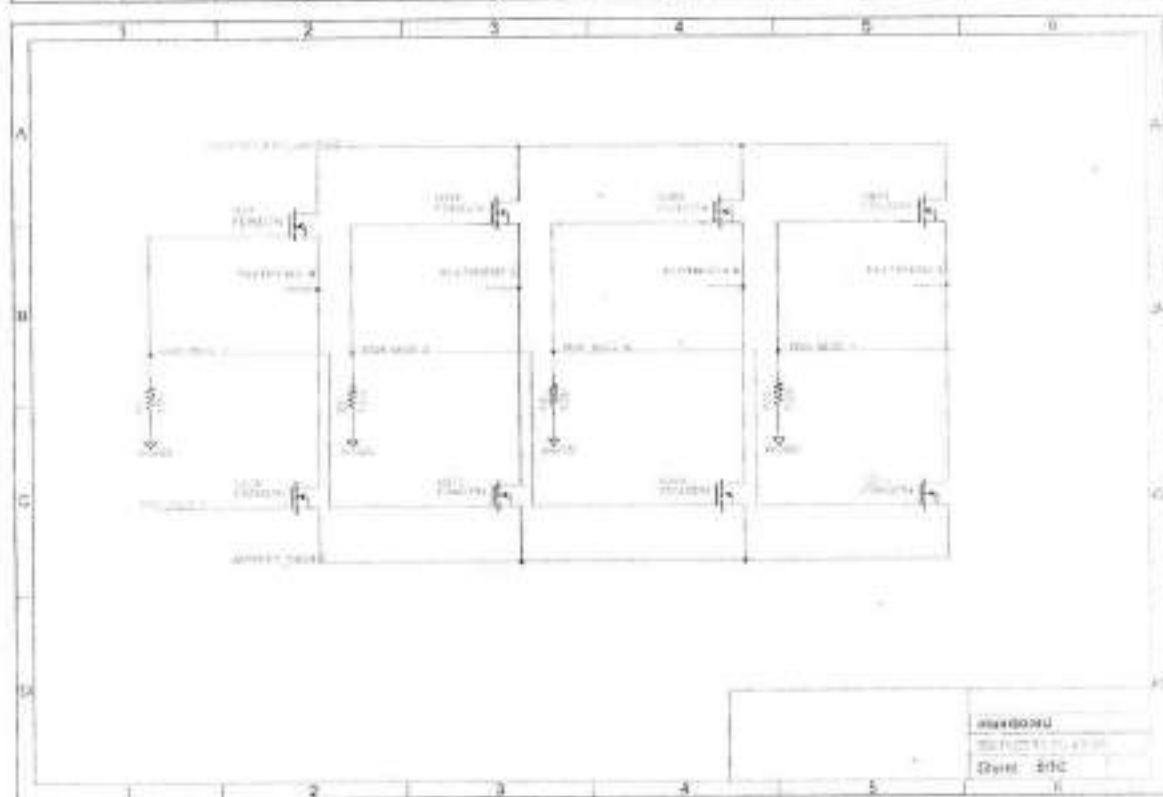
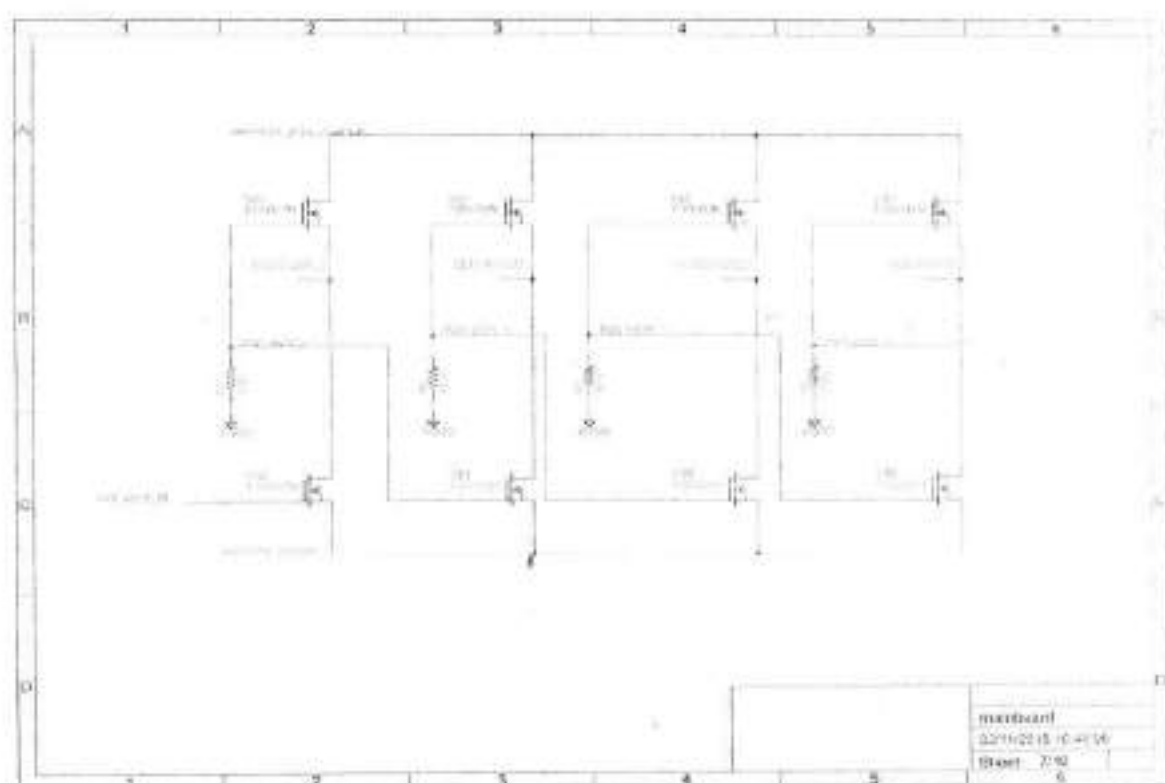
23. FAIRCHILD SEMICONDUCTOR. **FDN337 N-Channel Logic Level Enhancement Mode Field Effect Transistor**. Copyright © 1998 Fairchild Semiconductor Corporation.
24. TEXAS INSTRUMENTS. **INA188 Precision, Zero-Drift, Rail-to-Rail Out, High-Voltage Instrumentation Amplifier**. Copyright © 2015, Texas Instruments Incorporated.
25. TEXAS INSTRUMENTS. **Tiva C Series TM4C123G LaunchPad Evaluation Board – User's Guide**. Copyright © 2013, Texas Instruments Incorporated.

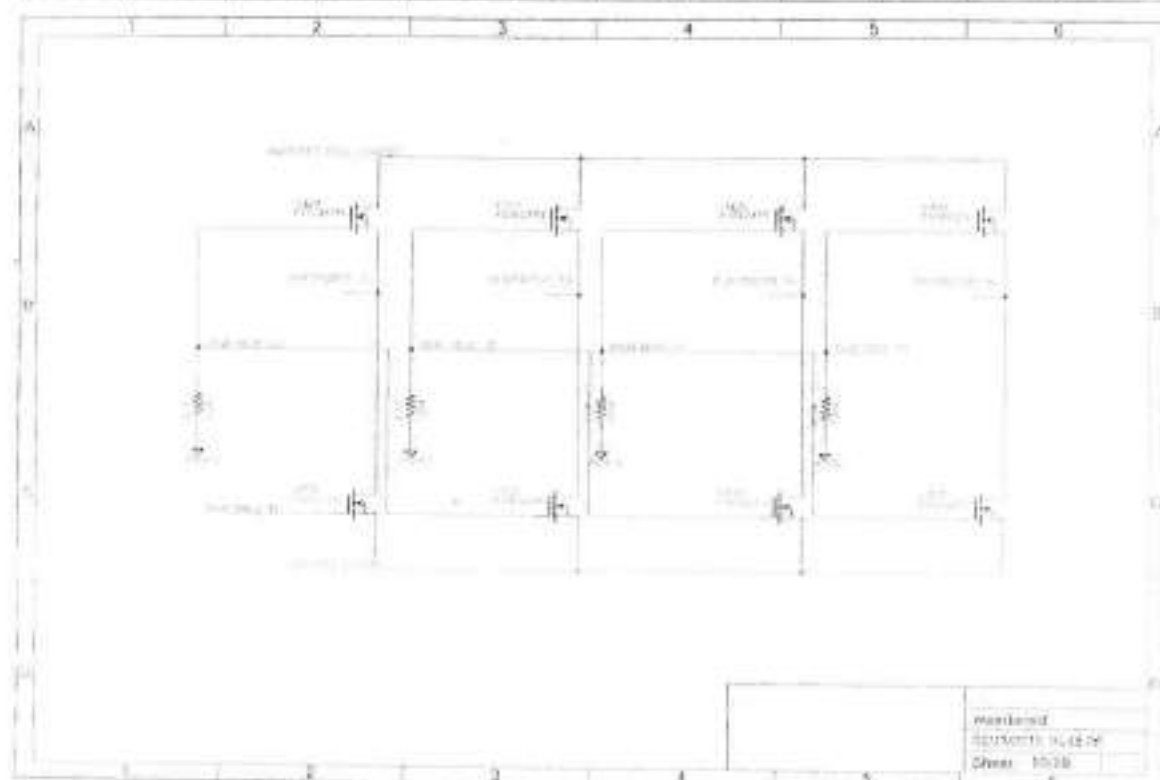
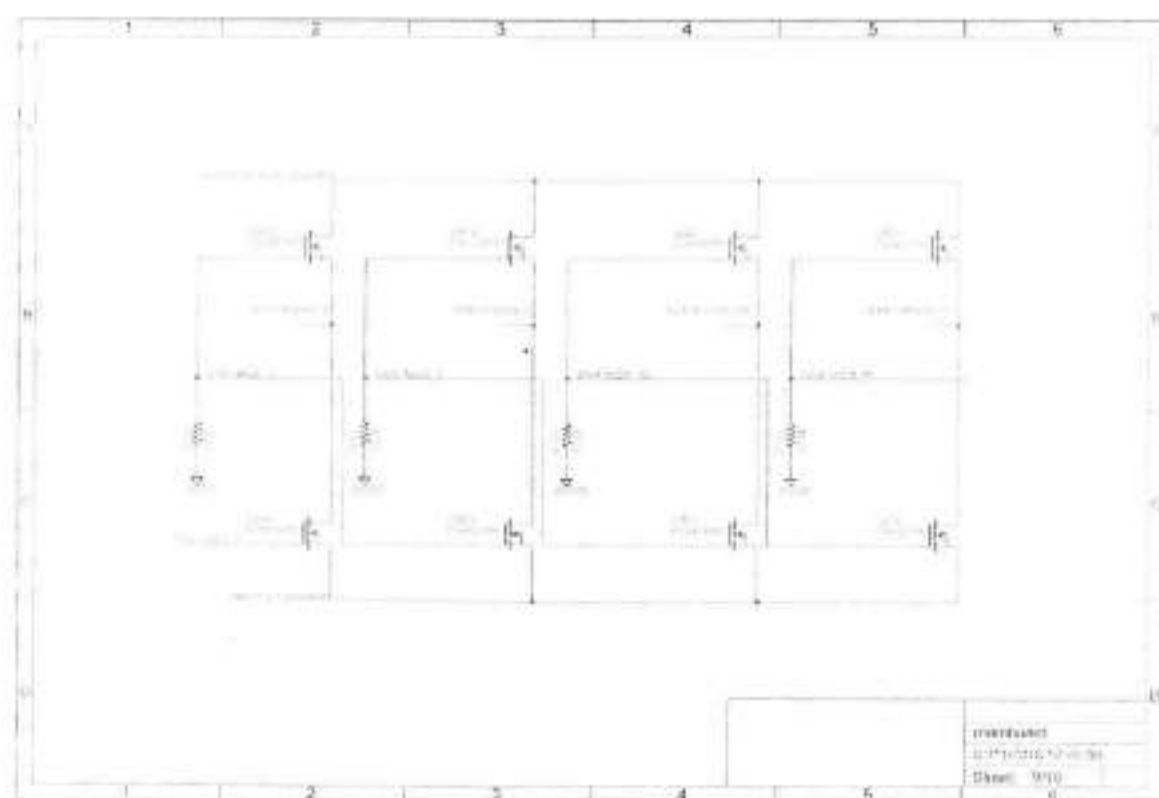
7. ANEXOS

Anexo A - Esquemático final do hardware









Anexo B – Código do firmware para Tiva-C Launchpad

main.c

```
#include "systemWrapper.h"

/*
 * This error routine that is called if the driver library encounters an error.
 */
#ifdef DEBUG
void __error__(char *pFileName, uint32_t line)
{
}
#endif

int main(void) {
    ConfigureSistema();
    ConfigureUARTViaPCComBaudRateDe115200(); //Inicializa UART
    //ConfigureUARTViaBaudRateComBaudRateDe9600();
    ConfigureADCDaAmplificadoraDeInstrumentacaoDeTensao();
    ConfigureADCDaAmplificadoraDeInstrumentacaoDeCorrente();
    ConfigureTimerInterruptoUmSegundo(); //Inicializa Timer e Interrupções
    ConfigurePinosMuxTensao(); //Inicializa Pinos e Pinos do Mux de Tensão
    ConfigurePinosMuxCorrente(); //Inicializa Pinos e Pinos do Mux de Corrente
    ConfigureWatchdog();

    HabilitaPinos();
    desabilitaMuxCorrente();

    while (1) {
        //DesligaMuxCorrenteDeterminadoPerio();

        if (getComandoRecebidoDoPC() == COMANDO_INICIAR) {
            IniciaColetaRecebidoDoPC();
            IniciaVarreduraDeEletrodos();
        } else {
            //RelaxaMux();
        }
    }
}
```

systemWrapper.h

```
#include "Drivers/ADC.h"
#include "Drivers/SYS.h"
#include "Drivers/UART.h"
#include "Drivers/Timer.h"
#include "Drivers/Watchdog.h"

#include "Tasks/Interruptos.h"
#include "Tasks/IniciaVarredura.h"
#include "Tasks/DebugBluetooth.h"
#include "Tasks/DebugMux.h"
```

Drivers/ADC.h e ADC.c

```
/*
 * ADC.h
 *
 * Created on: 04/09/2015
 * Author: Rodrigo
 */

#include <stdint.h>
#include <stdio.h>
#include <stdbool.h>
#include <string.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_types.h"
#include "driverlib/adc.h"
#include "driverlib/debug.h"
#include "driverlib/gpio.h"
#include "driverlib/i2c.h"
#include "driverlib/irq.h"
#include "driverlib/rtc.h"
#include "driverlib/spi.h"
#include "driverlib/timer.h"
#include "driverlib/uart.h"
#include "driverlib/usb.h"
#include "utils/uartstdio.h"

#ifndef ADC_HEADER
#define ADC_HEADER

void ConfigureADCDaAmplificadoraDeInstrumentacaoDeTensao();
void ConfigureADCDaAmplificadoraDeInstrumentacaoDeCorrente();
```

```

void adquirirTensaoDoAmplificadorDeInstrumentacao(uint32_t dados)
void adquirirCorrenteDoAmplificadorDeInstrumentacao(uint32_t dados);

//-----
// ADC.c
//
// Created on: 04/08/2015
// Author: Rodrigo
//
#include "ADC.h"

void ConfigurarADCDoAmplificadorDeInstrumentacaoDeTensao() {
    //
    // The ADC0 peripheral must be enabled for use.
    //
    SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC0);

    //SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    //SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);

    //GPIOPinTypeADC(GPIO_PORTA_BASE, GPIO_PIN_3);
    //GPIOPinTypeADC(GPIO_PORTB_BASE, GPIO_PIN_4); //PB4

    //
    // Enable sample sequence 3 with a processor signal trigger. Sequence 3
    // will do a single sample when the processor sends a signal to start the
    // conversion. Each ADC module has 4 programmable sequences, sequence 0
    // to sequence 3. This example is arbitrarily using sequence 3.
    //
    ADCSequenceConfigure(ADC0_BASE, 3, ADC_TRIGGER_PROCESSOR, 0);

    //
    // Configure step 0 on sequence 3. Sample channel 0 (ADC_CTL_CH0) in
    // single-ended mode (default) and configure the interrupt flag
    // (ADC_CTL_IE) to be set when the sample is done. Tell the ADC logic
    // that this is the last conversion on sequence 3 (ADC_CTL_END). Sequence
    // 3 has only one programmable step. Sequence 1 and 2 have 4 steps, and
    // sequence 0 has 8 programmable steps. Since we are only doing a single
    // conversion using sequence 3 we will only configure step 0. For more
    // information on the ADC sequences and steps, reference the datasheet.
    //
    ADCSequenceStepConfigure(ADC0_BASE, 3, 0,
        ADC_CTL_CH0 | ADC_CTL_IE | ADC_CTL_END);
    ADCHardwareOversampleConfigure(ADC0_BASE, 64);

    //ADCSequenceStepConfigure(ADC0_BASE, 0, 7, ADC_CTL_CH0 | ADC_CTL_IE | ADC_CTL_END);

    //
    // Since sample sequence 3 is now configured, it must be enabled.
    //
    ADCSequenceEnable(ADC0_BASE, 3);

    //
    // Clear the interrupt status flag. This is done to make sure the
    // interrupt flag is cleared before we sample.
    //
    ADCIntClear(ADC0_BASE, 3);
}

void adquirirTensaoDoAmplificadorDeInstrumentacaoDeCorrente() {
    //GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2);
    //Trigger acquisition on ADC
    ADCProcessorTrigger(ADC0_BASE, 3);
    while (ADCIntStatus(ADC0_BASE, 3, false)) {
    }
    ADCIntClear(ADC0_BASE, 3);
    GPIOPinWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0);
    //When done, do ADC
    ADCSequenceDataGet(ADC0_BASE, 3, dados);
}

void ConfigurarADCDoAmplificadorDeInstrumentacaoDeCorrente() {
    //
    // The ADC0 peripheral must be enabled for use.
    //
    SysCtlPeripheralEnable(SYSCTL_PERIPH_ADC1);

    //SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    //SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);

    //GPIOPinTypeADC(GPIO_PORTA_BASE, GPIO_PIN_3);
    //GPIOPinTypeADC(GPIO_PORTB_BASE, GPIO_PIN_6); //PB6

    //
    // Enable sample sequence 3 with a processor signal trigger. Sequence 3
    // will do a single sample when the processor sends a signal to start the
    // conversion. Each ADC module has 4 programmable sequences, sequence 0
    // to sequence 3. This example is arbitrarily using sequence 3.
    //
    ADCSequenceConfigure(ADC1_BASE, 3, ADC_TRIGGER_PROCESSOR, 0);

    //
    // Configure step 0 on sequence 3. Sample channel 0 (ADC_CTL_CH0) in
    // single-ended mode (default) and configure the interrupt flag
    // (ADC_CTL_IE) to be set when the sample is done. Tell the ADC logic
    // that this is the last conversion on sequence 3 (ADC_CTL_END). Sequence

```



```

//DEFINIÇÕES DE PINOS
#define I_POSITIVO_SA_PIN          GPIO_PIN_0 /PA2
#define I_POSITIVO_SB_PIN          GPIO_PIN_1 /PA3
#define I_POSITIVO_SC_PIN          GPIO_PIN_2 /PA4
#define I_POSITIVO_SD_PIN          GPIO_PIN_3 /PA5
#define I_POSITIVO_CHIP_SELECT_1_PIN  GPIO_PIN_4 /PA6

//Definições do Mux de Corrente Negativa
#define I_NEGATIVO_SA_PORT          GPIO_PORTD_BASE
#define I_NEGATIVO_SB_PORT          GPIO_PORTD_BASE
#define I_NEGATIVO_SC_PORT          GPIO_PORTD_BASE
#define I_NEGATIVO_SD_PORT          GPIO_PORTD_BASE
#define I_NEGATIVO_CHIP_SELECT_1_PORT  GPIO_PORTD_BASE

//DEFINIÇÕES DE PINOS
#define I_NEGATIVO_SA_PIN          GPIO_PIN_0 /PA2
#define I_NEGATIVO_SB_PIN          GPIO_PIN_1 /PA3
#define I_NEGATIVO_SC_PIN          GPIO_PIN_2 /PA4
#define I_NEGATIVO_SD_PIN          GPIO_PIN_3 /PA5
#define I_NEGATIVO_CHIP_SELECT_1_PIN  GPIO_PIN_5 /PA6

void inicializaGPIO(void);

void ligarLEDVermelho(void);
void ligarLEDAzul(void);
void ligarLEDVerde(void);

void desligarLEDVermelho(void);
void desligarLEDAzul(void);
void desligarLEDVerde(void);

void ConfigurarPinosMuxTensao();
void SelecionarMuxTensaoPositiva(uint8_t numeroMuxTensao);
void SelecionarMuxTensaoNegativa(uint8_t numeroMuxTensao);
void HabilitarMuxTensaoPositiva(uint8_t numeroEntrada);
void HabilitarMuxTensaoNegativa(uint8_t numeroEntrada);

void ConfigurarPinosMuxCorrente();
void SelecionarMuxCorrentePositiva(uint8_t numeroMuxCorrente);
void SelecionarMuxCorrenteNegativa(uint8_t numeroMuxCorrente);
void HabilitarMuxCorrentePositiva(uint8_t numeroSaida);
void HabilitarMuxCorrenteNegativa(uint8_t numeroSaida);

/*
 * GPIO.c
 *
 * Created on: 04/09/2015
 * Author: Rodrigo
 */
#include "GPIO.h"

void inicializaGPIO(void) {
    RDM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    RDM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    RDM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
    RDM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    RDM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    RDM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOF);
}

void ligarLEDVermelho(void) {
    GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_1, GPIO_PIN_1);
}

void desligarLEDVermelho() {
    GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_1, 0);
}

void ligarLEDAzul(void) {
    GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_2, GPIO_PIN_2);
}

void desligarLEDAzul() {
    GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_2, 0);
}

void ligarLEDVerde(void) {
    GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_3, GPIO_PIN_3);
}

void desligarLEDVerde(void) {
    GPIOWrite(GPIO_PORTF_BASE, GPIO_PIN_3, 0);
}

```

Drivers/MUX.h e MUX.c

```

#include <stdint.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "mux.h"

```



```

void ConfigurarPinsDoCanalA() {
    // Habilita pines dos muxes de corrente
    ROM_GPIOWrite(GPIOOutputL_PAN_SA_PORT,
        (PAR_SA_PIN | PAR_SB_PIN | L_PAN_SC_PIN | L_PAN_SD_PIN |
        ROM_GPIOInput(GPIOInputL_PAN_CHP_SELECT_1_PORT,
        L_PAN_CHP_SELECT_1_PIN);
}

void HabilitaEntradaMuxTensaoPositiva(uint8_t numeroEntrada) {
    // numeroEntrada entre 0 e 15
    // Habilita o mux de tensao positiva
    numeroEntrada = numeroEntrada - 1;

    switch (numeroEntrada) {
        case 0:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN);
            break;
        case 1:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SA_PIN);
            break;
        case 2:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SB_PIN);
            break;
        case 3:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN);
            break;
        case 4:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SC_PIN);
            break;
        case 5:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SC_PIN | V_POSITIVO_SA_PIN);
            break;
        case 6:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SC_PIN | V_POSITIVO_SB_PIN);
            break;
        case 7:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SC_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SA_PIN);
            break;
        case 8:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN);
            break;
        case 9:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SA_PIN);
            break;
        case 10:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SB_PIN);
            break;
        case 11:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN);
            break;
        case 12:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SC_PIN);
            break;
        case 13:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SA_PIN);
            break;
        case 14:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SB_PIN);
            break;
        case 15:
            GPIOWrite(V_POSITIVO_SA_PORT, V_POSITIVO_SA_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SD_PIN |
            V_POSITIVO_SD_PIN | V_POSITIVO_SC_PIN | V_POSITIVO_SB_PIN | V_POSITIVO_SA_PIN);
            break;
        default:
            break;
    }

    GPIOWrite(V_POSITIVO_CHP_SELECT_1_PORT, V_POSITIVO_CHP_SELECT_1_PIN, 0);
}

void HabilitaEntradaMuxTensaoNegativa(uint8_t numeroEntrada) {
    // numeroEntrada entre 0 e 15
    // Habilita o mux de tensao negativa
    numeroEntrada = numeroEntrada - 1;

    switch (numeroEntrada) {
        case 0:
            GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
            break;
        case 1:

```

```

V_NEGATIVO_SA_PIN);
break;
case 2:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SB_PIN);
break;
case 3:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN);
break;
case 4:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SC_PIN);
break;
case 5:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SC_PIN | V_NEGATIVO_SA_PIN);
break;
case 6:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SC_PIN | V_NEGATIVO_SB_PIN);
break;
case 7:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SC_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SA_PIN);
break;
case 8:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | 0);
break;
case 9:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SA_PIN);
break;
case 10:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SB_PIN);
break;
case 11:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN);
break;
case 12:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SC_PIN);
break;
case 13:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SA_PIN);
break;
case 14:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SB_PIN);
break;
case 15:
    GPIOWrite(V_NEGATIVO_SA_PORT, V_NEGATIVO_SA_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SD_PIN);
V_NEGATIVO_SD_PIN | V_NEGATIVO_SC_PIN | V_NEGATIVO_SB_PIN | V_NEGATIVO_SA_PIN);
break;
default:
    break;
}

GPIOWrite(V_NEGATIVO_CHIP_SELECT_1_PORT, V_NEGATIVO_CHIP_SELECT_1_PIN, 0);
}

void HandleParMuxComodo(juint8_t numeroSede) {
    //NumeroEntrada entre 0 e 15
    //Chaveira o max de corrente
    numeroSede = numeroSede - 1;

    switch (numeroSede) {
    case 0:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, 0);
        break;
    case 1:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SA_PIN);
        break;
    case 2:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SB_PIN);
        break;
    case 3:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SA_PIN | L_PAR_SB_PIN);
        break;
    case 4:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SB_PIN | L_PAR_SC_PIN);
        break;
    case 5:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SD_PIN | L_PAR_SC_PIN | L_PAR_SA_PIN);
        break;
    case 6:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SD_PIN | L_PAR_SC_PIN | L_PAR_SB_PIN);
        break;
    case 7:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SD_PIN | L_PAR_SC_PIN | L_PAR_SB_PIN | L_PAR_SA_PIN);
        break;
    case 8:
        GPIOWrite(L_PAR_SA_PORT, L_PAR_SA_PIN | L_PAR_SB_PIN | L_PAR_SC_PIN | L_PAR_SD_PIN, L_PAR_SD_PIN | L_PAR_SC_PIN | L_PAR_SB_PIN | L_PAR_SA_PIN);
        break;
    }
}

```



```

void ConfigureUARTBluetoothComBaudRateDe115200(void)
{
    void printViaUARTViaPC(const char* format, ...);
    char receiveCharViaPC();
    char receiveCharSerialBlueserViaPC();
    void sendCharViaPC(char c);
    void sendCharViaBluetooth(char c);
    void sendStringConcat2DeArgumentViaPC(const char* str, int2_t arg);
    uint8_t contentCharNoBufferViaPC();
}

/*
 * UART.c
 *
 * Created on: 14/03/2015
 * Author: Rodrigo
 */
#include "UART.h"

void ConfigureUARTViaBluetoothComBaudRateDe9600(void)
{
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);

    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART1);
    ROM_GPIOPinConfigure(GPIO_PA0, U1RX);
    ROM_GPIOPinConfigure(GPIO_PA1, U1TX);
    ROM_GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);
    UARTClockSourceSet(UART1_BASE, UART_CLOCK_PIOSC);

    //
    // Initialize the UART for console IO.
    //
    UARTStdioConfig(1, 9600, 1600000);
}

void ConfigureUARTViaPCComBaudRateDe115200(void)
{
    //
    // Enable the GPIO Peripheral used by the UART.
    //
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOA);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOB);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOC);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOE);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_GPIOD);

    //
    // Enable UART0
    //
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_UART0);

    //
    // Configure GPIO Pins for UART mode.
    //
    ROM_GPIOPinConfigure(GPIO_PA0, U0RX);
    ROM_GPIOPinConfigure(GPIO_PA1, U0TX);
    ROM_GPIOPinTypeUART(GPIO_PORTA_BASE, GPIO_PIN_0 | GPIO_PIN_1);

    //
    // Use the internal 16MHz oscillator as the UART clock source.
    //
    UARTClockSourceSet(UART0_BASE, UART_CLOCK_PIOSC);

    //
    // Initialize the UART for console IO.
    //
    UARTStdioConfig(0, 115200, 16000000);
    sendCharViaPC('x');
    sendCharViaPC('y');
}

char receiveCharViaPC()
{
    return UARTCharGetNonBlocking(UART0_BASE);
}

char receiveCharSerialBlueserViaPC()
{
    return UARTGet();
}

void sendCharViaPC(char c)
{
    return UARTCharPutNonBlocking(UART0_BASE, c);
}

void sendCharViaBluetooth(char c)
{
    UARTCharPut(UART1_BASE, c);
}

uint8_t contentCharNoBufferViaPC()
{
    return UARTCharsAvail(UART0_BASE);
}

```

```
void enviaStringConstr(32DeArgumentosViaPC)(const char *str, uint32_t arg)
{
    UARTprintf(str, arg);
}
```

Drivers/Watchdog.h e Watchdog.c

```
/*
 * Watchdog.h
 *
 * Created on: 15/10/2015
 * Author: Rodrigo
 */
#include <stdint.h>
#include <stdbool.h>
#include <stdio.h>
#include <string.h>
#include "inc/hw_ints.h"
#include "inc/hw_memmap.h"
#include "inc/hw_sysctl.h"
#include "driverlib/adc.h"
#include "driverlib/comp.h"
#include "driverlib/crc.h"
#include "driverlib/dsp.h"
#include "driverlib/gpio.h"
#include "driverlib/i2c.h"
#include "driverlib/uart.h"
#include "driverlib/watchdog.h"
#include "utils/utils.h"

#define DRIVERS_WATCHDOG_H_
#define DRIVERS_WATCHDOG_H_

#define WATCHDOG_SEGUNDOS 15

void ConfigureWatchdog();
void LinguaInterupcaoDoWatchdog();
uint8_t CheckStatusDoWatchdog();
void DesabilitaWatchdog();
void HabilitaWatchdog();

#ifndef DRIVERS_WATCHDOG_H_
#define DRIVERS_WATCHDOG_H_

/*
 * Watchdog.c
 *
 * Created on: 15/10/2015
 * Author: Rodrigo
 */

#include "Watchdog.h"

volatile uint8_t watchDogStatus = 1;

void ConfigureWatchdog()
{
    ROM_SysCtlPeripheralEnable(SYSCTL_PERIPH_WDOG0);
    ROM_IntEnable(INT_WATCHDOG);
    ROM_WatchdogReloadSet(WATCHDOG0_BASE, SysCtlClockGet() * WATCHDOG_SEGUNDOS);
    ROM_WatchdogResetEnable(WATCHDOG0_BASE);
    ROM_WatchdogEnable(WATCHDOG0_BASE);
}

void LinguaInterupcaoDoWatchdog()
{
    ROM_WatchdogIntClear(WATCHDOG0_BASE);
}

uint8_t CheckStatusDoWatchdog()
{
    return watchDogStatus;
}

void DesabilitaWatchdog()
{
    watchDogStatus = 0;
}

void HabilitaWatchdog()
{
    watchDogStatus = 1;
}
```

Tools/ProtocoloTransmissao.h e ProtocoloTransmissao.c

```
/*
 * ProtocoloTransmissao.h
 *
 * Created on: 06/09/2015
```

```

/* Author: Rodrigo
*/

#include MESSAGE_E_OPCODES_STRUCT
#include MESSAGE_E_OPCODES_STRUCT
#include "UART.h"
#include "DriverSYS.h"

#define HEADER_STRING "-----Sistema de Detecção de Fissura v1.0-----\n"
/* Desenvolvido por:
 *      Walter Moraes
 *      Pedro Henrique Amorim
 *      Rodrigo Midea Coelho
 *      Brasil, São Paulo - TOC, 2015
 */
/* Entre com a opção desejada:\n
 * 1) Iniciar Teste
 */

typedef struct {
    uint32_t idData;
    uint32_t qntdo;
    uint32_t elevezaTensaoNeg;
    uint32_t elevezaTensaoPos;
    uint32_t elevezaCorrenteNeg;
    uint32_t elevezaCorrentePos;
} Message;

typedef enum {
    ERRO = 10,
    COMANDO_PARAR = 12,
    COMANDO_INICIAR = 11,
    HEADER = 10
} Opcodes;

uint8_t recebeComandoViaPC() {
    int8_t recebeStringViaPC(char* msg, int comprimento);
    int8_t enviaStringViaPC(const char* msg);
    uint8_t enviaStringBletooth(const char* msg);
    Opcodes analisaComandoViaPC(char* msg);
    uint32_t calculaCRC(Message msg);
    uint32_t calculaCRCBuffer(uint32_t *buffer, uint32_t size);
    uint32_t transmiteBuffer(uint32_t *buffer, uint32_t size);
    void serializarFimDaTransmissao();
}

/*
 * ProtocoloTransmissao.h
 * Created on: 04/09/2015
 * Author: Rodrigo
 */

#define "ProtocoloTransmissao.h"

uint8_t recebeComandoViaPC() {
    unsigned char c;
    if (UARTGetChar() != 0) {
        c = recebeCharViaPC();
        return (uint8_t)c;
    }
}

Opcodes analisaComandoViaPC(char* msg) {
    int8_t status = 0;

    if (strcmp(msg, "1") == 0) {
        status = enviaStringViaPC("OK_INICIAR");
        return COMANDO_INICIAR;
    }
    else {
        status = enviaStringViaPC(HEADER_STRING);
        return ERRO;
    }
}

int8_t recebeStringViaPC(char* msg, int comprimento) {
    uint8_t contadorDeChars = 0;
    while (contemCharNoBufferViaPC()) {
        msg[contadorDeChars] = recebeComandoViaPC();
        contadorDeChars++;
        if (contadorDeChars == comprimento)

```

```

        return -1;
    }
    msg(contadorDeChars) = 0;
    return 0;
}

int8_t enviaStringViaPC(const char* msg){
    while(msg != 0){
        enviaCharViaPC(*msg);
        msg++;
    }
    enviaCharViaPC('\n');
    enviaCharViaPC('\n');
    return 0;
}

int8_t enviaStringBluetooth(const char* msg){
    while(msg != 0){
        enviaCharViaBluetooth(*msg);
        msg++;
    }
    enviaCharViaBluetooth('\n');
    enviaCharViaBluetooth('\n');
    return 0;
}

uint32_t calculaCRC(Mensagem msg){
    uint32_t LRC = 0;
    LRC ^= (msg.adData);
    LRC ^= msg.gerenciador;
    LRC ^= msg.estadoTensaoNeg;
    LRC ^= msg.estadoTensaoPos;
    LRC ^= msg.estadoCorrenteNeg;
    LRC ^= msg.estadoCorrentePos;
    return LRC;
}

uint32_t calculaCRCBuffer(uint32_t *buffer, uint32_t size){
    uint32_t LRC = 0;
    int i;
    for (i = 0; i < size; i++)
    {
        LRC ^= buffer[i];
    }
    return LRC;
}

uint32_t tamanhoBuffer(uint32_t *buffer, uint32_t size){
    uint32_t LRC;
    char c;
    int i;
    LRC = calculaCRC(buffer, size);
    enviaStringConteudoDeArgumentoViaPC("%4d", size * 2);
    for (i = 0; i < size; i++)
    {
        enviaStringConteudoDeArgumentoViaPC("%4d", buffer[i]);
    }
    enviaStringConteudoDeArgumentoViaPC("%4d", LRC); // Nesta função adiciono um \r sempre antes de um \n, logo não há necessidade de inserir o CR
    c = recebeCharSerialViaPC();
    if (c == '\n')
        return 1;
    else
        return 0;
}

void sinalizaFimDoTransmissao(){
    UART printf("SUCESSO\n");
}

```

Tools/LED.h e LED.c

```

/*
 * LED.h
 *
 * Created on: 04/09/2015
 * Author: Rodrigo
 */

#include "Arduino.h"

void toggleLEDVerde();
void toggleLEDAzul();
void toggleLEDVerde();

void desligaLEDAzulImediatamente();

/*
 * LED.c
 *
 * Created on: 04/09/2015
 * Author: Rodrigo
 */

#include "LED.h"

```

```

static uint8_t statusLEDVerde = 0x00;
static uint8_t statusLEDAzul = 0x00;
static uint8_t statusLEDVermelho = 0x00;

void toggleLEDVermelho() {
    //static uint8_t statusLEDVermelho = 0x01;
    if(statusLEDVermelho) {
        desligLEDVermelho();
        statusLEDVermelho = 0x00;
    } else {
        ligLEDVermelho();
        statusLEDVermelho = 0x01;
    }
}

void toggleLEDAzul() {
    //static uint8_t statusLEDAzul = 0x01;
    if(statusLEDAzul) {
        desligLEDAzul();
        statusLEDAzul = 0x00;
    } else {
        ligLEDAzul();
        statusLEDAzul = 0x01;
    }
}

void desligLEDAzulImpendeStatus() {
    desligLEDAzul();
    statusLEDAzul = 0x00;
}

void toggleLEDVerde() {
    if(statusLEDVerde) {
        desligLEDVerde();
        statusLEDVerde = 0x00;
    } else {
        ligLEDVerde();
        statusLEDVerde = 0x01;
    }
}

```

Tasks/Interrupcoes.h e Interrupcoes.c

```

/*
 * Interrupcoes.h
 *
 * Created on: 14/08/2015
 * Author: Rodrigo
 */

#ifndef INTERRUPTOES_H
#define INTERRUPTOES_H

#include "../Tools/LED.h"
#include "../Tools/ProtocoloTransmissao.h"
#include "../Drivers/UART.h"
#include "../Drivers/Watching.h"

void nomeACadaUmSegundo();
void nomeACadaUmSegundo_String();
void WatchdogHabilitar();

Opcoes getComandoRecbidoViaPC();
void ImporComandoRecbidoViaPC();

#endif

/*
 * Interrupcoes.c
 *
 * Created on: 07/08/2015
 * Author: Rodrigo
 */

#include "Interrupcoes.h"

#define Opcoes comandoRecbido;

void nomeACadaUmSegundo_String() {
    int i;
    char msg[30];
    RDM_TimerHClear(TIMER0_BASE, TIMER_TIMA_TIMEOUT);
    toggleLEDAzul();
    if (contemCharNoBufferViaPC()) {
        desligLEDAzul();
        toggleLEDVerde();
        resposta = recebeStringViaPC(msg, 30);
    }
}

```

```

        comandoRecebido = analisaComandoViaPC(msg);
        //resposta = enviaString(msg);
        //toggleLEDVerde();
    } else
    {
        void WatchdogHandler()
        {
            /*Se o Watchdog estiver desativado, retorno do handler. Quando entrar novamente nesta função o sistema será reiniciado.*/
        }
        if (ChecarStatusDoWatchdog() == 0)
        {
            return;
        }
        limpaInterupcaoDoWatchdog();
        //toggleLEDVerde();
    }

    Opcoes getComandoRecebidoViaPC()
    {
        return comandoRecebido;
    }

    void limpaComandoRecebidoViaPC()
    {
        comandoRecebido = 0;
    }
}

```

Tasks/IniciaVarredura.h e IniciaVarredura.c

```

/*
 * IniciaVarredura.h
 *
 * Created on: 07/09/2015
 * Author: Rodrigo
 */

#ifndef INICIAVARREDURAHEADER
#define INICIAVARREDURAHEADER

#include "../Tools/ProtocoloTransmissao.h"
#include "../Drivers/MUX.h"
#include "../Drivers/ADC.h"
#include "../Drivers/Watchdog.h"

#define NUM_MEDIDAS 16
#define TXBUFFER_SIZE 252
#define DELAY_VARREDURA_EM_MILISEGUNDOS 1
#define DELAY_ENTRADA_ELETRODOS 10
#define CLOCK_UTILIZADO 20000000
#define SAMPLES_TO_AVERAGE 128

void iniciaVarreduraDeEntodos(void);

#endif

typedef struct
{
    uint32_t voltage;
    uint32_t current;
} Medicao;

/*
 * IniciaVarredura.c
 *
 * Created on: 07/09/2015
 * Author: Rodrigo
 */

#include "IniciaVarredura.h"

void iniciaVarreduraDeEletrodos(void) {
    uint32_t i;
    uint32_t adcTensao[];
    uint32_t adcCorrente[];
    uint32_t eleTensaoNeg, eleTensaoPos;
    uint32_t eleCorrenteNeg, eleCorrentePos;
    uint32_t numMedidasTensao, numMedidasCorrente;
    uint32_t numDeDadosAdquiridos;
    uint32_t pBuffer[TXBUFFER_SIZE];
    uint32_t pBufferSize;
    uint32_t iAcc;
    uint32_t sampleNumber;
    Medicao medicao;

    eleTensaoNeg = 3;
    eleTensaoPos = 4;
    eleCorrenteNeg = 2;
    eleCorrentePos = 1;
    numMedidasTensao = 0;
    numMedidasCorrente = 0;
    pBufferSize = 0;

    numDeDadosAdquiridos = 0;

    eleCorrenteNeg = 2;
    eleCorrentePos = eleCorrenteNeg + 1;
    numMedidasCorrente = 0;
}

```

```

while (numMedidasCorrente < NUM_MEDIDAS){
    numMedidasTensao = 0;
    elctTensaoNeg = 1;
    elctTensaoPos = elctTensaoNeg + 1;
    UARTprintf("ADC[Ganho][ParCorrente][ParTensao][NumAmostra]n");
    bufferSize = 0;
    HabilitaParMuxCorrente(elctCorrentePos); // Na realidade habilita o par N e N+1, com o polo positivo em N e o negativo em N+1
    while (numMedidasTensao < NUM_MEDIDAS) {
        if (elctTensaoNeg == NUM_MEDIDAS) {
            elctTensaoPos = 1;
        }
        if (elctCorrenteNeg == NUM_MEDIDAS + 1) {
            elctCorrentePos = 1;
        }
        if (elctTensaoNeg == elctCorrenteNeg)
            elctTensaoNeg == elctCorrentePos;
        else if (elctTensaoPos == elctCorrentePos)
            elctTensaoPos == elctCorrenteNeg;
    } while {
        //HabilitaParMuxCorrente(elctCorrentePos); // Na realidade habilita o par N e N+1, com o polo positivo em N e o
        SysCtlDelay(SysCtlClockGet()/1000*DELAY_VARREDURA_EM_MILISEGUNDOS);

        HabilitaEntradaMuxTensaoPositiva(elctTensaoPos);
        HabilitaEntradaMuxTensaoNegativa(elctTensaoNeg);
        SysCtlDelay(SysCtlClockGet()/1000*DELAY_VARREDURA_EM_MILISEGUNDOS);

        medicao_voltage = 0;
        medicao_current = 0;

        for (sampleNumber = 0; sampleNumber<SAMPLES_TO_AVERAGE ; sampleNumber++){
            adquiereTensaoDoAmplificadorDeInstrumentacao(adc0data);
            medicao_voltage += adc0data[0];
            adquiereCorrenteDoAmplificadorDeInstrumentacao(adc1data);
            medicao_current += adc1data[0];
            SysCtlDelay(SysCtlClockGet()/1000*DELAY_VARREDURA_EM_MILISEGUNDOS);
        }
        adc0data[0] = medicao_voltage/sampleNumber;
        adc1data[0] = medicao_current/sampleNumber;

        lbuffer[bufferSize] = adc0data[0];
        lbufferSize++;

        lbuffer[bufferSize] = 5000;
        lbufferSize++;

        lbuffer[bufferSize] = adc1data[0];
        lbufferSize++;

        lbuffer[bufferSize] = 10;
        lbufferSize++;

        lbuffer[bufferSize] = elctCorrenteNeg;
        lbufferSize++;

        lbuffer[bufferSize] = elctCorrentePos;
        lbufferSize++;

        lbuffer[bufferSize] = elctTensaoPos;
        lbufferSize++;

        lbuffer[bufferSize] = elctTensaoNeg;
        lbufferSize++;

        sumDeDadosAdquiridos++;

        lbuffer[bufferSize] = sumDeDadosAdquiridos;
        lbufferSize++;

    }
    numMedidasTensao++;
    elctTensaoNeg++;
    elctTensaoPos = elctTensaoNeg + 1;

}
desabilitaMuxCorrente();
//envia o pacote da ultima leitura e checka se a resposta do receptor foi a de recebimento
//Caso contrario renvia pacote
DesabilitaWatchdog();
while (transmissaoBuffer[buffer, lbufferSize] != 1);
//reiniciaWatchdog();
//Incrementa numero de medidas de corrente
numMedidasCorrente++;
//Incrementa eletrodos de corrente
elctCorrenteNeg++;
elctCorrentePos = elctCorrenteNeg - 1;
SysCtlDelay(SysCtlClockGet()/1000*DELAY_ENTRE_ELETRODOS);
//envia pacote sinalizando fim de transmissao
enviaZeroinDeTransmissao();

```

startup_ccs.c

```

//-----
// startup_ccs.c - Startup code for use with Tia Code Composer Studio.
//
// Copyright (c) 2012-2015 Texas Instruments Incorporated. All rights reserved.
// Software License Agreement
//
// Texas Instruments (TI) is supplying this software for use solely and
// exclusively on TI's microcontroller products. The software is owned by
// TI and/or its suppliers, and is protected under applicable copyright
// laws. You may not combine this software with "viral" open-source
// software in order to form a larger program.
//
// THIS SOFTWARE IS PROVIDED "AS IS" AND WITH ALL FAULTS.
// NO WARRANTIES, WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT
// NOT LIMITED TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
// A PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. TI SHALL NOT, UNDER ANY
// CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL, OR CONSEQUENTIAL
// DAMAGES, FOR ANY REASON WHATSOEVER.
//
// This is part of revision 2.1.1.71 of the EK-TM320GXL Firmware Package.
//-----

#include <stdint.h>
#include "inc/hw_mmc.h"
#include "inc/hw_types.h"
#include "tasks/interrupts.h"

//-----
// Forward declaration of the default fault handlers.
//
//-----
void ResetISR(void);
static void NmiISR(void);
static void FaultISR(void);
static void IntDefaultHandler(void);

//-----
// External declaration for the reset handler that is to be called when the
// processor is started
//
//-----
extern void _c_int00(void);

//-----
// Linker variable that marks the top of the stack.
//
//-----
extern uint32_t __STACK_TOP;

//-----
// The vector table. Note that the proper construct must be placed on this to
// ensure that it ends up at physical address 0x00000000 or at the start of
// the program if located at a start address other than 0.
//
//-----
#pragma DATA_SECTION(g_pfnVectors, ".intvecs")
void (* const g_pfnVectors[])(void) =
{
    (void (*)(void))((uint32_t)&__STACK_TOP), // The initial stack pointer
    ResetISR, // The reset handler
    NmiISR, // The NMI handler
    FaultISR, // The hard fault handler
    IntDefaultHandler, // The MPU fault handler
    IntDefaultHandler, // The bus fault handler
    IntDefaultHandler, // The usage fault handler
    0, // Reserved
    0, // Reserved
    0, // Reserved
    0, // Reserved
    IntDefaultHandler, // SVCall handler
    IntDefaultHandler, // Debug monitor handler
    0, // Reserved
    IntDefaultHandler, // The PendSV handler
    IntDefaultHandler, // The SysTick handler
    IntDefaultHandler, // GPIO Port A
    IntDefaultHandler, // GPIO Port B
    IntDefaultHandler, // GPIO Port C
    IntDefaultHandler, // GPIO Port D
    IntDefaultHandler, // GPIO Port E
    IntDefaultHandler, // UART0 Rx and Tx
    IntDefaultHandler, // UART1 Rx and Tx
    IntDefaultHandler, // SS0 Rx and Tx
    IntDefaultHandler, // I2C0 Master and Slave
    IntDefaultHandler, // PWM Fault
    IntDefaultHandler, // PWM Generator 0
    IntDefaultHandler, // PWM Generator 1
    IntDefaultHandler, // PWM Generator 2
    IntDefaultHandler, // Quadrature Encoder 0
}

```



```

0, // Reserved
IntDefaultHandler, // GPIO Port P (Summary or P0)
IntDefaultHandler, // GPIO Port P1
IntDefaultHandler, // GPIO Port P2
IntDefaultHandler, // GPIO Port P3
IntDefaultHandler, // GPIO Port P4
IntDefaultHandler, // GPIO Port P5
IntDefaultHandler, // GPIO Port P6
IntDefaultHandler, // GPIO Port P7
IntDefaultHandler, // GPIO Port Q (Summary or Q0)
IntDefaultHandler, // GPIO Port Q1
IntDefaultHandler, // GPIO Port Q2
IntDefaultHandler, // GPIO Port Q3
IntDefaultHandler, // GPIO Port Q4
IntDefaultHandler, // GPIO Port Q5
IntDefaultHandler, // GPIO Port Q6
IntDefaultHandler, // GPIO Port Q7
IntDefaultHandler, // GPIO Port R
IntDefaultHandler, // GPIO Port S
IntDefaultHandler, // PWM 1 Generator 0
IntDefaultHandler, // PWM 1 Generator 1
IntDefaultHandler, // PWM 1 Generator 2
IntDefaultHandler, // PWM 1 Generator 3
IntDefaultHandler, // PWM 1 Fault
};

// This is the code that gets called when the processor first starts execution
// following a reset event. Only the absolutely necessary set is performed,
// after which the application supplied entry() routine is called. Any fancy
// actions (such as making decisions based on the reset cause register, and
// reading the bits in that register) are left solely in the hands of the
// application.
//
//-----
void
ResetISR(void)
{
    //
    // Jump to the CCS C initialization routine. This will enable the
    // floating-point unit as well, so that does not need to be done here.
    //
    __asm(" _global _c_int00\n"
        " _call _c_int00");
}

// This is the code that gets called when the processor receives a NMI. This
// simply enters an infinite loop, preserving the system state for examination
// by a debugger.
//
//-----
static void
NMIISR(void)
{
    //
    // Enter an infinite loop.
    //
    while(1)
    {
    }
}

// This is the code that gets called when the processor receives a fault
// interrupt. This simply enters an infinite loop, preserving the system state
// for examination by a debugger.
//
//-----
static void
FaultISR(void)
{
    //
    // Enter an infinite loop.
    //
    while(1)
    {
    }
}

// This is the code that gets called when the processor receives an unexpected
// interrupt. This simply enters an infinite loop, preserving the system state
// for examination by a debugger.
//
//-----
static void
IntDefaultHandler(void)
{
    //
    // Go into an infinite loop.
    //
    while(1)
    {
    }
}

```

Anexo C – Scripts MATLAB do protocolo de comunicação e chamda a execução do algoritmo de reconstrução de imagem

parseFRAME.m:

```
function y = parseFrame(x)
size = x(1);
tamanhoDaRepeticao = 9;
linha = 1;
for n=2:tamanhoDaRepeticao:(size-1)
    y(linha,1) = x(n);
    y(linha,2) = x(n+1);
    y(linha,3) = x(n+2);
    y(linha,4) = x(n+3);
    y(linha,5) = x(n+4);
    y(linha,6) = x(n+5);
    y(linha,7) = x(n+6);
    y(linha,8) = x(n+7);
    y(linha,9) = x(n+8);
    linha = linha + 1;
end
end
```

calculaLRC.m:

```
% Longitudinal Redundancy Check
function y = calculaLRC(x)
LRC = 0;
size = x(1);
%disp(['LRC Recebido: ', num2str( x(size))]);

for n=2:(size-1)
    LRC = bitxor(LRC,x(n));
end
%disp(['LRC Calculado: ', num2str( LRC)]);
if x(size) == LRC
    y = 1;
else
    y = 0;
end
```

SerialPort_basic_V5.m:

```
% SerialPort_basic_V5.m
% Script
clearvars
clear all
close all
delete(instrfindall);
s = serial('COM3');
set(s, 'BaudRate', 115200, 'Terminator', 'CR/LF', 'InputBufferSize', 50000 );
fopen(s); %Abre porta serial, considerando que receberá r/n como sinal de
% fim de mensagem

%Seleciona Opcao
fprintf(s, '%s', 'q'); %envia qualquer coisa para receber o header
resposta = fscanf(s, '%s')
%disp(resposta)
opcao = input('\n ');
if opcao == 0
```

```

        disp('saíndo do script')
        break
    end
    %%Inicializa matriz de resultados
    mat = [0 0 0 0 0 0 0 0 0]

    %%Envia para o Microcontrolador a opção desejada
    switch opcao
        case 0
            disp('saíndo do script')

            %%Inicia recebimento de dados
            case 1
                fprintf(s, '%s', '\n');
                resposta = fscanf(s, '%s')
                if strcmp(resposta, 'OK_INICIAR') == 1
                    for i=1:100
                        out = fscanf(s, '%d')
                        %disp(out)
                        %%Checa se recebeu número, caso contrário checa se é string
                        %e se é SUCESSO, indicando fim da transmissão.
                        if isnumeric(out)
                            %%Este bloco de try/catch serve para analisar os frames
                            %%recebidos. Algumas vezes o microcontrolador envia
                            %%erroneamente o primeiro byte, gerando um erro ao
                            %%tentar acessar um certo elemento de out(n).
                            %%A parte de catch solicita o reenvio do frame.
                            try
                                if calculaLRC(out) == 1
                                    %necessário especificar %c para que o MATLAB
                                    %não envie \n após o 'b'
                                    fprintf(s, '%c', 'k');
                                    disp(['Frame recebido com sucesso. No:
', num2str(i)])

                                    %%Vai concatenando o resultado dos frames
                                    %Formato da matriz mat:
                                    % [Valor Tensão] [Ganho
Tensão] [Valor Corrente] [Ganho Corrente] [Eletrodo Corrente -] [Eletrodo
Corrente +] [Eletrodo Tensão -] [Eletrodo Tensão +] [NumAmostra]
                                    mat = [mat; parseFrame(out)];
                                else
                                    fprintf(s, '%c', 'n');
                                    disp(['Falha no recebimento do Frame. No:
', num2str(i)])

                                    i = i-1;
                                end
                            catch
                                fprintf(s, '%c', 'n');
                                disp(['Falha no recebimento do Frame. No:
', num2str(i)])

                                i = i-1;
                            end
                        else
                            if ischar(out)
                                if strcmp(out, 'SUCESSO', 7) == 1
                                    disp('Terminou com sucesso!');
                                else

```

```

                                disp('Nao identificou palavra');
                                end
                                break
                            else
                                disp('Recebeu mau numero');
                                break
                            end
                        end
                    end
                else
                    disp('Microcontrolador nao respondeu corretamente!')
                end

                disp('Fim')
            case 2
                fprintf(s, '%s', 'TAPAR');
            end
        fclose(s);

% Chamada à execução do EIDORS (algoritmo de reconstrução de imagem)

run startup.m;

x = mat (2:209,:);
voltageMatrix = parseEIDORS (x);

run cidrc.m;

```

Anexo D – Script MATLAB de automatização do processo

parseEIDORS.m:

% Função que adapta os frames vindos do protocolo autoral para formato
% recebido pelo EIDORS
function y = parseEIDORS(x)

```

y = zeros (16,16);
j = 1;
exceptions =    [16 1 2;
                  1 2 3;
                  2 3 4;
                  3 4 5;
                  4 5 6;
                  5 6 7;
                  6 7 8;
                  7 8 9;
                  8 9 10;
                  9 10 11;
                  10 11 12;
                  11 12 13;
                  12 13 14;
                  13 14 15;
                  14 15 16;
                  15 16 1];

for i=1:256

    % Atualiza a rodada:
    round = fix(i/16) + 1;
    eletrodo = mod (i,16);
    if ( mod(i,16) == 0 )    % ajuste da variável round na última
        round = round - 1; % medição de cada rodada
        eletrodo = 16;
    end

    % Exceções são ajustadas a cada rodada
    if ( mod(i,16) == 1 ) % verifica se a rodada mudou
        exception1 = (round-1)*16 + exceptions(round,1);
        exception2 = (round-1)*16 + exceptions(round,2);
        exception3 = (round-1)*16 + exceptions(round,3);
    end

    % Atualiza a matrix Y de voltagens
    if ( i == exception1 | i == exception2 | i == exception3 )
        y(eletrodo,round) = 0;
    else
        y(eletrodo,round) = x(j,i);
        j = j + 1;
    end
end
end

```

Anexo E – Script MATLAB do algoritmo de reconstrução de imagem por impedância elétrica – EIDORS

eidors.m:

```
% Autoria:
% Heitor de Moraes Santos - 7209885
% Pedro Henrique de Araújo Amorim - 7203648
% Rodrigo Nides Coelho - 7289526
%
% Algoritmo de reconstrução de imagem de EIT - EIDORS
% Projeto de Dispositivo de detecção de fissuras estruturais via
% impedância elétrica para smartphones
%
% São Paulo
% 08 de dezembro de 2015

%% 0) Matriz V de medições experimentais de tensões %
V = voltageMatrix;

%% 1) Geração do reticulado da placa quadrada usando o método dos
% elementos finitos %
imd1_2d2 = mk_common_model('e2s',16); % Objeto 2D quadrado, com 16
% elementos do reticulado e 16 eletrodos

%% 2) Geração da imagem e distribuição de condutividades do reticulado %
% 2a) Objeto sem perturbações
sim_img2 = mk_image(imd1_2d2);

% 2b) Resolução do problema direto
homg_data2 = fwd_solve(sim_img2);

% Transformação da matriz de tensões em um vetor, para ser compatível com o
% algoritmo
measure = reshape(V,256,1);

%% 3) Reconstrução da imagem (resolução do problema inverso)%%
rec_img2 = inv_solve(imd1_2d2, homg_data2, measure);

%% 4) Cotação da imagem e conversão em .png %%
show_fig(rec_img2);
print_convert_img.png;
```

Anexo F – Gerenciamento de projeto

O projeto em questão foi decomposto em três blocos principais: circuito condicionador de sinal, circuito conversor de dados e processamento via smartphone. Para efeitos de planejamento em um primeiro nível, foram considerados como etapa única o desenvolvimento do circuito conversor e do condicionador de sinal, pois ambos incluíam tarefas idênticas tais como prototipagem e compras, além da intenção inicial de se ter uma única placa de circuito impresso para ambos.

Etapas de Gerenciamento do Projeto

As seguintes propostas de atividades foram propostas ao término da etapa de concepção do projeto e planejamento de projeto, concluída em junho/15. Pode-se dizer que este cronograma pôde ser quase que integralmente seguido, o que indica a qualidade do processo de gerenciamento do projeto, feito pelos próprios integrantes da equipe.

Etapa 1 – Circuito de Condicionamento e Conversor de Dados

1. Sub-etapa 1 – Design dos Circuitos

Atividade: Definição elétrica do ADC

Descrição: Definir circuito elétrico do conversor analógico digital, envolvendo pesquisa de componentes que atendam as especificações do projeto.

Entregável: Esquemático do circuito

Duração: 14 dias

Pessoas: Rodrigo

Recursos: PC, Simulador SPICE

Predecessores: nenhum

Atividade: Definição elétrica do MUX

Descrição: Definir circuito elétrico do multiplexador bidirecional, envolvendo pesquisa de componentes que atendam as especificações do projeto.

Entregável: Esquemático do circuito

Duração: 14 dias

Pessoas: Heitor

Recursos: PC, Simulador SPICE

Predecessores: nenhum

Atividade: Definição elétrica do DAC

Descrição: Definir circuito elétrico do conversor digital analógico, envolvendo pesquisa de componentes que atendam as especificações do projeto.

Entregável: Esquemático do circuito

Duração: 14 dias

Pessoas: Pedro

Recursos: PC, Simulador SPICE

Predecessores: nenhum

Atividade: Simulação elétrica do DAC

Descrição: Simulação do conversor analógico digital, validando o circuito

Entregável: Formas de onda na frequência máxima de operação

Duração: 7 dias

Pessoas: Pedro

Recursos: Simulador SPICE

Predecessores: Definição elétrica do DAC

Atividade: Software do Microcontrolador

Descrição: Desenvolver o firmware do microcontrolador.

Entregável: Firmware completo

Duração: 35 dias

Pessoas: Rodrigo

Recursos: PC

Predecessores: Definição elétrica do ADC, Simulação Elétrica do DAC, Definição Elétrica do MUX.

2. Sub-etapa 2 – Desenvolvimento da Placa de Circuito Impresso

Atividade: Elaboração da Lista de Materiais

Descrição: Utilizando os esquemáticos consolidados, elaborar a lista de materiais a serem comprados.

Entregável: Lista de Materiais

Duração: 7 dias

Pessoas: Rodrigo, Heltor, Pedro

Recursos: PC

Predecessores: Definição elétrica do ADC, Simulação Elétrica do DAC, Definição Elétrica do MUX.

Atividade: Compra dos Componentes

Descrição: Efetuar a compra dos componentes utilizando o melhor fornecedor, considerando custo/tempo de entrega.

Entregável: Confirmação de Pedido de Compra

Duração: 14 dias

Pessoas: Rodrigo

Recursos: PC

Predecessores: Elaboração da Lista de Materiais.

Atividade: Elaboração do Layout

Descrição: Utilizando os esquemáticos consolidados, elaborar o layout que contenha o circuito do ADC, do multiplexador bidirecional e do DAC.

Entregável: Layout do Conversor

Duração: 14 dias

Pessoas: Rodrigo, Heltor, Pedro

Recursos: PC, Software de Layout

Predecessores: Definição elétrica do ADC, Simulação Elétrica do DAC, Definição Elétrica do MUX.

Atividade: Montagem da Placa

Descrição: Desenvolver a placa de circuito impresso utilizando o layout previamente feito.

Entregável: Placa de circuito impresso

Duração: 7 dias

Pessoas: Pedro

Recursos: PC

Predecessores: Layout do Conversor

Atividade: Soldagem

Descrição: Soldar os componentes na placa.

Entregável: Placa de circuito impresso com componentes

Duração: 7 dias

Pessoas: Rodrigo

Recursos: Estação de Solda, Esquemáticos

Predecessores: Montagem da Placa

3. Sub-etapa 3 – Testes dos Circuitos

Atividade: Testes Funcionais do MUX

Descrição: Depurar o circuito do multiplexador, validando o correto funcionamento do mesmo.

Entregável: Relatório de funcionamento do MUX.

Duração: 14 dias

Pessoas: Heitor

Recursos: Osciloscópio, gerador de funções

Predecessores: Soldagem

Atividade: Testes Funcionais do ADC+MUX

Descrição: Depurar o circuito ADC, utilizando o circuito do MUX previamente depurado.

Entregável: Relatório de funcionamento do MUX+ADC.

Duração: 14 dias

Pessoas: Rodrigo, Heitor

Recursos: Osciloscópio, gerador de funções

Predecessores: Testes Funcionais do MUX

Atividade: Testes Funcionais do DAC+MUX

Descrição: Depurar o circuito DAC, utilizando o circuito do MUX previamente depurado.

Entregável: Relatório de funcionamento do MUX+DAC.

Duração: 14 dias

Pessoas: Heitor, Pedro

Recursos: Osciloscópio, gerador de funções

Predecessores: Testes Funcionais do MUX

Atividade: Testes Funcionais do DAC+ADC+MUX

Descrição: Depurar o circuito conversor, em conjunto com o microcontrolador.

Entregável: Planilha com os dados binários de uma conversão simples

Duração: 21 dias

Pessoas: Heitor, Pedro, Rodrigo

Recursos: Osciloscópio, gerador de funções, PC

Predecessores: Testes Funcionais do MUX, Testes Funcionais do DAC+MUX, Testes Funcionais do ADC+MUX, Software do Microcontrolador.

Etapas 2 – Software Smartphone

1. Sub-etapa 1 – Validação do Algoritmo

Atividade: Implementação do Algoritmo em MATLAB

Descrição: Simular o algoritmo descrito nos artigos no MATLAB

Entregável: Representação gráfica de uma fissura

Duração: 21 dias

Pessoas: Heitor, Pedro

Recursos: PC, MATLAB

Predecessores: nenhum

2. Sub-etapa 2 – Desenvolvimento do aplicativo para Smartphone

Atividade: Obtenção de dados via USB

Descrição: Estudar e desenvolver uma solução para smartphone que utilize a comunicação USB.

Entregável: Comprovação de transferência correta de bytes via USB

Duração: 14 dias

Pessoas: Rodrigo

Recursos: PC, Smartphone

Predecessores: Nenhum

Atividade: Elaboração da Interface

Descrição: Desenvolver a interface gráfica do aplicativo.

Entregável: Interface gráfica do aplicativo

Duração: 14 dias

Pessoas: Heltor

Recursos: PC, Smartphone

Predecessores: Nenhum

Atividade: Implementação do Algoritmo

Descrição: Implementar o algoritmo validado previamente no MATLAB no aplicativo.

Entregável: Aplicativo com algoritmo implementado.

Duração: 14 dias

Pessoas: Pedro

Recursos: PC, Smartphone

Predecessores: Implementação do Algoritmo em MATLAB, Elaboração da Interface

Atividade: Implementação do Algoritmo

Descrição: Implementar o algoritmo validado previamente no MATLAB no aplicativo.

Entregável: Aplicativo com algoritmo implementado.

Duração: 14 dias

Pessoas: Pedro

Recursos: PC, Smartphone

Predecessores: Implementação do Algoritmo em MATLAB, Elaboração da Interface

3. Sub-etapa 3 – Validação do Sistema e Ajustes

Atividade: Integração com circuito conversor

Descrição: Integrar aplicativo e circuito conversor, fazendo os ajustes necessários para que atenda as especificações do projeto

Entregável: Dispositivo de detecção de fissuras em estruturas para smartphones

Duração: 56 dias

Pessoas: Rodrigo, Heitor, Pedro

Recursos: PC, Smartphone, Osciloscópio, Gerador de Funções, Estrutura com Fissura

Predecessores: Implementação do Algoritmo, Testes Funcionais do DAC+ADC+MUX

Anexo G – Análise de riscos

Todos os riscos elencados na Tabela 11 abaixo tinham um potencial de afetar a correta execução do projeto. Pudemos citar como riscos a logística de entrega de componentes, a utilização de tecnologias novas, possíveis defeitos de montagem da placa de circuito impresso, mudança de interface conversor-smartphone, e redefinição do circuito do conversor devido ao mal dimensionamento do sistema.

Probabilidade	Frequente				
	Provável		Defeito de Montagem		
	Alto			Atraso na entrega de componentes	
	Médio		Mudança de Interface Conversor-Smartphone		Redefinição do Circuito Conversor
	Baixo				
		Baixo	Médio	Alto	Muito Alto
		Impacto			

Tabela 11 – Análise de riscos do projeto

Lidando com os riscos

A obtenção de componentes no prazo estipulado foi essencial para o correto andamento do projeto. Todos os componentes utilizados puderam ser obtidos facilmente via *samples* e compras de grandes distribuidores internacionais. Como previsto, um dificultador foi o processo na alfândega brasileira que taxou os componentes e, de forma mais significativa, as PCB's (fabricadas na China), praticamente que dobrando o custo de fabricação de transporte destas.

Não houve momento em que os *delays* gerados por tempo de fabricações e entregas de componentes pararam as atividades uma vez que estas eram facilmente paralelizáveis.

Como já relatado no corpo da monografia, a dedicação e cuidados com o projeto e montagem de hardware fez com que esse ponto, cuja classificação na tabela de riscos apontava como provável e de impacto médio, livesse seu impacto quase reduzido a zero, uma vez que durante toda a etapa de testes apenas um componente (1x MOSFET NMOS) precisou ser trocado, gerando um atraso desprezível em relação ao tempo de projeto.

Ao propormos utilizar a conexão USB como interface entre o conversor e smartphone, tínhamos em mente que este protocolo não é trivial e que requer conhecimento profundo de como um driver USB funciona. Neste ponto também, a implementação foi bem-sucedida. A comunicação do projeto final via serial entre microcontrolador e servidor foi implementada no firmware da Tiva C e em MATLAB, sem maiores problemas.

O projeto de layout da PCB exigiu, como previsto, o uso de placas de dupla camada e sua dimensão mínima de trilhas fez com que optássemos por terceirizar a fabricação, uma vez que a tecnologia disponível na universidade nos disponibilizaria equipamentos para trilhas de no mínimo 1 mm, enquanto o projeto possuía trilhas dos MUX's dos eletrodos de 6 Mils (= 0,1524mm). Os custos, então, se elevaram.

Por fim, o projeto mostrou que um dos maiores riscos foi exatamente na implementação do algoritmo de reconstrução de imagem no smartphone. Por se tratar de um projeto de dedicação parcial (a sugestão dos professores da disciplina foi de reservar 5h/semana), a equipe conscientemente adotou uma estratégia de modificar o escopo do projeto mantendo a proposta de uma ferramenta automatizada de geração de imagens de estruturas e localização de fissuras. Visto que o algoritmo de reconstrução de imagem não é trivial aliado à pouca expertise dos membros da equipe em apps mobile, este foi o ponto de maior risco associado no projeto e que exigiu mudanças estratégicas da equipe durante o projeto.

Anexo H – Custo do projeto

O custo do projeto aqui descrito engloba todas as fases do projeto, desde a concepção inicial e prova de conceito até testes e validação. Nesta seção estão todos os custos envolvidos no desenvolvimento do projeto, divididos pela etapas do projeto.

Prova de Conceito

Descrição	Quantidade	Custo	Total
Placa de aço zincado furada	3	R\$ 10,00	R\$ 30,00

Validação de Circuito

Descrição	Quantidade	Custo	Total
MOSFET FDN337	19	R\$ 0,67	R\$ 12,73
MOSFET IRLML6401	19	R\$ 0,81	R\$ 15,39
Resistor shunt de 0,1 ohm	5	R\$ 1,50	R\$ 7,50
MOSFET IRL510PBF	1	R\$ 3,70	R\$ 3,70
Frete	1	R\$ 10,00	R\$ 10,00
			R\$ 49,32

Protótipo

Descrição	Quantidade	Custo	Total
MOSFET FDN337	48	R\$ 0,76	R\$ 36,48
LM317T	3	R\$ 2,22	R\$ 6,66
LM317LM	5	R\$ 2,79	R\$ 13,95
LM337	3	R\$ 2,76	R\$ 8,28
Capacitores	1	R\$ 6,00	R\$ 6,00
Resistores	1	R\$ 10,00	R\$ 10,00
Placa de Circuito Impresso	10	R\$ 40,42	R\$ 404,20
Cabo extensor ATX	1	R\$ 37,10	R\$ 37,10
Jacarê	20	R\$ 1,80	R\$ 36,00
Frete	1	R\$ 10,00	R\$ 10,00
			R\$ 568,67

Teste Finais

Descrição	Quantidade	Custo	Total
Placa de aço zincado furada	5	R\$ 9,60	R\$ 48,00

Custo Total do Projeto	R\$ 695,99
---------------------------------------	-------------------

Abaixo estão estimados os custos de uma unidade do protótipo.

Custo de uma unidade de Protótipo

Descrição	Quantidade	Custo		Total	
MOSFET FDN337	32	R\$	0,76	R\$	24,32
LM317T	1	R\$	2,22	R\$	2,22
LM317LM	2	R\$	2,79	R\$	5,58
LM337	1	R\$	2,76	R\$	2,76
Capacitores	1	R\$	6,00	R\$	6,00
Resistores	1	R\$	10,00	R\$	10,00
Placa de Circuito Impresso	1	R\$	40,42	R\$	40,42
Cabo extensor ATX	1	R\$	37,10	R\$	37,10
Jacarê	20	R\$	1,80	R\$	36,00

**Custo
total de
uma
unidade**

R\$ 164,40