



Universidade Tecnológica Federal do Paraná – UTFPR
Departamento Acadêmico de Eletrônica
Engenharia Industrial Elétrica com ênfase Eletrônica e
Telecomunicações
Trabalho de conclusão de curso 2



AIRTON HITOSHI MITSUYASU
FELIPE DA SILVA RIBEIRO
LUIZ DAVI LEITÃO MARTINS

MICROSCÓPIO DE TUNELAMENTO POR VARREDURA - STM

RELATÓRIO DE TRABALHO DE CONCLUSÃO DE CURSO 2

CURITIBA

2011

AIRTON HITOSHI MITSUYASU
FELIPE DA SILVA RIBEIRO
LUIZ DAVI LEITÃO MARTINS

MICROSCÓPIO DE TUNELAMENTO POR VARREDURA - STM

Relatório do trabalho de conclusão do curso apresentado à disciplina de Trabalho de conclusão de curso 2, como requisito parcial para a obtenção de grau de engenheiro em eletrônica, no Curso de Engenharia Industrial Elétrica com ênfase em Eletrônica e Telecomunicações, da Universidade Tecnológica Federal do Paraná - UTFPR.

Orientadores:

Prof. Dr. Sérgio Francisco Pichorim

Prof. Dr. Fábio Kurt Schneider

CURITIBA

2011

AIRTON HITOSHI MITSUYASU
FELIPE DA SILVA RIBEIRO
LUIZ DAVI LEITÃO MARTINS

MICROSCÓPIO DE TUNELAMENTO POR VARREDURA - STM

Este Trabalho de Conclusão de curso foi julgado e aprovado como requisito parcial para obtenção do título de Engenheiro Industrial Eletricista com ênfase em Eletrônica e Telecomunicações pela Universidade Tecnológica Federal do Paraná.

Curitiba, 04 de dezembro de 2011.

Prof. Dr. Hilton José Silva Azevedo
Coordenador de Curso

Departamento Acadêmico de Eletrônica

Prof. Dr. Sérgio Francisco Pichorim
Orientador do Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

Prof. Dr. Fábio Kurt Schneider
Orientador do Trabalho de Conclusão de Curso
Departamento Acadêmico de Eletrônica

BANCA EXAMINADORA

Prof. Dr. Rubens Alexandre Faria

Prof. Dr. Bertoldo Schneider Jr.

Prof. Dr. Carlos Raimundo Erig Lima

AGRADECIMENTOS

A Deus pela vida e por ter dado a nossa equipe a oportunidade e condições para desenvolver este projeto.

Ao Prof. Dr. Sérgio Francisco Pichorim e Prof. Dr. Fábio Kurt Schneider, pela oportunidade de orientação concedida, pela paciência e pelo apoio.

À Universidade Tecnológica Federal do Paraná por fornecer os laboratórios utilizados e permitir que os alunos estendam seus conhecimentos ao realizarem pesquisas, incentivando o crescimento acadêmico e pessoal.

A todos aqueles que de alguma maneira auxiliaram no desenvolvimento do projeto, entre eles professores Julio Cesar Klein das Neves, Arandi Ginane Bezerra Jr, Bertoldo Schneider Jr., e ao professor Wolfgang Vollmann. Cada um a sua maneira contribuiu grandiosamente com a realização desse projeto.

Aos colegas da engenharia, amigos e amigas, pelas boas dicas, assessoramentos, *network*, e horas de conversas animadoras mesmo nos momentos de *stress* e dificuldade.

Às nossas mães, pais, irmãos, tios, tias, primos, primas e outros familiares pelo apoio irrestrito, sempre incentivando e acreditando no sucesso desse projeto.

Às namoradas também pelo apoio, paciência, e incentivo nas horas mais complicadas.

A CAPES pelo investimento financeiro nas compras de materiais dos mais diversos tipos que possibilitaram a execução deste projeto.

RESUMO

MARTINS; Luiz Davi Leitão; MITSUYASU, Airton Hitoshi; RIBEIRO, Felipe da Silva.

Microscópio de tunelamento por tunelamento – STM. 2011. Trabalho de conclusão de curso – Universidade Tecnológica Federal do Paraná, Curitiba, 2011.

A nanotecnologia tem potencial para ser responsável pela próxima grande revolução tecnológica, já que pode ser empregada nas mais diferentes áreas do conhecimento. Segundo informações da Agência Brasil, o mercado brasileiro de produtos com base em nanotecnologias desenvolvidas originalmente no país, somou no ano passado cerca de R\$ 115 milhões, posicionando o Brasil como responsável por 0,03% da produção mundial de nanotecnologia (AGÊNCIA BRASIL 2011). Esse projeto tem por objetivo desenvolver um microscópio de tunelamento por varredura (do inglês, *Scanning Tunneling Microscope* – STM) de baixo custo. A metodologia aplicada ao projeto é o modelo clássico, seguindo estes passos: referencial teórico, projeto, implementação e testes, sendo a parte crítica o teste do *hardware*. Durante o decorrer do projeto, faz parte da metodologia adotada a documentação dos passos e decisões tomadas para permitir a utilização e reaproveitamento do conhecimento técnico adquirido em projetos científicos posteriores. O microscópio foi constituído de quatro grandes módulos: o módulo que permite a detecção da corrente de tunelamento, o módulo de varredura da superfície, o módulo de digitalização e pré-processamento, e o módulo de processamento e geração de imagem. Esse sistema realiza a varredura com a sonda na superfície a ser analisada, e concomitantemente é realizada a detecção da corrente gerada por efeito túnel entre a superfície e a sonda. O sistema de controle é responsável pela varredura e correção da altura da sonda. O microcontrolador coleta e digitaliza as informações de posição no plano cartesiano (XY) e altura da sonda, e envia esses dados a um computador, onde o *software* desenvolvido no projeto gera as imagens a partir dos dados obtidos. Dessa forma o projeto tem como resultado um sistema que possui *hardware*, *firmware* e *software* específicos a essa aplicação, com custo total para compra cerca de 50% mais barato que um equipamento semelhante a este encontrado na Europa. Outro resultado é o auxílio à UTFPR no sentido de viabilizar a implantação de uma linha de pesquisa envolvendo nanotecnologia.

Palavras-chave: nanotecnologia, microscopia de tunelamento, efeito túnel, STM.

ABSTRACT

MARTINS; Luiz Davi Leitão; MITSUYASU, Airton Hitoshi; RIBEIRO, Felipe da Silva.

Scanning tunneling microscope – STM. 2011. Trabalho de conclusão de curso – Universidade Tecnológica Federal do Paraná, Curitiba, 2011.

Nanotechnology has the potential to be responsible for the next great technological revolution, since it acts in different areas of knowledge. According to Agência Brasil, the Brazilian market for products based on nanotechnologies originally developed in the country last year totaled about \$115 million, placing Brazil as responsible for 0.03% of the world of nanotechnology (AGÊNCIA BRASIL 2011). This project aims to develop a scanning tunneling microscope at low cost. The methodology applied to the project is the classic model by following these steps: research, design, implementation and testing, in which hardware is a critical part. During its course, the methodology included a process of documenting the steps and decisions in order to allow the use and reuse of expertise acquired in subsequent scientific projects. The microscope is composed of four main modules: the module that allows the tunneling current's detection, the surface scanning module, the discretization and pre-processing module, and the processing and imaging one. This system can perform a scan with a probe on the surface to be analyzed, which is held concurrently by detecting the current generated by tunnel effect between the surface and probe. The control system is responsible for scanning and correcting the probe height. The microcontroller collects and digitizes the position information (XY) and the probe's height, and sends that data to the computer, where there is software that generates images from the data obtained. The final goal of this project is a system that has hardware, firmware and software specific to that application, with commercial value about 50% cheaper than similar to some equipment found in Europe. Another result to UTFPR was the first steps in order to enable some nanotechnology research to take place at the university.

Key-words: nanotechnology, tunneling microscopy, tunnel effect, STM.

LISTA DE QUADROS

Quadro 1 - Principais funções do módulo de gerenciamento da conexão USB.....	58
Quadro 2 - Definições comuns SVN.....	69
Quadro 3 - Cronograma detalhado.....	73
Quadro 4 - Datas importantes.....	74
Quadro 5- Responsabilidade das ações e custo.....	75
Quadro 6- Análise de riscos.....	76
Quadro 7- Definição do negócio.....	102
Quadro 8 - Dados da empresa.....	102
Quadro 9 - Produtos e Serviços.....	102
Quadro 10 - Análise de mercado.....	103
Quadro 11 - Perfil do cliente.....	103
Quadro 12 - Estratégias de Marketing.....	103
Quadro 13 - Metas de Vendas.....	104
Quadro 14 - Demonstração resumida de lucro.....	104
Quadro 15 - Marcos Relevantes.....	105
Quadro 16 - Visão.....	105
Quadro 17 - Missão.....	105
Quadro 18 - Gestão de Qualidade.....	105
Quadro 19 - Valores.....	106
Quadro 20 - Aspectos Jurídicos.....	107
Quadro 21 - Composição de Capital.....	107
Quadro 22 – Conselheiros.....	107
Quadro 23 - Equipe gerencial.....	108
Quadro 24 - Descrição STM.....	109
Quadro 25 - Descrição STM Software.....	109
Quadro 26 - Ciclo de vida dos produtos.....	110
Quadro 27 - Produtos e Serviços Planejados.....	110
Quadro 28 - Definição de mercado.....	111
Quadro 29 - Segmento de Mercado.....	111
Quadro 30 - Pontos Fortes e Fracos.....	112
Quadro 31 - Oportunidades e ameaças.....	112
Quadro 32 - Perfil do cliente.....	113
Quadro 33 – Concorrência.....	113
Quadro 34 - Riscos.....	114
Quadro 35 - Objetivos.....	115
Quadro 36 - Metas de vendas.....	115
Quadro 37 - Estratégia de desenvolvimento de produtos.....	115
Quadro 38 - Material de divulgação.....	116
Quadro 39 - Equipe de Vendas.....	116
Quadro 40 - Canais de comercialização.....	116
Quadro 41 – Parcerias.....	117
Quadro 42 - Políticas de preços.....	117
Quadro 43 - Investimentos iniciais.....	118
Quadro 44 - Custos fixos mensais.....	119
Quadro 45 – Impostos.....	119
Quadro 46 – Faturamento.....	120
Quadro 47 - Margem de contribuição.....	120
Quadro 48 - Ponto de equilíbrio.....	120
Quadro 49 - Lucro estimado.....	121

Quadro 50 - Estoque inicial	121
Quadro 51 - Fluxo de caixa ano de 2012.....	122
Quadro 52 - Fluxo de caixa ano de 2013.....	122
Quadro 53 - Fluxo de caixa ano de 2013.....	123
Quadro 54 - Investimento necessario	123
Quadro 55 – Rentabilidade.....	124
Quadro 56 - Prazo de retorno	124
Quadro 57 - VPL	124

LISTA DE FIGURAS

Figura 1 – Diagrama em blocos do sistema completo.....	15
Figura 2 – Modelo inicial da estrutura mecânica do microscópio.....	17
Figura 3 – Princípio de funcionamento de um STM	23
Figura 4 – Comportamento de material piezelétrico (PINO e SOUZA 2011)	24
Figura 5 – Disco piezelétrico dividido em quadrantes	31
Figura 6 – conjunto de movimentação e sonda	32
Figura 7 – Buffer de Polarização	33
Figura 8 – Amplificador de transimpedância	34
Figura 9 – Controle de posição.....	36
Figura 10 - Malha de controle PI analógica	38
Figura 11 - Condicionamento do sinal CONTROLEZ.....	41
Figura 12 - (a) aquisição do sinal SCT1 (b) entrega do sinal processado.	42
Figura 13 - Microcontrolador PIC 18F4550 (MICROCHIP Technology Inc. 2004).....	44
Figura 14 - Pinagem do Conversor Digital-Analógico MCP4822 (MICROCHIP 2010)	45
Figura 15 – Ciclo de leitura do MCP4822 (MICROCHIP Technology Inc. 2010).....	46
Figura 16 - conexão USB	47
Figura 17- Fluxograma do firmware para o uso da malha de controle analógico.....	48
Figura 18- Fluxograma do controle digital.....	51
Figura 19 – Visão geral do componente de software	57
Figura 20 - Tela do software de monitoramento na configuração simplificada.....	58
Figura 21 - Tela do software de monitoramento na configuração avançada.....	59
Figura 22 - Dados do Dispositivo.....	59
Figura 23 - Parâmetros de Varredura	60
Figura 24 - Representação gráfica dos parâmetros de varredura.....	61
Figura 25 - Diagrama de sequência do acesso de um usuário a Galeria STM	62
Figura 26 - Diagrama do componente MATLAB para .NET.....	63
Figura 27 – página inicial da SoluçãoWeb.....	65
Figura 28 – Print da solução Web para PC.....	66
Figura 29 – Print da solução Web para tablets	67
Figura 30 – Fotografia da solução Web acessada em um smartphone.....	67
Figura 31 – Diagrama de blocos do sistema de controle de versões	68
Figura 32 - Sequência de fluxo de informações num repositório SVN (WONG 2010).....	70
Figura 33 - Sonda e sua fixação no detalhe	79
Figura 34 - Microscópio completo	79
Figura 35 - Placa com alimetacao da rede.....	80
Figura 36 - Circuitos do prototipo final.....	82
Figura 37 - Montagem da experiência para verificação da deflexão do disco piezelétrico.....	83
Figura 38 - Visualização da dependência exponencial da corrente de tunelamento	86
Figura 39 - Gráfico SCT vs Tempo e linha de tendência exponencial vs Tempo.....	88
Figura 40 - Gráfico ln(SCT) vs Tempo e linha de tendência vs Tempo	88
Figura 41 - amostra um – completa	89
Figura 42 - amostra um - primeira fatia.....	89
Figura 43 - amostra um - segunda fatia	90
Figura 44 - amostra um - terceira fatia	91
Figura 45 - amostra um - quarta fatia	92
Figura 46 – amostra dois - dia anterior.....	93
Figura 47 - amostra dois - dia posterior.....	94
Figura 48 - amostra dois - vista inclinada	95

Figura 49- amostra dois - fatia da esquerda.....	95
Figura 50 - amostra dois – fatia da direita.....	97
Figura 51 - amostra dois - fatia central.....	96

SUMÁRIO

1.	INTRODUÇÃO.....	12
	1.1. OBJETIVOS.....	12
	1.2. JUSTIFICATIVA.....	14
	1.3.ESCOPO.....	15
	1.4.METODOLOGIA.....	16
	1.5.APRESENTAÇÃO DO DOCUMENTO.....	18
2.	FUNDAMENTOS TEÓRICOS.....	19
	2.1.EFEITO TÚNEL.....	19
	2.2.MODO DE CORRENTE CONSTANTE.....	21
	2.3.MODO DE ALTURA CONSTANTE.....	21
	2.4.EFEITO PIEZELÉTRICO.....	22
	2.5.COMUNICAÇÃO SPI.....	22
	2.6.COMUNICAÇÃO USB.....	23
	2.7.CONSIDERAÇÕES.....	24
3.	ESCOPO DO PRODUTO.....	25
	3.1.MERCADO CONCORRENTE.....	25
	3.2.MERCADO CONSUMIDOR.....	25
	3.3.ÁNALISE DE PATENTES.....	25
	3.4.CONSIDERAÇÕES.....	27
4.	DESENVOLVIMENTO.....	28
	4.1.HARDWARE.....	28
	4.1.1.MÓDULO DE DETECÇÃO DE CORRENTE DE TUNELAMENTO.....	28
	4.1.1.1 TRANSDUTOR PIEZELÉTRICO.....	28
	4.1.1.2.BUFFER DE POLARIZAÇÃO.....	29
	4.1.1.3.AMPLIFICADOR DE TRANSIMPEDÂNCIA.....	30
	4.1.2.MÓDULO DE VARREDURA XY.....	32
	4.1.2.1.CONTROLE DE POSIÇÃO.....	32
	4.1.2.2.MALHA DE CONTROLE PI ANALOGICA.....	35
	4.1.2.3. MALHA DE CONTROLE PI DIGITAL.....	37
	4.1.3.MÓDULO DE DIGITALIZAÇÃO E PRÉ-PROCESSAMENTO.....	39
	4.1.4.MICROCONTROLADOR PIC 18F4550.....	41
	4.1.5 CONVERSOR DIGITAL ANALÓGICO MCP4822.....	43
	4.1.6 COMUNICAÇÃO SPI.....	44
	4.1.7 COMUNICAÇÃO USB.....	44
	4.2. FIRMWARE.....	45
	4.2.1 MALHA DE CONTROLE ANALOGICA.....	45
	4.2.2 MALHA DE CONTROLE DIGITAL.....	49
	4.3.PROTOCOLO DE COMUNICAÇÃO USB ENTRE FIRMWARE E SOFTWARE.....	52
	4.3.1.PROTOCOLO DE CONEXÃO.....	53
	4.3.2. PROTOCOLO PARA INICIAR VARREDURA.....	53
	4.3.3. PROTOCOLO PARA INDICAR INICIO DE VARREDURA.....	53
	4.3.4. ESTRUTURAÇÃO DO PACOTE DE DADOS.....	54
	4.3.5 PROTOCOLO PARA INDICAR FINAL DE VARREDURA.....	55
	4.4. SOFTWARE.....	55
	4.5.SEGURANÇA DE INFORMAÇÕES E CONTROLE DE VERSÕES.....	67
	4.6.CONSIDERAÇÕES.....	70
5.	GESTÃO.....	73
	5.1.CRONOGRAMA E CUSTOS.....	73
	5.2.RISCOS.....	76
	5.3.CONSIDERAÇÕES.....	77
6.	RESULTADOS.....	78
	6.1.ESTRUTURA FÍSICA.....	78
	6.1.1.VIBRAÇÕES MECÂNICAS.....	78
	6.1.2 CONTRUÇÃO DA ESTRUTURA MECÂNICA.....	78
	6.2. HARDWARE ELETRÔNICO.....	79

6.3.	DEFLEXÃO DO DISCO PIEZELÉTRICO.....	82
6.4.	CORRENTE DE TUNELAMENTO.....	84
6.5.	COMPROVAÇÃO POR REPETITIVIDADE.....	87
6.5.1.	PRIMEIRO TESTE DE REPETIÇÕES DE PADRÕES ENTRE MEDIÇÕES.....	87
6.5.2.	SEGUNDO TESTE DE REPETIÇÕES DE PADRÕES ENTRE MEDIÇÕES.....	91
6.6.	DISCUSSÕES.....	95
6.7.	CONSIDERAÇÕES.....	96
7.	PLANO DE NEGÓCIOS.....	98
7.1.	INTRODUÇÃO.....	98
7.2.	RESUMO EXECUTIVO.....	99
7.2.1.	DEFINIÇÃO DO NEGÓCIO.....	99
7.2.2.	DADOS DA EMPRESA.....	99
7.2.3.	PRODUTOS E SERVIÇOS.....	99
7.2.4.	ANÁLISE DE MERCADO.....	100
7.2.5.	PERFIL DO CLIENTE.....	100
7.2.6.	PLANO DE MARKETING E VENDAS.....	100
7.2.7.	PLANO FINANCEIRO.....	101
7.3.	VISÃO E MISSÃO.....	102
7.3.1.	MARCOS RELEVANTES.....	102
7.3.2.	VISÃO.....	102
7.3.3.	MISSÃO.....	102
7.3.4.	GESTÃO DE QUALIDADE.....	102
7.3.5.	VALORES.....	103
7.4.	DADOS DA EMPRESA.....	104
7.4.1.	ASPECTOS JURÍDICOS.....	104
7.4.2.	CONSELHEIROS.....	104
7.4.3.	RESPINSABILIDADE GERENCIAL: EQUIPE DE GERENTES.....	105
7.4.4.	ADVOGADO E CONTADOR.....	105
7.5.	PRODUTOS E SERVIÇOS.....	106
7.5.1.	PRODUTOS E SERVIÇOS DISPONÍVEIS.....	106
7.5.1.1.	DESCRIÇÃO DA LINHA DE PRODUTOS.....	106
7.5.1.2.	CICLO DE VIDA DO PRODUTO.....	107
7.5.1.3.	PRODUTOS E SERVIÇOS PLANEJADOS.....	107
7.6.	ANÁLISE DE MERCADO.....	108
7.6.1.	DEFINIÇÃO DE MERCADO.....	108
7.6.2.	SEGMENTO DE MERCADO.....	108
7.6.3.	PONTOS FORTES E FRACOS.....	109
7.6.4.	OPORTUNIDADE E AMEAÇAS.....	109
7.6.5.	PERFIL DO CLIENTE.....	110
7.6.6.	CARACTERIZAÇÃO DA CONCORRÊNCIA.....	110
7.6.7.	RISCOS.....	111
7.7.	PLANO DE MARKETING E VENDAS.....	112
7.7.1.	OBJETIVOS E METAS QUANTO AS VENDAS E AOS CLIENTES.....	112
7.7.2.	ESTRATÉGIAS DE MARKETING.....	112
7.7.2.1.	PARA O MERCADO NACIONAL.....	112
7.8.	PLANO FINANCEIRO.....	115
7.8.1.	INVESTIMENTOS.....	115
7.8.2.	CUSTOS FIXOS MENSAIS.....	116
7.8.3.	IMPOSTOS.....	116
7.8.4.	FATURAMENTO.....	117
7.8.5.	MARGEM E CONTRIBUIÇÃO.....	117
7.8.6.	PONTO DE EQUILÍBRIO.....	117
7.8.7.	LUCRO ESPERADO PARA O FATURAMENTO ESTIMADO.....	118
7.8.8.	ESTOQUE INICIAL.....	118
7.8.9.	NECESSIDADE DE CAPITAL DE GIRO.....	118
7.8.10.	INVESTIMENTO NECESSÁRIO.....	120
7.8.11.	TAXA DE RENTABILIDADE.....	121
7.8.12.	PRAZO DE RETORNO.....	121
7.8.13.	ANÁLISE DE VPL(VALOR LÍQUIDO PRESENTE).....	121
7.9.	CONSIDERAÇÕES.....	122

8.	CONCLUSÃO FINAL.....	123
9.	REFERÊNCIAS BIBLIOGRÁFICAS.....	126

1. INTRODUÇÃO

O desenvolvimento da nação brasileira tem possibilitado a ampliação de campos de pesquisa antes inexistentes ou de fraca presença no país. Na área de nano instrumentação, as ferramentas atualmente disponíveis no mercado são em sua maioria importadas e possuem preços que dificultam o seu acesso e aquisição por parte de centros de pesquisa.

O microscópio de tunelamento por varredura aqui proposto tem sua principal vantagem na ênfase no baixo custo. Microscópios que utilizam o mesmo princípio tem um preço proibitivo para a grande maioria dos centros de pesquisa do país. E isso se deve, além do custo da tecnologia e material utilizados, a que o equipamento é normalmente importado, acrescentando no seu preço final a carga tributária referente à importação.

Este relatório apresenta uma abordagem técnica sobre o microscópio de tunelamento por varredura, bem como um capítulo financeiro baseado na prova de conceito para teste de viabilidade econômica do produto no mercado nacional.

Além disso, o texto contém o estudo de mercado realizado, o detalhamento dos módulos que formam o sistema, além de um panorama geral sobre a gestão do projeto. Este primeiro capítulo tem por objetivo informar as justificativas para a escolha do projeto, os objetivos pretendidos, a metodologia adotada e explicar a estrutura do documento.

1.1. OBJETIVOS

Este trabalho tem por objetivo desenvolver um microscópio de tunelamento por varredura (do inglês, *Scanning Tunneling Microscope* – STM) de baixo custo. Esse equipamento poderia ser facilmente utilizado em empresas e faculdades que possuem verba reduzida para aquisição dos mesmos.

Para tanto, serão desenvolvidos os seguintes módulos:

- Um módulo da detecção de corrente de tunelamento.
- Um módulo de varredura XY.
- Um módulo de digitalização e pré-processamento.
- Um módulo de processamento e geração de imagem final.

O diagrama do sistema completo é mostrado na *Figura 1*. O módulo de detecção de corrente de tunelamento cria condições favoráveis ao efeito túnel, e por consequência para a corrente de tunelamento fluir, além de amplificar essa corrente e transformá-la em uma tensão

possível de ser processada. O módulo de varredura XY é responsável pelo posicionamento da sonda. O módulo de digitalização e pré-processamento transforma o sinal analógico em um sinal discretizado em tempo e amplitude e faz o controle de altura da sonda, monta e envia pacotes de dados para o módulo seguinte, o de processamento e geração de imagens. Este módulo recebe esses pacotes de dados e a partir da informação adquirida gera uma imagem em três dimensões.

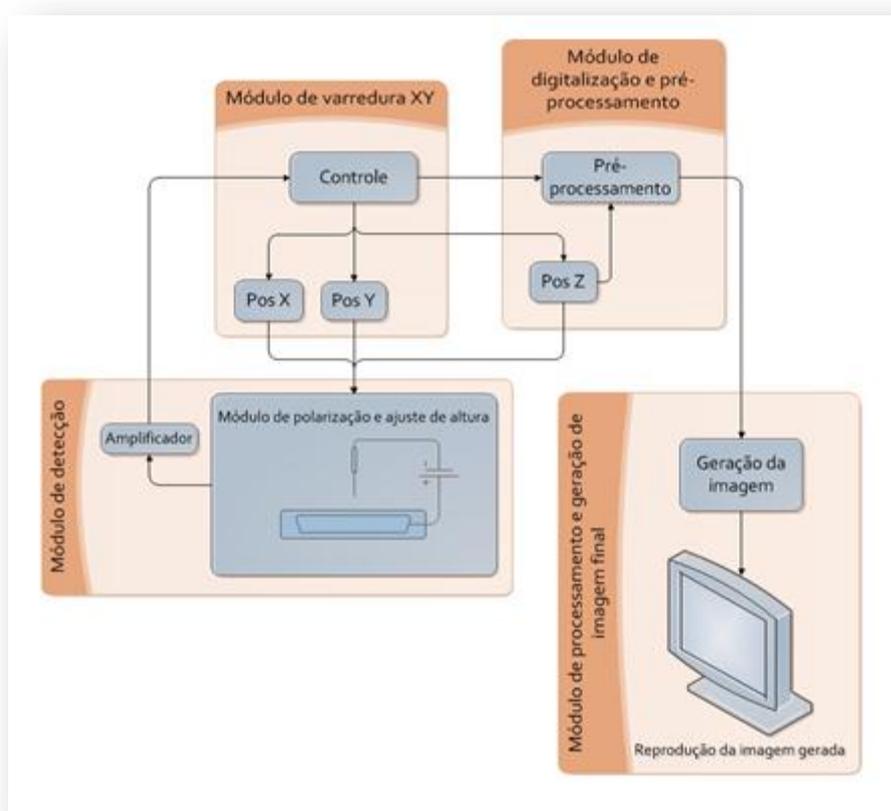


Figura 1 – Diagrama em blocos do sistema completo

Fonte: Própria

1.2. JUSTIFICATIVA

A nanotecnologia estuda a compreensão do comportamento da matéria em dimensões um bilhão de vezes menores que o metro, e tem potencial de manipular elementos em nível atômico de maneira a controlar as propriedades dos materiais numa escala antes impossível. Atuando nas mais variadas áreas de pesquisa como a medicina, engenharia

eletrônica, engenharia dos materiais, física, química, biologia, a nanotecnologia tem capacidade de revolucionar grande parte da tecnologia atual.

Nesse contexto, um microscópio que permita a análise de superfícies com alta resolução é uma ferramenta indispensável nas pesquisas de nanotecnologia. Em 1981 o microscópio de tunelamento por varredura (do inglês, *Scanning Tunneling Microscope* – STM) foi inventado na IBM em Zurique na Suíça. Invenção que rendeu o prêmio Nobel aos seus inventores em 1986. Esse aparelho deu origem a um novo segmento de instrumentos de visualização em escala nanométrica.

A Europa, por ser pioneira nesta tecnologia, é detentora de conhecimentos mais avançados sobre os microscópios de tunelamento. A partir de um programa de intercâmbio entre a Universidade Tecnológica Federal do Paraná e a *Beuth Hochschule für Technik Berlin* iniciou-se uma pesquisa com o objetivo de construir um STM de baixo custo em Berlim. Para que o aproveitamento do programa de intercâmbio seja pleno, os conhecimentos adquiridos na universidade alemã devem ser compartilhados por ambas as instituições de ensino, sendo essa uma justificativa tecnológica.

Além dessa motivação, os preços destes STMs de baixo custo na Alemanha se encontram em média no valor de € 8000,00. Para uma universidade brasileira, que normalmente possui isenção fiscal, o valor calculado fica em torno de R\$ 21.000,00 para importação. A resolução média destes aparelhos está na faixa de 8 pm (picômetros) para a varredura XY e 4 pm (picômetros) para a varredura Z.

Segundo informações da Agência Brasil, o mercado brasileiro de produtos com base em nanotecnologias desenvolvidas originalmente no país, somou no ano passado cerca de R\$ 115 milhões, posicionando o Brasil como responsável por 0,03% da produção mundial de nanotecnologia.

Baseado nesses dados e utilizando o conhecimento adquirido na universidade parceira, justifica-se o desenvolvimento de um STM de baixo custo na UTFPR. O valor comercial desse produto é no mínimo 50% mais barato que o equipamento acima descrito. Assim sendo possibilitará o acesso a esse tipo de tecnologia a outros laboratórios que antes não o teriam por questões econômicas, além de contribuir para o aumento da produção nacional de nanotecnologias.

1.3. ESCOPO

O sistema é composto por uma parte mecânica e outra eletrônica.

A parte mecânica é composta por duas placas paralelas separadas por parafusos micrométricos fixados em suas extremidades. Esses parafusos são responsáveis pelo ajuste grosso da distância entre a sonda e o substrato a ser visualizado. Em uma das extremidades são fixados a sonda e o material piezelétrico, que se conectam através de fios com o sistema eletrônico.

A *Figura 2* permite uma visualização em três dimensões dessa estrutura.

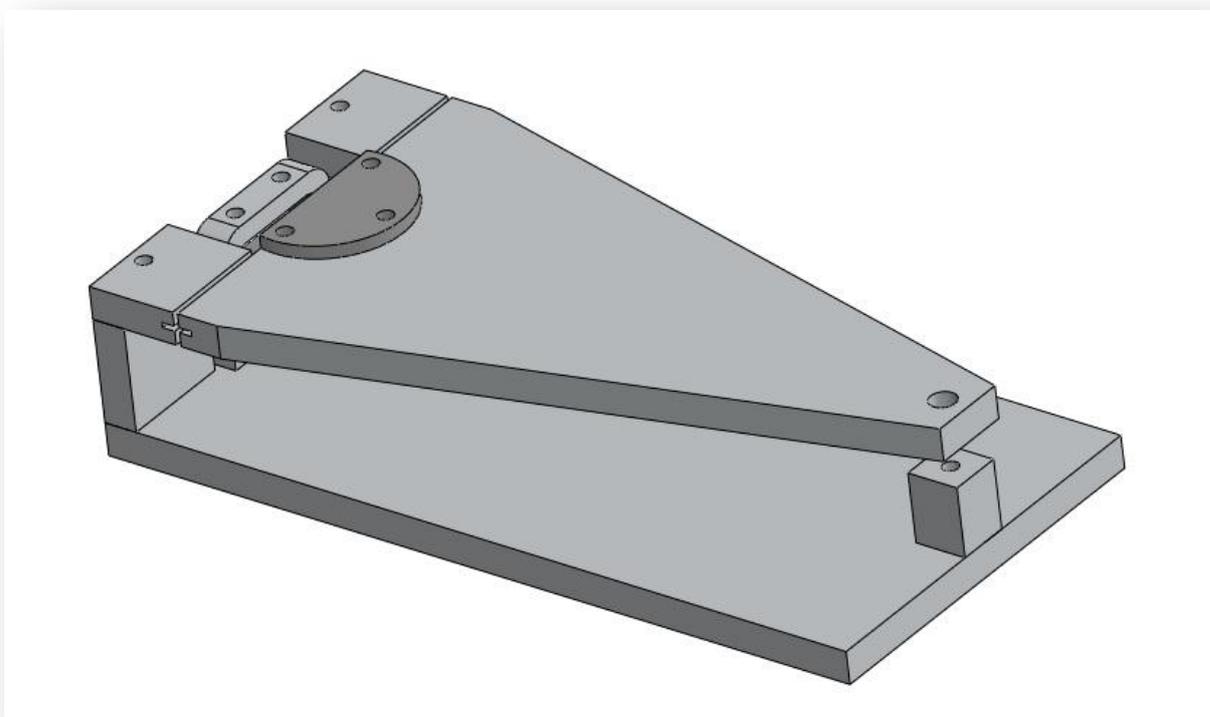


Figura 2 – Modelo inicial da estrutura mecânica do microscópio

Fonte: Própria

A parte eletrônica foi implementada da seguinte maneira:

- O módulo de detecção da corrente de tunelamento é constituído pelo bloco de polarização e ajuste de altura da sonda (sentido Z), que permite gerar uma corrente de tunelamento, e pelo bloco amplificador, que amplifica e filtra a corrente gerada através do

amplificador operacional LF411CN, gerando o que se chama de sinal de tunelamento. Esse sinal passando pelo bloco de controle permite o ajuste da altura da sonda.

- Para fazer a varredura (Módulo de varredura XY) são utilizados: um sistema com amplificadores operacionais TL071, o microcontrolador PIC 18F4550 e o conversor digital-analógico MCP4822. A frequência de varredura de cada eixo é independente e definido no *firmware* do microcontrolador. As entradas dos amplificadores são controladas por um sinal analógico enviado pelo microcontrolador depois de passarem pelo D/A. O sinal de saída dos amplificadores é aplicado em um material piezelétrico conectado a sonda, fazendo com que a sonda se mova no sentido XY.

- Com o conhecimento da posição XY e fazendo amostragem periódica do sinal de tunelamento sincronizada com a varredura, amostras da corrente de tunelamento (responsáveis pela informação da posição Z), o módulo de digitalização e pré-processamento tem informações necessárias para fazer o condicionamento dos sinais. Então fazer com que sejam passados ao módulo seguinte, o módulo de processamento e geração da imagem final. Estes dados pré-processados são enviados a um computador através de uma conexão do tipo USB e são transferidos ao *software* responsável pelo processamento e geração da imagem final.

1.4. METODOLOGIA

A metodologia aplicada ao projeto foi o modelo clássico, seguindo os seguintes passos: estudos, projeto, implementação e testes, sendo a parte crítica o teste do *hardware*. Durante o decorrer do mesmo, fez parte da metodologia adotada, um processo de documentação dos passos e decisões tomadas com o objetivo de permitir a utilização e reaproveitamento do conhecimento técnico adquirido em projetos científicos posteriores.

- **Pesquisar e estudar o princípio físico do efeito túnel e tecnologias que podem ser aplicados no projeto.**

O desenvolvimento deste projeto teve início na pesquisa de toda a base teórica do efeito túnel e a possibilidade de sua aplicação na construção de microscópios de tunelamento. Essa pesquisa se deteve no entendimento apenas a compreensão do princípio físico e das tecnologias que são aplicadas na construção de *hardwares* semelhantes ao proposto. Durante todo o processo, uma documentação dos passos e escolhas tomadas foi feita em paralelo para facilitar a compreensão do desenvolvimento em si e a elaboração deste relatório. Para as tais decisões, a

maior quantidade possível das informações relevantes foram observadas e devidamente registradas, importante para justificar da melhor maneira os passos tomados.

- **Construir a estrutura mecânica.**

Após o embasamento teórico, iniciou-se a construção da estrutura mecânica. Para a sua fabricação, o auxílio de estudantes do curso de engenharia mecânica da UTFPR se fez necessário pela falta de experiência dos integrantes do projeto com construções mecânicas.

- **Desenvolver o *hardware*.**

Paralelamente à construção da estrutura mecânica ocorreu o desenvolvimento o *hardware* eletrônico, que foi subdividido em duas etapas construtivas principais. A primeira é o projeto dos circuitos de polarização, amplificação, varredura, controle, digitalização e comunicação com um computador. A segunda se refere à especificação dos componentes e verificação de sua disponibilidade no mercado. Ainda nesta última etapa foi efetuada a aquisição dos componentes, confecção das placas de circuito impresso, montagem das placas e integração do *hardware* com a estrutura mecânica.

- **Executar testes.**

Com a estrutura mecânica e o projeto eletrônico pronto, foram executados testes para conhecer melhor as características do piezolétrico e capacidade de resolução do microscópio. Esses testes permitiram uma análise mais detalhada das informações recuperadas e sua devida documentação para os passos seguintes do projeto. Os resultados desses testes são discutidos no capítulo 8.

- **Desenvolver o *firmware* para digitalização, pré-processamento dos sinais e comunicação com o PC**

O *firmware* foi desenvolvido para um microcontrolador PIC de arquitetura RISC da fabricante Microchip, com o intuito de digitalização dos sinais correspondentes a corrente de tunelamento (plano Z), envio dos sinais de varredura no plano XY, além da comunicação com um computador. Após esse pré-processamento no microcontrolador, desenvolveu-se a troca de dados por meio de uma conexão USB para interface entre o mesmo e o computador.

- **Reproduzir a imagem no PC a partir dos dados obtidos.**

O processamento feito no PC tem os seguintes objetivos básicos: a reconstrução da imagem a partir dos dados enviados pelo microcontrolador e o processamento da mesma para melhor visualização por parte do usuário. Além disso, faz parte desse subitem a conversão da imagem reproduzida num formato compactado e a preparação para a exibição na tela. O STM dispõe de um sistema capaz de mostrar numa tela a representação gráfica de uma superfície metálica em nível molecular.

1.5. APRESENTAÇÃO DO DOCUMENTO

Este relatório foi dividido em oito capítulos. No capítulo 2, são apresentados os fundamentos teóricos, através de um estudo feito na literatura, base para o entendimento dos princípios de funcionamento do efeito túnel. No capítulo 3, é feita a definição do escopo do produto através de um estudo de mercado que engloba a análise dos clientes, dos concorrentes e de patentes pertinentes. O capítulo 4 mostra todo o processo de desenvolvimento de *hardware*, *firmware* e *software*. No capítulo 5, é descrito o processo de gestão do projeto, com a apresentação do cronograma utilizado, relação dos custos e os riscos avaliados no início do projeto. No capítulo 6 são apresentados os resultados conseguidos com o microscópio construído e discussões sobre esses resultados. O capítulo 7 apresenta um plano de negócios, baseado na prova de conceito. Por fim, o capítulo 8 traz as conclusões obtidas.

2. FUNDAMENTOS TEÓRICOS

Um STM é um instrumento que auxilia na análise de superfícies e é constituído basicamente por uma sonda, uma cerâmica piezolétrica para posicionar o sensor e executar a varredura, um circuito de controle de altura da sonda, um sistema de armazenamento, processamento e exibição da imagem (LINDSAY 2008). Um STM pode operar basicamente em dois modos: de corrente constante e altura constante.

2.1. EFEITO TÚNEL

O efeito túnel é um fenômeno relacionado com a mecânica quântica, portanto os formalismos matemáticos da mecânica clássica não se enquadram. Ele acontece quando na trajetória de uma partícula há uma barreira com energia maior que a da própria partícula. Do ponto de vista da mecânica clássica, a partícula seria totalmente refletida. Contudo, sob a ótica da mecânica quântica existe uma probabilidade que essa mesma partícula tunele por esse obstáculo, ou seja, “atravesse” essa barreira e apareça do outro lado. Isso ocorre porque o elétron, quando analisado pela mecânica quântica, possui comportamento dual, onda-partícula. Essa probabilidade é exponencialmente proporcional à largura da barreira (PARK SYSTEMS s.d.).

Quando a largura dessa barreira for suficientemente pequena (cerca de 100 nm), a probabilidade do tunelamento ocorrer aumenta para muitos elétrons, de maneira que eles passam de uma superfície à outra, formando a denominada corrente de tunelamento. A direção do tunelamento é dependente da tensão aplicada entre amostra e sonda (LAMAS 1999).

O efeito túnel pode ser verificado em microscópios de tunelamento por varredura (STM), onde os elétrons tunelam entre a sonda e a superfície.

O princípio de tunelamento de um STM consiste em uma sonda metálica que está próxima à superfície a ser analisada. A corrente de tunelamento aparece entre a sonda e a amostra quando estas estão polarizadas (BINNIG e ROHRER 1983).

A corrente de tunelamento, no caso da polarização sonda-amostra ter ordem de grandeza na casa de dezenas de milivolts, é dada pela equação 1 (BINNIG e ROHRER 1983).

$$I \sim V_t \cdot e^{-A\sqrt{\phi} \cdot d} \quad (1)$$

onde:

I: é a corrente de tunelamento,

V_t : é a tensão de polarização,

ϕ : é a altura média da barreira (função trabalho),

d: é a distância da sonda até a amostra,

A: ≈ 1 se a unidade de ϕ eV (elétron Volts) e a unidade de d for Å (ångström).

De acordo com Binnig e Rohrer, para funções de trabalho na ordem de alguns eV, a corrente de tunelamento se altera de uma ordem de magnitude para cada ångström de variação de d (BINNIG e ROHRER 1983).

O princípio de funcionamento desse tipo de microscópio (ilustrado na Figura 3) é dado pelos passos:

- A) Polarizar a sonda e a amostra com uma tensão contínua de baixa amplitude.
- B) Aproximar sonda e amostra até que ocorra o efeito túnel.
- C) A corrente de tunelamento deve ser verificada usando-se um amperímetro.

Dessa forma, a partir da leitura da corrente de tunelamento é possível estimar a distância entre a sonda e a superfície da amostra. Aplicando esse procedimento a vários pontos da amostra, pode-se montar uma imagem muito próxima da superfície real.

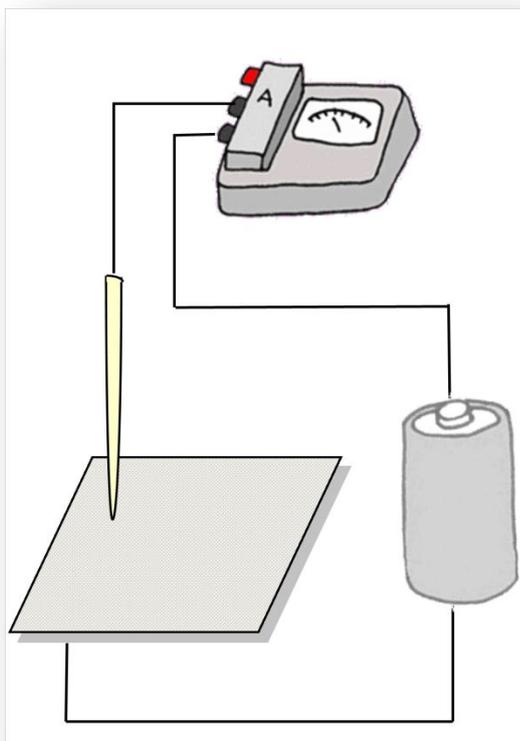


Figura 3 – Princípio de funcionamento de um STM

Fonte: Própria

2.2. MODO DE CORRENTE CONSTANTE

O modo de operação de corrente constante é o mais comumente utilizado. O princípio básico deste método é que a sonda guarda uma distância constante da amostra, mantendo assim a corrente de tunelamento também constante. A imagem pode atingir resolução atômica. Ela é obtida amostrando-se o sinal que representa a variação de altura da sonda, ou seja, o sinal aplicado ao sistema de controle de altura da sonda. Porém, apresenta uma velocidade de varredura reduzida.

2.3. MODO DE ALTURA CONSTANTE

O modo de altura constante é um método menos comum. Ele apresenta um modelo de operação no qual a sonda mantém-se a uma altura constante, ou seja, diferentemente do modo de corrente constante, não há correção na altura da sonda. Dessa forma, a própria variação da

corrente de tunelamento é a informação utilizada para a geração da imagem. Isso possibilita uma velocidade de varredura maior, porém apresenta uma resolução inferior ao método anterior. Esse método permite inclusive, devido à maior velocidade de varredura, a geração de vídeo.

2.4. EFEITO PIEZELÉTRICO

O efeito piezométrico é um fenômeno associado à conversão de energia elétrica em deformação mecânica ou vice-versa. Esse efeito é encontrado em alguns materiais cerâmicos, cristais e semicondutores. A piezoelectricidade ocorre quando um material é capaz de gerar cargas elétricas (tensão elétrica) em sua superfície devido a uma deformação mecânica aplicada ao corpo. O comportamento contrário também é válido: ao aplicar uma tensão elétrica na superfície do material piezométrico, este se deforma a fim de acomodar as cargas nele aplicadas. Como é ilustrado na Figura 4 (a), aplicando uma tensão contínua o material piezométrico se deforma de acordo com a polaridade da tensão, ou aplicando-se uma força suficiente para provocar deformação, e na Figura 4 (b) aplicando uma tensão alternada nesse mesmo material, ele vibra (ATCP Engenharia Física 2004).

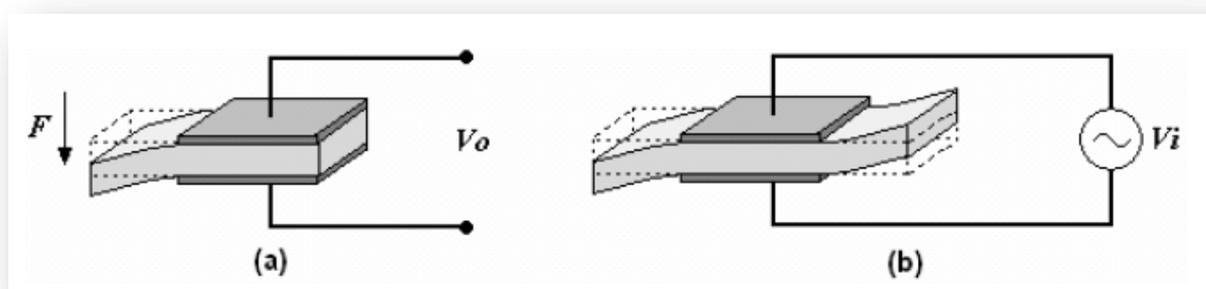


Figura 4 – Comportamento de material piezométrico

Fonte: (PINO e SOUZA 2011)

Os transdutores piezométricos são muito usados em *buzzers*, acelerômetros, sensores ultrassônicos de pressão, distância, de vibração e umidificadores de ar.

2.5. COMUNICAÇÃO SPI

SPI é a sigla para *Serial Peripheral Interface*, que é um protocolo síncrono de comunicação serial, tipicamente Mestre-Escravo. É um tipo de comunicação muito usada entre

dispositivos que necessita normalmente de quatro linhas, uma para *clock* (SCK – *Serial Clock*), outra de entrada de dados (SDI – *Serial Data In*), outra de saída de dados (SDO – *Serial Data Out*) e a última de seleção de dispositivos (CS\ - *Chip Select*). Durante a comunicação o sinal de *clock* é fornecido pelo mestre. E a cada *clock* o mestre deve colocar o *bit* desejado na linha SDO ou ler o *bit* desejado na linha SDI (CATSOULIS 2005).

2.6. COMUNICAÇÃO USB

USB é a sigla para *Universal Serial Bus*, que é um padrão de comunicação serial usado para comunicações entre computadores e outros dispositivos eletrônicos. Foi desenvolvido na década de 1990 por um grupo de sete empresas: *Compaq, DEC, IBM, Intel, Microsoft, NEC e Nortel*, com o intuito de facilitar o uso de periféricos, sem a necessidade de configuração manual e/ou reinicialização do sistema.

O barramento USB é feito através de quatro pinos até a versão 2.0 e nove pinos na versão 3.0, mas apresentam compatibilidade entre as versões. As versões 1.0 e 2.0 contêm quatro pinos no barramento, sendo dois para alimentação (5 V e GND) e dois para transferência de dados (D+ e D-) (AXELSON 2009, 425-438).

A não ser que o dispositivo tenha fonte de alimentação própria, ele pode ser alimentado pelos pinos de 5 V e GND, alimentação que geralmente é fornecida pelo computador. Entretanto, essa alimentação é limitada a 500 mA, e são fornecidos “pacotes” de 100 mA que devem ser solicitados pelo dispositivo.

A transferência de dados se dá pelo par diferencial D+ e D-, de maneira que o *bit* é escolhido no receptor fazendo a diferença entre esses dois sinais. Os cabos são trançados para ruído e interferência eletromagnética externa. Há uma limitação de 5 metros de cabo, entretanto esse limite pode ser superado com a utilização de *hubs* USB.

Hubs USB são dispositivos que expandem uma simples porta USB para mais portas, além de reajustar o sinal das perdas de transmissão pelo cabo e disponibilizar mais 5 metros de cabo para ligação de dispositivos.

A arquitetura das ligações entre dispositivos USB é feita em árvore, de maneira que existe um *host* que controla a comunicação. Isso impossibilitava a comunicação entre dispositivos sem a presença de um computador ou um dispositivo *host*. Isso mudou em 2006

com atualização para viabilizar a chamada *USB On the Go* que possibilitou a comunicação entre dispositivos sem a presença de um *host*.

A velocidade de operação da USB depende da versão que foi implementada. Na versão 1.0 existe a possibilidade de duas taxas de transmissão a chamada *Low Speed* com 1,5 Mbps e a chamada *Full Speed* com 12 Mbps. Na versão 2.0 foi denominada a *Hi-Speed* com velocidade de 480 Mbps. E na versão 3.0 a velocidade de transmissão denominada *Super Speed* é de 5 Gbps (GORTAN 2010).

Para concretizar a facilidade de uso de periféricos com o computador foram desenvolvidos *drivers* especiais para as classes de dispositivos a serem conectados num computador pessoal. Essas classes podem ser de armazenamento, Áudio, Imagem, Vídeo, comunicações, impressoras, *hubs*, HID (*Human Interface Device*), entre outras (USB-IF, Inc 2001).

Um dispositivo HID faz parte de uma classe genérica que está definido por um conjunto de documentos do *USB Implementers Forum's Device Working Group* (USB-IF, Inc 2001). Dentre estes estão teclados, *mouses*, *joysticks*, e controles para jogos. Além destes já citados, a especificação do HID é utilizada na construção de simuladores automotivos, dispositivos de telefonia, termômetros, controles de áudio e instrumentação médica.

2.7. CONSIDERAÇÕES

Através do estudo dos fundamentos teóricos que envolvem a microscopia de tunelamento, sobre os modos de operação de um microscópio, e do funcionamento de materiais piezelétricos e das informações que podem ser retiradas eles, podem ser estabelecidos os melhores parâmetros para o funcionamento do sistema. O modo de corrente constante foi escolhido para se realizar a aquisição dos dados utilizados na geração das imagens. As comunicações SPI e USB foram de extrema importância para agilizar o processo de desenvolvimento do microscópio, pois já eram conhecidas pela equipe.

3. ESCOPO

3.1. MERCADO CONCORRENTE

Dentre as empresas que atuam nesse mercado, destacam-se as empresas alemãs. A empresa alemã PHYWE comercializa um microscópio de tunelamento por varredura no valor de € 8000,00, aproximadamente R\$ 21000,00 para importação, com resolução de 8pm para os eixos XY e 4 pm (picômetros) para o eixo Z (PHYWE physics sem data). A empresa suíça Nanosurf possui um microscópio modelo easyScan™ STM no valor de US\$ 8500,00 com resolução de 3 pm (picômetros) para o eixo Z e 15 pm (picômetros) para o eixo XY. A empresa alemã OMICRON trabalha com o modelo UHV – STM 1 no valor de U\$ 200.000,00 com resolução atômica e uma garantia de 25 anos.

3.2. MERCADO CONSUMIDOR

Este projeto busca atender as necessidades de laboratórios de pesquisa e universidades buscando melhorias na análise de metais e crescimento tecnológico. Existem hoje no Brasil 2314 universidades (INEP – 2009).

3.3. ANÁLISE DE PATENTES

Dentro da classificação internacional de patentes, o projeto a ser desenvolvido se enquadra em:

Seção H: eletricidade.

01: Elementos elétricos básicos.

J: válvulas de descarga elétrica ou lâmpadas de descarga.

37: Válvulas de descarga com meios para introduzir objetos ou materiais a serem expostos à descarga.

As patentes encontradas são mostradas abaixo:

Título: Scanning Tunneling Microscope.

Data da patente: 3 de julho de 1990.

Inventores: Bando; Hiroshi (Tsukuba, JP), Sakai; Akira (Kawasaki, JP).

Código: Int. CI. H01J37/26

U.S. CI. 250/306, 250/307, 250/310, 250/311, 250/442

Site:

<http://www.google.com/patents?id=qwAnAAAAEBAJ&printsec=abstract&zoom=4#v=onepage&q&f=false>

Resumo: Um microscópio de varredura por tunelamento de acordo com a presente invenção compreende uma ponta emissores de elétrons spin-polarizada e um ímã para a aplicação de um campo magnético para esta ponta. Esta ponta é constituída de uma sonda de metal feita de um tungstênio e uma camada de cerca de 400 EUS. ANG. que é revestida na superfície da sonda de metal. Esta ponta é posicionada a uma distância tão próxima da superfície da amostra que a corrente flui do túnel. Um campo magnético é aplicado à direção axial da ponta uma tensão de polarização é aplicada entre a ponta ea amostra ea ponta é feita a varredura da superfície da amostra. A ponta de um campo magnético aplicado na direção axial do mesmo emite apenas spin-polarizada elétrons com spins do elétron orientado na direção axial. O spin-polarizada elétrons flui apenas na área na superfície da amostra onde os estados vazios para os elétrons com spins na direção paralela são distribuídos. Detectando o fluxo dos elétrons, ou seja, o túnel atual, é possível obter a distribuição dos elétrons com spins para baixo ou para cima. A forma da ponta do bico é quase a mesma que a de uma ponta usada em um STM comum. Isto torna possível para medir a distribuição dos spins dos elétrons com uma resolução em escala atômica.

Título: Scanning Tunneling Microscope

Data da patente: 15 de outubro de 1998.

Inventores: IWATSUKI MASASHI.

Código:

Internacional: G01B7/34; G01N37/00; G01Q30/04; G01Q60/10; G01Q60/12; H01J37/28;
(IPC1-7): G01B7/34; G01N37/00; H01J37/28

Site:

http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en_EP&FT=D&date=19990924&CC=JP&NR=11258249A&KC=A

Título: Scanning Tunneling Microscope

Data da patente: 16 de abril de 1996.

Inventores: KAINO MASATOMO.

Código:

Internacional: G01N37/00; G01Q60/10; G01Q60/16; H01J37/28; H01L21/66; (IPC1-7):
G01N37/00; H01J37/28; H01L21/66

Site:

http://worldwide.espacenet.com/publicationDetails/biblio?DB=EPODOC&adjacent=true&locale=en_EP&FT=D&date=19960416&CC=JP&NR=8101221A&KC=A

Título: Scanning microscopy using resonant quantum tunneling

Data da patente: 29 de janeiro de 2007.

Inventores: Flory; Curt A (Los Altos, CA).

Código:

Current U.S. Class: 250/306 ; 73/105

Current International Class: G01N 23/00 (20060101)

Site:

<http://patft.uspto.gov/netacgi/nph->

[Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetacgi%2FPTO%2Fsearch-bool.html&r=2&f=G&l=50&co1=AND&d=PTXT&s1=%22tunneling+scanning+microscope%22&OS="tunneling+scanning+microscope"&RS="tunneling+scanning+microscope"](http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO2&Sect2=HITOFF&p=1&u=%2Fnetacgi%2FPTO%2Fsearch-bool.html&r=2&f=G&l=50&co1=AND&d=PTXT&s1=%22tunneling+scanning+microscope%22&OS=)

3.4. CONSIDERAÇÕES

Na pesquisa de patentes foram encontradas 4 patentes relevantes, muito semelhantes ao projeto, que podem auxiliar no desenvolvimento.

Tanto a análise dos concorrentes quanto a pesquisa de patentes concluiu que não existe fabricante no Brasil de um produto igual ou semelhante ao que está sendo desenvolvido. Considerando que o produto tenha uma resolução semelhante aos que existem no mercado, percebe-se que o projeto aqui proposto possui mercado, principalmente sendo de um fabricante nacional, o que o tornaria mais barato que os importados.

4. DESENVOLVIMENTO

4.1. HARDWARE

O *hardware* do sistema é verificado em três dos quatro módulos: módulo de detecção de corrente de tunelamento, módulo de varredura XY, e módulo de digitalização e pré-processamento. Nesta seção serão tratados os circuitos utilizados no projeto de maneira separada. O esquemático completo pode ser visto nos Anexos A, B e C.

4.1.1. Módulo de detecção de corrente de tunelamento

Esse módulo é responsável por estabelecer as condições para a ocorrência de efeito túnel, bem como adquirir a corrente de tunelamento gerada para que possa ser processada posteriormente. Sendo constituído dos circuitos descritos nos itens 4.1.1.1, 4.1.1.2 e 4.1.1.3.

4.1.1.1. Transdutor Piezelétrico

O transdutor piezelétrico é um dos componentes mais caros de um STM, com o intuito de diminuir o custo do projeto, utilizou-se a idéia de John Alexander. ele sugere que para a movimentação da sonda, utilize-se um disco com material piezelétrico modificado.

O transdutor piezelétrico escolhido para o projeto é o mesmo disco usado para construção de *buzzers* comuns. Suas dimensões são de 22 mm de diâmetro de material piezelétrico sobreposto a um disco de 35 mm de diâmetro de metal. Ele será responsável pela movimentação da sonda. Como foi representado na Figura 5, o disco cerâmico foi cortado, com o auxílio de um bisturi, em quatro quadrantes (X+, X-, Y+, Y-), que representam os eixos a serem percorridos (Alexander 2003).

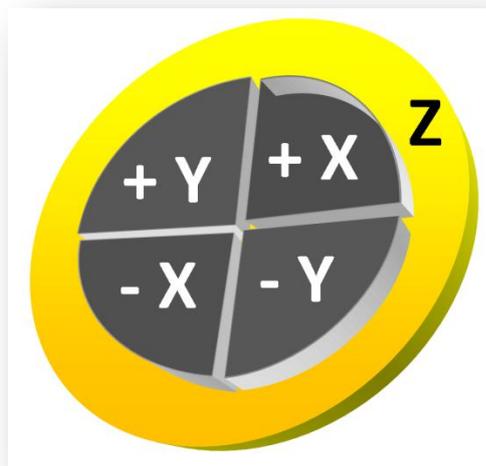


Figura 5 – Disco piezelétrico dividido em quadrantes

Fonte: Própria

A sonda foi fixada no meio do disco, exatamente entre os quatro quadrantes de maneira que um sinal aplicado entre $X+$ e $X-$ ou $Y+$ e $Y-$ faça com que a sonda se movimente no sentido dos eixos cartesianos X e Y . Na Figura 5, a tensão aplicada no disco metálico controla a altura da sonda (eixo cartesiano Z).

A tensão aplicada entre os quadrantes $X+$ e $X-$ ou $Y+$ e $Y-$ é de intensidade igual com polaridades opostas. O afastamento do ponto de repouso do sistema é diretamente proporcional à intensidade da tensão aplicada entre os quadrantes de um mesmo eixo, ou seja é proporcional a tensão aplicada no disco metálico.

Na Figura 6 é possível observar a estrutura do conjunto de movimentação com a sonda fixada e se perceber o tamanho a partir da comparação com uma moeda de um real. O resultado do conjunto de movimentação de sonda mostrado na figura 6 ficou muito parecido com o proposto por Alexander.

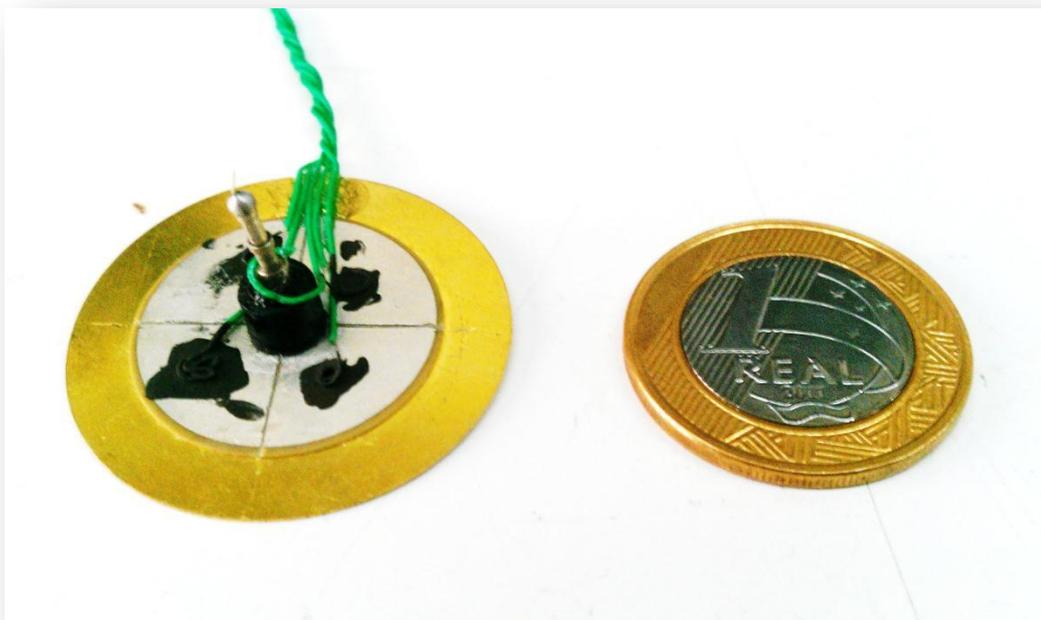


Figura 6 – conjunto de movimentação de sonda

Fonte: Própria

4.1.1.2. Buffer de Polarização

Constitui um amplificador operacional na configuração seguidor de tensão, com três resistores limitadores de corrente de aproximadamente 100 nA (Figura 7). Esse circuito é utilizado para polarizar a amostra e a sonda, de maneira a se atingir as condições necessárias para o efeito de tunelamento. Os resistores limitadores de corrente foram adicionados ao circuito, pois nos testes iniciais foram detectados problemas quando há contato entre a sonda e a amostra e, esse curto-circuito aquece a sonda que, por sua vez, se expande e o sistema não funciona de maneira adequada. Foram usados três resistores de 33 M Ω em série, pois houve grande dificuldade de encontrar no Brasil resistores de 100 M Ω .

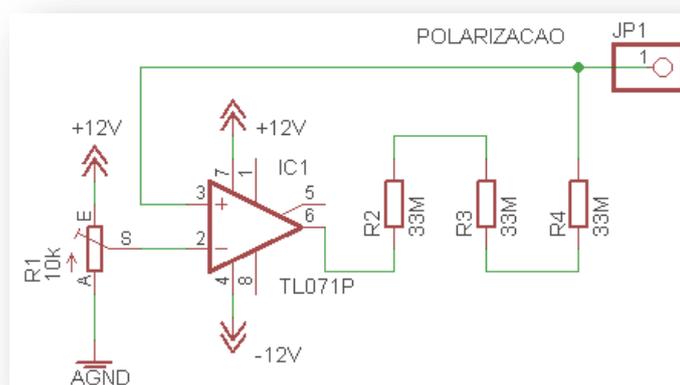


Figura 7 – Buffer de Polarização

Fonte: Própria

A saída de um seguidor de tensão é igual à tensão aplicada na entrada inversora. O circuito de polarização foi ajustado para uma tensão de saída de 70 mV. Esse valor da tensão foi escolhido para garantir que o efeito túnel influencie de maneira majoritária na formação da corrente. A saída do circuito de polarização (JP1) foi conectada à amostra.

4.1.1.3. Amplificador de Transimpedância

Esse circuito converte a corrente de tunelamento em uma tensão de saída. No circuito mostrado na Figura 8 é usado o amplificador LF411CN, com entradas e JFET (*Junction gate Field-Effect Transistor*) o que resulta numa alta impedância de entrada (10 TΩ). Esse amplificador também possui uma grande taxa de rejeição de modo comum, tipicamente de 100 dB (NATIONAL Semiconductor Corporation 2000). Esse fator é importante para diminuir o efeito causado por ruídos devido ao baixo valor do sinal de entrada. A corrente de tunelamento é tratada aqui como AMOSTRAZ e a saída como SCT (sinal da corrente de tunelamento).

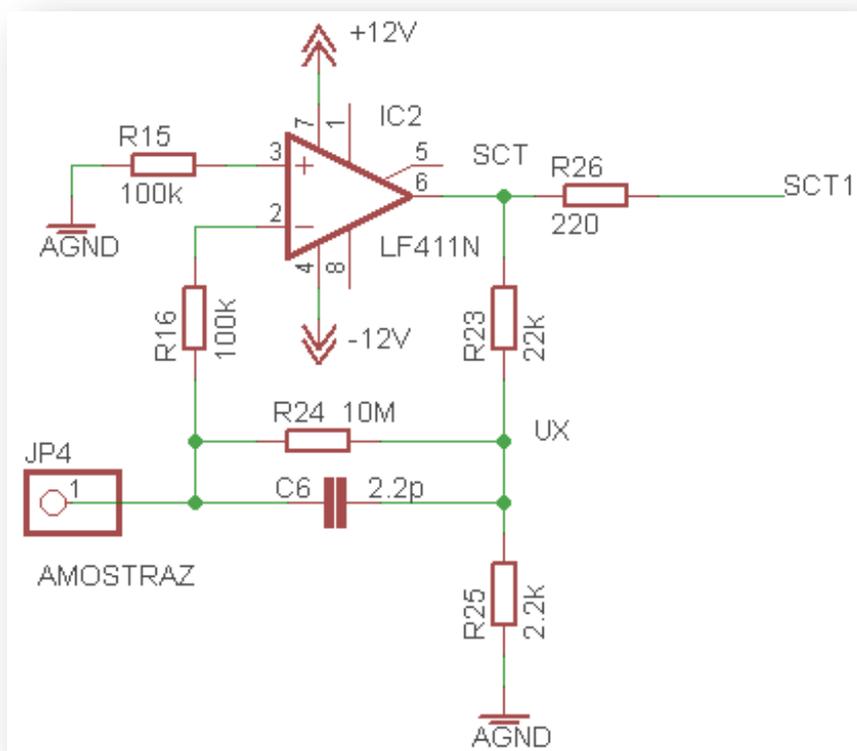


Figura 8 – Amplificador de transimpedância

Fonte: Própria

Através da teoria de amplificadores operacionais e análise nodal (primeira lei de Kirchhoff) o ponto de partida para o cálculo do ganho do amplificador de transimpedância são as equações:

$$U_x = -AMOSTRAZ \cdot R24 \quad (2)$$

$$\frac{U_x}{R24} = \frac{U_x}{R25} - \frac{SCT - U_x}{R23} \quad (3)$$

Substituindo a equação 2 na 3 e fazendo as manipulações necessárias chega-se à seguinte equação que relaciona a entrada AMOSTRAZ com a saída SCT:

$$\frac{SCT}{AMOSTRAZ} = -\frac{R23 \cdot R24}{R25} - R23 - R24 \quad (4)$$

Observando que $R24 \gg R23$ é possível desconsiderar a influência de $R23$ na fórmula, simplificando-a para:

$$\frac{SCT}{AMOSTRAZ} = -\frac{R24 \cdot (R23 + R25)}{R25} \quad (5)$$

Substituindo os valores escolhidos de $R23$, $R24$ e $R25$, o ganho deste amplificador de transimpedância é de 0,11 V/nA. Dessa forma a corrente de entrada ($AMOSTRAZ$) varia numa escala de aproximadamente 0 a 90 nA e a tensão de saída (SCT) varia de 0 V a 10 V. A entrada $AMOSTRAZ$ é ligada na sonda que foi fixada junto ao piezelétrico. A saída SCT é conectada na malha de controle de altura da sonda.

4.1.2. Módulo de varredura

Esse módulo é responsável por deslocar a sonda sobre a superfície da amostra nos eixos X e Y e o controle de altura (eixo Z), sendo dividido em duas partes: controle de posição e malha de controle.

4.1.2.1. Controle de posição

O controle de posição é feito através do conversor digital-analógico (D/A) MCP4822 que é conectado ao microcontrolador PIC18F4550 por um barramento de comunicação serial SPI, que será melhor discutida no item 4.1.6.

O MCP4822 é um conversor D/A com arquitetura de cadeia resistiva, dois canais independentes de saída e 12 *bits* de resolução. Cada canal controla um eixo de varredura, o canal A é usado para o eixo X e o canal B para o eixo Y. Os canais de saída do D/A tem variação de 0 V a 4,096 V. Esses sinais de saída são condicionados por dois estágios de amplificadores operacionais TL071, de modo que a tensão aplicada ao piezelétrico tenha

variação de -10 V a +10 V para cada eixo (X e Y). O primeiro estágio retira o nível DC e ajusta o ganho, esse sinal é aplicado no quadrante negativo. O segundo estágio funciona como inversor com ganho unitário e que é aplicado ao quadrante positivo. Essa variação maior aplicada ao piezelétrico possibilita área de varredura máxima.

Na *Figura 9* é possível ver o esquemático do circuito de controle de posição. Os sinais CS_XY, SCK, SDI e LDAC_XY são ligados no barramento de comunicação SPI. Já os sinais dos *jumpers* JP5, JP6, JP7 e JP8 estão conectados aos pontos X+, Y+, X- e Y-, respectivamente, da *Figura 5*.

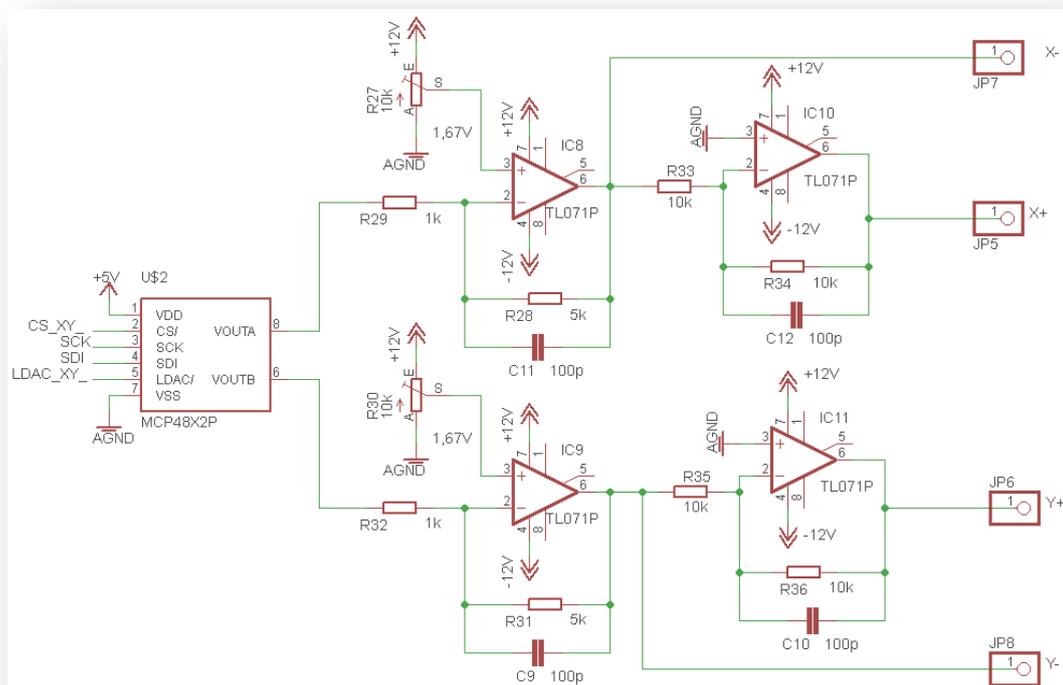


Figura 9 – Controle de posição

Fonte: Própria

O condicionamento dos sinais foi calculado conforme demonstrado a seguir. Considerando que a tensão de 4 V na saída dos canais A e B do conversor D/A equivale, na saída do estágio de condicionamento, a + 10 V e que quando a tensão for nula, ela corresponde a - 10 V. Observa-se que é necessário fornecer um ganho e alterar o *offset*. O ganho é dado pelos resistores R29, R28, R31 e R32, e o *offset* é dado pelos sinais VX e VY que vem dos potenciômetros R27, R30. Utilizando as teorias de amplificadores operacionais e a segunda lei de Kirchhoff (lei dos nós), segue:

$$V_{OUTX1} = \frac{V_X \cdot (R_{28} + R_{29}) - R_{28} \cdot V_{OUTA}}{R_{29}} \quad (6)$$

$$V_{OUTY1} = \frac{V_Y \cdot (R_{31} + R_{32}) - R_{31} \cdot V_{OUTB}}{R_{32}} \quad (7)$$

$$V_{OUTX2} = -V_{OUTX1} \quad (8)$$

$$V_{OUTY2} = -V_{OUTY1} \quad (9)$$

Substituindo os valores das tensões limite de entrada e saída, e fixando o valor de um dos resistores, é possível obter um sistema de equações de segunda ordem no qual as variáveis são V_X/V_Y e R_{28}/R_{31} .

Substituindo nas equações 6 e 7 os valores escolhidos para $R_{29} = R_{32} = 1 \text{ k}\Omega$, $V_{OUTA} = V_{OUTB} = 0 \text{ V}$ e $V_{OUTX1} = V_{OUTY1} = -10 \text{ V}$ têm-se:

$$V_X = -\frac{10000}{1000 - R_{28}} \quad (10)$$

$$V_Y = -\frac{10000}{1000 - R_{31}} \quad (11)$$

Substituindo nas equações 6 e 7 os valores escolhidos para $R_{29} = R_{32} = 1 \text{ k}\Omega$, $V_{OUTA} = V_{OUTB} = 4 \text{ V}$ e $V_{OUTX1} = V_{OUTY1} = +10 \text{ V}$ têm-se:

$$V_X = -\frac{10000 + 4 \cdot R_{28}}{1000 - R_{28}} \quad (12)$$

$$V_Y = -\frac{10000 + 4 \cdot R_{31}}{1000 - R_{31}} \quad (13)$$

Resolvendo os sistemas de equações formados pelas equações 10 e 12 para o eixo X e 11 e 13 para o eixo Y chega-se aos valores $V_X = V_Y = 1,67 \text{ V}$ e $R_{28} = R_{31} = 5 \text{ k}\Omega$.

4.1.2.2. Malha de controle PI analógica

Inicialmente foi projetada e implementada uma malha de controle PI (Proporcional-Integral) analógica, representada na Figura 10. A malha é constituída por um circuito subtrator e um circuito integrador, que subtrai do valor de SetPoint o valor de SCT1, e coloca o resultado na entrada do integrador. Depois de integrado, esse sinal passa a se chamar CONTROLEZ. O SET_POINT representa o valor de controle de SCT do sistema. A saída da malha de controle (sinal CONTROLEZ) é aplicada diretamente no piezelétrico. Este por sua vez controla a altura da sonda, alterando SCT de maneira que o mesmo se aproxime ao valor do SET_POINT.

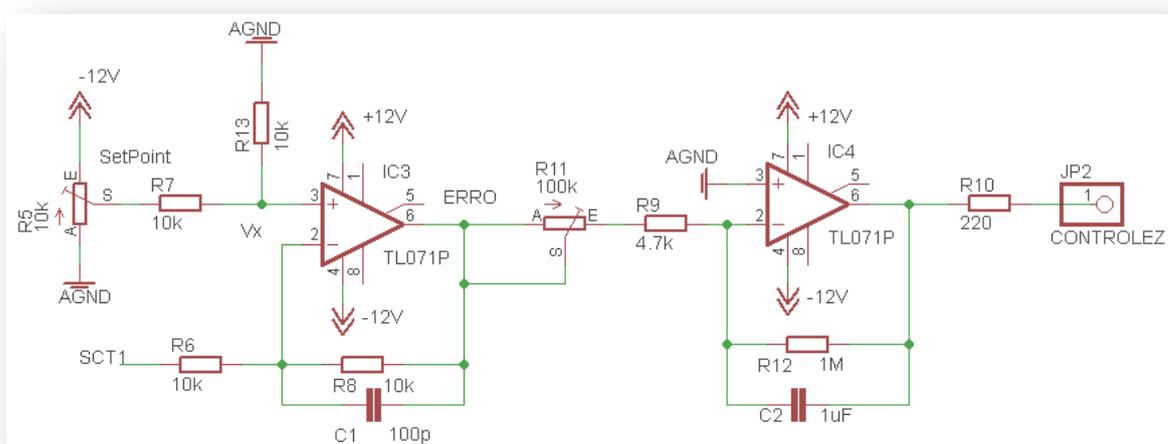


Figura 10 - Malha de controle PI analógica

Fonte: Própria

Devido à limitação da mudança de parâmetros do sistema dada pelo condicionamento de leitura da corrente de tunelamento SCT e do CONTROLEZ pelo sistema digital, que será comentado posteriormente, o SetPoint foi ajustado para uma tensão fixa de -6 V. A entrada SCT1 é conectada ao resistor de saída do amplificador de transimpedância. A saída CONTROLEZ é ligada no transdutor piezelétrico responsável pelo controle da altura.

Pela análise de circuito do primeiro estágio da malha de controle:

$$SetPoint = 2 \cdot V_x \quad (14)$$

$$\frac{V_x - SCT1}{R6} = \frac{ERRO - V_x}{R8} \quad (15)$$

Substituindo nas equações 14 e 15, $R6 = R8 = 10 \text{ k}\Omega$, $SetPoint = -6 \text{ V}$ e $SCT1$ tendo uma variação de -10 V a 0 V . Daí, tem-se que a variação do erro é descrita pela inequação a seguir: $-6 \text{ V} < ERRO < +4 \text{ V}$.

Analisando o segundo estágio, que é responsável pela integração do sinal da malha de controle, a constante de tempo τ é dada por:

$$\tau = (R9 + R11) \cdot C2 \quad (16)$$

Substituindo $R9 = 4,7 \text{ k}\Omega$, $C2 = 1 \text{ }\mu\text{F}$ e $R11$ tendo uma variação de 0 a $100 \text{ k}\Omega$ verifica-se que a variação a seguir: $4,7 \text{ ms} < \tau < 104,7 \text{ ms}$.

O período de integração foi ajustado para de $4,7 \text{ ms}$ para que o microscópio trabalhe em modo de corrente constante. Para a leitura do **CONTROLEZ** pelo sistema digital deve-se esperar um tempo suficiente para resposta do transdutor piezelétrico.

Entretanto, esse sistema apresentou certa dificuldade em conseguir manter o sinal **SCT** estável. Isso aliado à limitação de mudança de parâmetros da malha de controle, somado à complexidade de sintonia da malha **PI** analógica e principalmente por não se conhecer a função de transferência do piezelétrico e do sistema amostra-sonda, optou-se por testar um sistema de controle digital projetado a partir do método de discretização Bilinear, também conhecido como Tustin ou trapezoidal.

4.1.2.3. Malha de controle **PI** digital

Existem muitos métodos de discretização que podem ser aplicados a funções de transferência analógicas. Para este projeto foi escolhida a transformação bilinear, de Tustin ou trapezoidal, pelo motivo principal de apresentar um custo computacional baixo, simples e rápido. Este método é uma técnica padrão para a transformação de um sistema no domínio analógico (plano s) para o domínio digital (plano z). É fácil de se aplicar, pois se trata de uma

simples substituição de variáveis. Como a malha de controle analógica se apresentou estável, pela sua propriedade, a malha de controle digital aplicando este método de discretização também se mantém estável. Este método para transformação de s para z não apresenta *aliasing* (LIMA s.d.).

Através da análise de circuitos da malha de controle analógica, o controlador PI apresentado na *Figura 10* pode ser escrito da seguinte forma (considerando que $R = R11 + R9$ e $C = C2$):

$$\frac{SETPOINT - SCT(t)}{R} = \frac{\partial(CONTROLEZ(t))}{\partial t} \cdot C \quad (17)$$

Devido SCT ser um valor lido pelo A/D em uma resolução de 10 *bits* e o conversor D/A ser representado em um valor de 12 *bits*, a equação fica assim representada:

$$\frac{SETPOINT - SCT(t)}{R} = \frac{\partial(CONTROLEZ(t))}{\partial t} \cdot \frac{C}{4} \quad (18)$$

Aplicando a transformada de Laplace na equação 18, tem-se que:

$$CONTROLEZ(s) = 4 \cdot \left[\frac{SETPOINT - SCT(s)}{s \cdot C \cdot R} \right] \quad (19)$$

O método de discretização bilinear faz-se a simples substituição da variável s por, onde T representa o período de amostragem utilizado:

$$s = \frac{2 \cdot (z - 1)}{T \cdot (z + 1)} \quad (20)$$

Substituindo a equação 20 na 19, a função de transferência no plano z resulta na equação seguinte:

$$CONTROLEZ(z) = z \times CONTROLEZ(z) - \frac{2 \times T}{C \times R} \times [2 \times SETPOINT - (z + 1) \times SCT(z)] \quad (21)$$

Utilizando a transformada Z inversa na função de transferência representada pela equação 21, tem-se a representação no domínio do tempo discreto, onde k representa o instante de tempo discreto:

$$CONTROLEZ[k] = CONTROLEZ[k - 1] - \frac{2 \times T}{C \times R} \times (2 \times SETPOINT - SCT[k - 1] - SCT[k]) \quad (22)$$

A equação 22 foi implementada no microcontrolador para realizar o controle da altura da sonda.

4.1.3. Módulo de digitalização e pré-processamento.

Utilizando a malha de controle analógica, e o modo de operação de corrente constante, o sinal a ser adquirido para a formação da imagem é aquele que é aplicado no piezelétrico (CONTROLEZ - Figura 9). O condicionamento desse sinal é feito por dois amplificadores operacionais em configuração inversora: o primeiro inverte o sinal e o segundo fornece um ganho e adiciona um *offset* para que o sinal fique entre 0 V e 5 V. O sinal CONTROLEZ é a saída do estágio integrador da malha de controle PI e tem uma variação de 20 V (de -10 V a +10 V). A Figura 11 mostra o circuito de condicionamento do sinal CONTROLEZ.

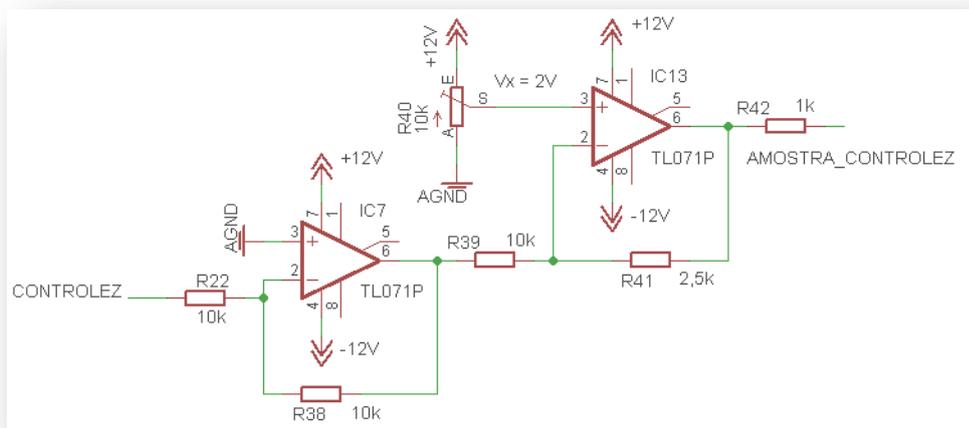


Figura 11 - Condicionamento do sinal CONTROLEZ

Fonte: Própria

Com os parâmetros supracitados e escolhendo $R22=R38=R39=10\text{ k}\Omega$, tem-se:

$$AMOSTRA_CONTROLEZ = CONTROLEZ \cdot \left(\frac{R38}{R22}\right) \cdot \left(\frac{R41}{R39}\right) + V_x \cdot \left(1 + \frac{R41}{R38}\right) \quad (23)$$

Substituindo-se os valores, montando o sistema de equações e resolvendo-o, chega-se a:
 $V_x = 2\text{ V}$ e $R41 = 2,5\text{ k}\Omega$.

Feito o condicionamento do sinal, segue-se para a sua amostragem. O conversor analógico-digital (A/D) utilizado está integrado no microcontrolador PIC 18F4550 e utiliza o método de digitalização de aproximações sucessivas com uma resolução de 10 *bits*.

Entretanto, como já discutido, esse sistema não se mostrou flexível em relação à sintonização da malha de controle e pouco estável, portanto decidiu-se implementar uma malha de controle digital.

Para a substituição da malha de controle analógica pela digital, se fez necessário condicionar o sinal SCT com um amplificador operacional para que pudesse ser amostrado pelo conversor A/D. O sinal de saída do controlador PI digital também teve que ser condicionado por um D/A e por um estágio de amplificador operacional para ser aplicado no piezelétrico.

A *Figura 12* apresenta os circuitos utilizados para aquisição (a) e entrega do sinal processado (b).

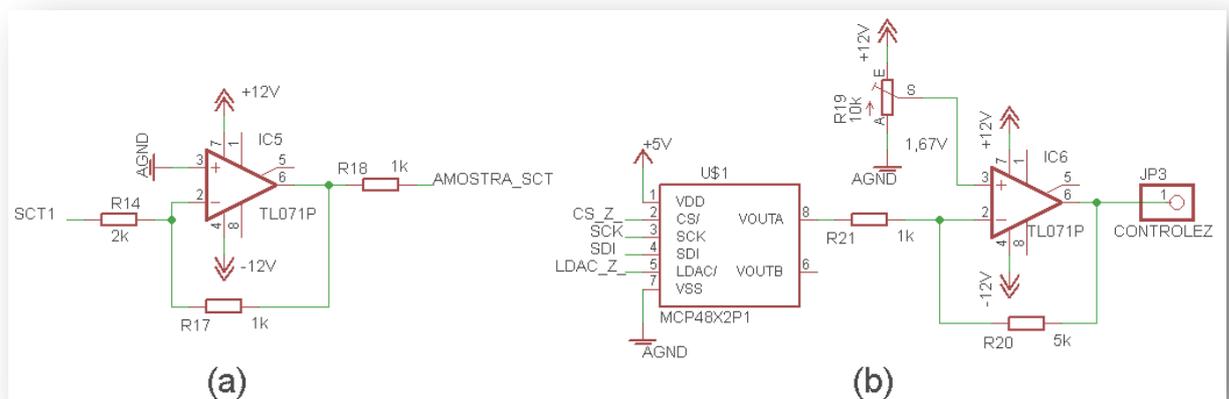


Figura 12 - (a) aquisição do sinal SCT1 (b) entrega do sinal processado.

Fonte: Própria

O conversor A/D tem a capacidade de discretizar sinais com variações entre 0 V e 5 V, portanto é preciso apenas reduzir SCT1 à metade e inverter seu sinal. Para isso usa-se um simples amplificador inversor com ganho 0,5 V/V com relação a SCT1 que possui uma variação de 0 V a -10 V, como mostrado na Figura 12(a). E o circuito da Figura 12(b) condiciona o sinal de saída ao piezelétrico da mesma maneira descrita na seção 4.1.2.1, uma vez que os requisitos são os mesmos.

4.1.4. MICROCONTROLADOR PIC 18F4550

O microcontrolador PIC18F4550 foi escolhido pelo conhecimento da arquitetura e linguagem de programação já adquirida em projetos anteriores pela equipe, além da disponibilidade desse componente no mercado local. A experiência tornou mais fácil e ágil o desenvolvimento do bloco em que foi implementado esse componente.

O PIC18F4550 é um microcontrolador de arquitetura RISC (*Reduced Instruction Set Computer*) de 8 bits com 40 pinos. Possui uma enorme lista de funcionalidades, dentre as quais vale destacar para esse projeto (MICROCHIP Technology Inc. 2004):

- 32 *kbytes* de memória de programa,
- 2 *kbytes* de memória SRAM,
- 256 *bytes* de EEPROM,
- 4 *timers*, sendo 1 de 8 bits e 3 de 16 bits,
- *Hardware* integrado para USB 1.1,
- *Hardware* integrado para SPI,
- Até 48 MHz de frequência de operação,
- 13 canais do conversor A/D de 10 bits.

A Figura 13 mostra o esquemático usado para o funcionamento do PIC 18F4550. Nela é possível observar o circuito oscilador que fornece o sinal de *clock*, o circuito de *reset* e a alimentação do microcontrolador.

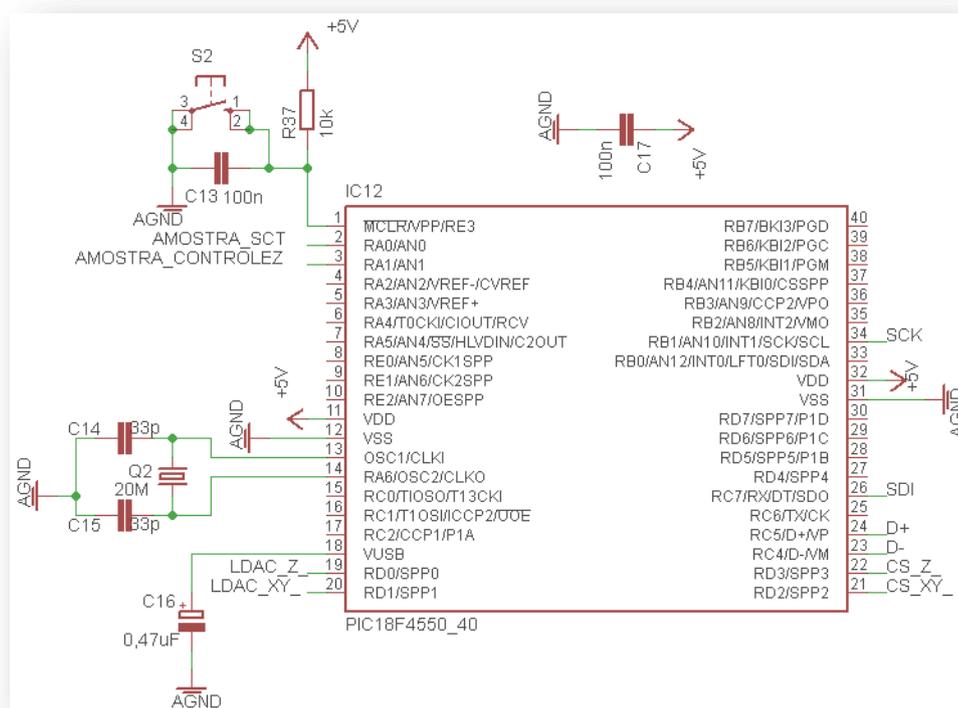


Figura 13 - Microcontrolador PIC 18F4550

Fonte: Própria

O cristal escolhido é de 20 MHz, que gera para o microcontrolador um *clock* interno de 5 MHz ou 200 ns de período de *clock*. Entretanto, essa velocidade se mostrou insuficiente para atender todas as funcionalidades necessárias ao projeto de maneira que foi necessário ativar o PLL (*phase locked loop*) para multiplicar a frequência, atingindo 48 MHz. Essa frequência gera um *clock* interno de 12 MHz ou 83,3 ns de período de *clock* ao PIC.

O circuito de *reset* é de extrema importância para o funcionamento de qualquer sistema microcontrolado. No instante em que o circuito é ligado o capacitor C13 está descarregado e, portanto o pino de *reset* do PIC esta em nível lógico 0 e como é um pino barrado o processador é “resetado”. O controlador se manterá em *reset* até que o capacitor se carregue atingindo nível lógico 1 no pino de *reset*. Isso ocorre porque flui uma corrente pelo resistor R37 que carrega C13. Ao fechar a chave S2 joga-se em nível lógico 0 o pino de *reset* e o capacitor se descarrega. E o *reset* acontece até que o capacitor se carregue novamente.

4.1.5. CONVERSOR DIGITAL ANALÓGICO MCP4822

O conversor digital analógico MCP4822 foi escolhido pela experiência da equipe com esse componente em projetos anteriores, o que facilitou e agilizou o desenvolvimento dos blocos em que foi implementado. Outros fatores que contribuíram para a escolha deste componente foi a facilidade de acesso e um preço reduzido comparado a outros conversores D/A.

O conversor digital analógico MCP4822 é um circuito integrado de 8 pinos da Microchip, mesmo fabricante do microcontrolador PIC18F4550. Ele possui 12 *bits* de resolução com tensão de referência interna e interface de comunicação SPI. Possui dois canais de saída (canal A e canal B) *rail-to-rail*. A comunicação SPI suporta um *clock* de 20 MHz. Possui um tempo de resposta rápida de 4,5 μ s. Sua tensão de operação é entre 2,7 V a 5,5 V. Por isso houve a necessidade de fazer o condicionamento do sinal de saída para o piezelétrico para se trabalhar em uma tensão variando entre -10 V a +10 V.

Os DACs da linha MCP482X (*Figura 14*) têm por características a alta precisão e grande desempenho em relação a ruído para aplicações industriais onde a calibração e a precisão de sinais são requeridos. Utiliza uma arquitetura de cadeia resistiva. Ele possui dois *buffers* de registradores, que podem ser carregados simultaneamente através do pino LDAC/.

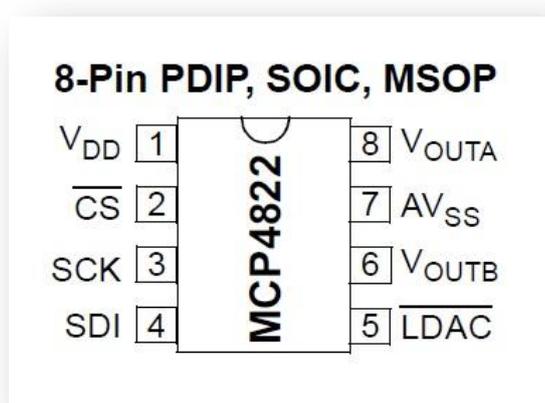


Figura 14 - Pinagem do Conversor Digital-Analógico MCP4822

Fonte: (MICROCHIP Technology Inc. 2010)

Como pode ser visto na *Figura 15*, o ciclo de leitura da MCP4822 é formado por 16 *bits*, compostos por 4 *bits* de comando e 12 *bits* de dados. Nos *bits* de comando é possível configurar

a seleção de escrita no *buffer* (*buffer A* para o canal A ou *buffer B* para o canal B), o ganho da saída, habilitar ou desabilitar a saída do D/A (MICROCHIP Technology Inc. 2010).

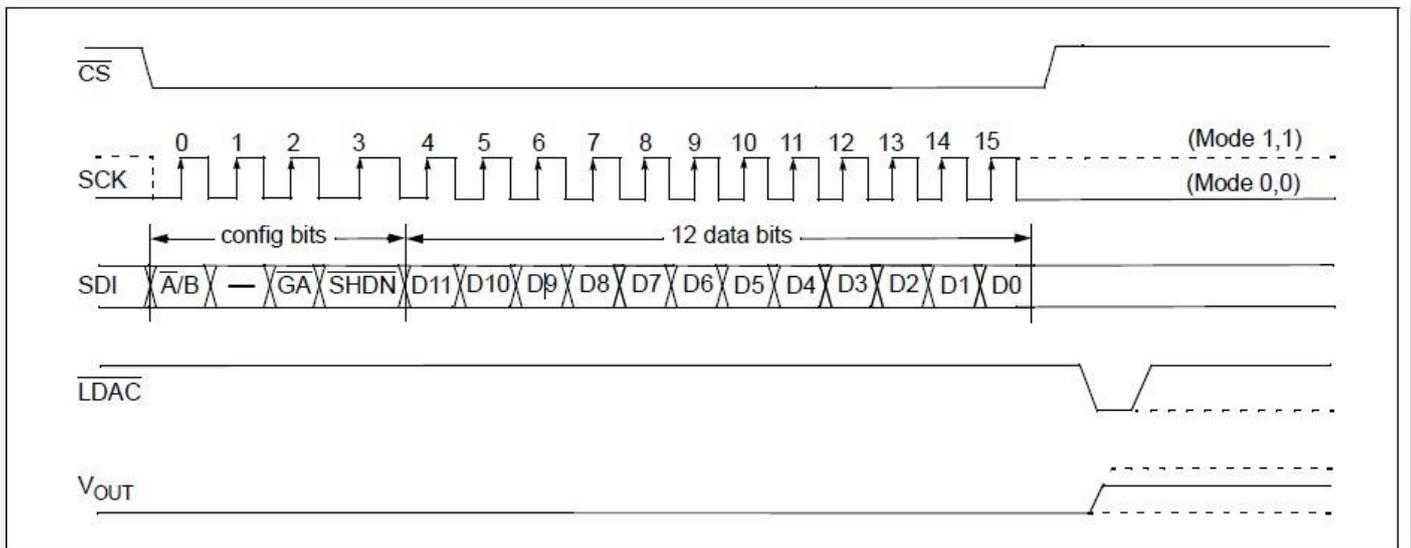


Figura 15 – Ciclo de leitura do MCP4822

Fonte: (MICROCHIP Technology Inc. 2010)

4.1.6. COMUNICAÇÃO SPI

O PIC 18F4550 possui *hardware* integrado para comunicação SPI, então se faz necessária apenas a ligação física dos pinos referentes à comunicação SPI (SCK, SDO e CS), além de um pino denominado LDAC que atualiza as saídas do MCP4822. As conexões podem ser observadas nos anexos A e C.

4.1.7. COMUNICAÇÃO USB

O PIC 18F4550 possui *hardware* integrado para comunicação USB, dessa forma a implementação de *hardware* a ser feita é conectar os quatro terminais do PIC referentes a USB (D+, D-, V_{usb} e GND) no conector USB. Essas conexões podem ser vistas na Figura 16.

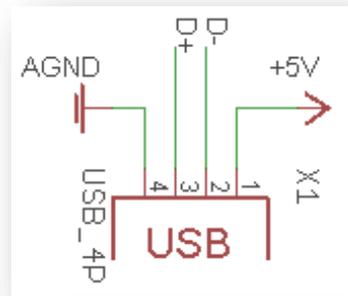


Figura 16 - conexão USB

Fonte: Própria

4.2. FIRMWARE

O *firmware* foi desenvolvido baseado no microcontrolador PIC18F4550 da *Microchip*. A linguagem de programação escolhida foi a linguagem C pelo fato de ser compatível com o *software* de programação. O *software* utilizado para programação foi o MikroC da MikroElektronika devido à facilidade de uso e bibliotecas prontas que agilizaram a programação. Após a programação do *firmware* no MikroC, o próprio *software* compila o código gerando um arquivo com extensão .hex. Com este arquivo, com o auxílio de um gravador da LabTools chamado ICD2BR e um *software* da própria Microchip denominado MpLab IDE, foi gravado o *firmware* internamente ao microcontrolador PIC18F4550.

Para a geração do código fonte da USB foi utilizado um aplicativo do MikroC chamado HID terminal. O código gerado foi implementado junto ao código do *firmware*.

Para o projeto STM foram desenvolvidos dois *firmwares* distintos, um para implementar o circuito com uma malha de controle analógico, e outro para a malha de controle digital.

4.2.1 MALHA DE CONTROLE ANALÓGICA

Para o controle da malha de controle analógico foram utilizados os seguintes periféricos integrados do microcontrolador:

- USB, para transmitir as informações de posição dos eixos X, Y e Z para o *software*.
- SPI, para fazer a comunicação com o MCP4822 responsável pela conversão D/A.
- Um canal (AN1) do conversor A/D de 10 *bits* para fazer a leitura do CONTROLEZ.

Na *Figura 17* é mostrado o fluxograma do *firmware* para a malha de controle analógica.

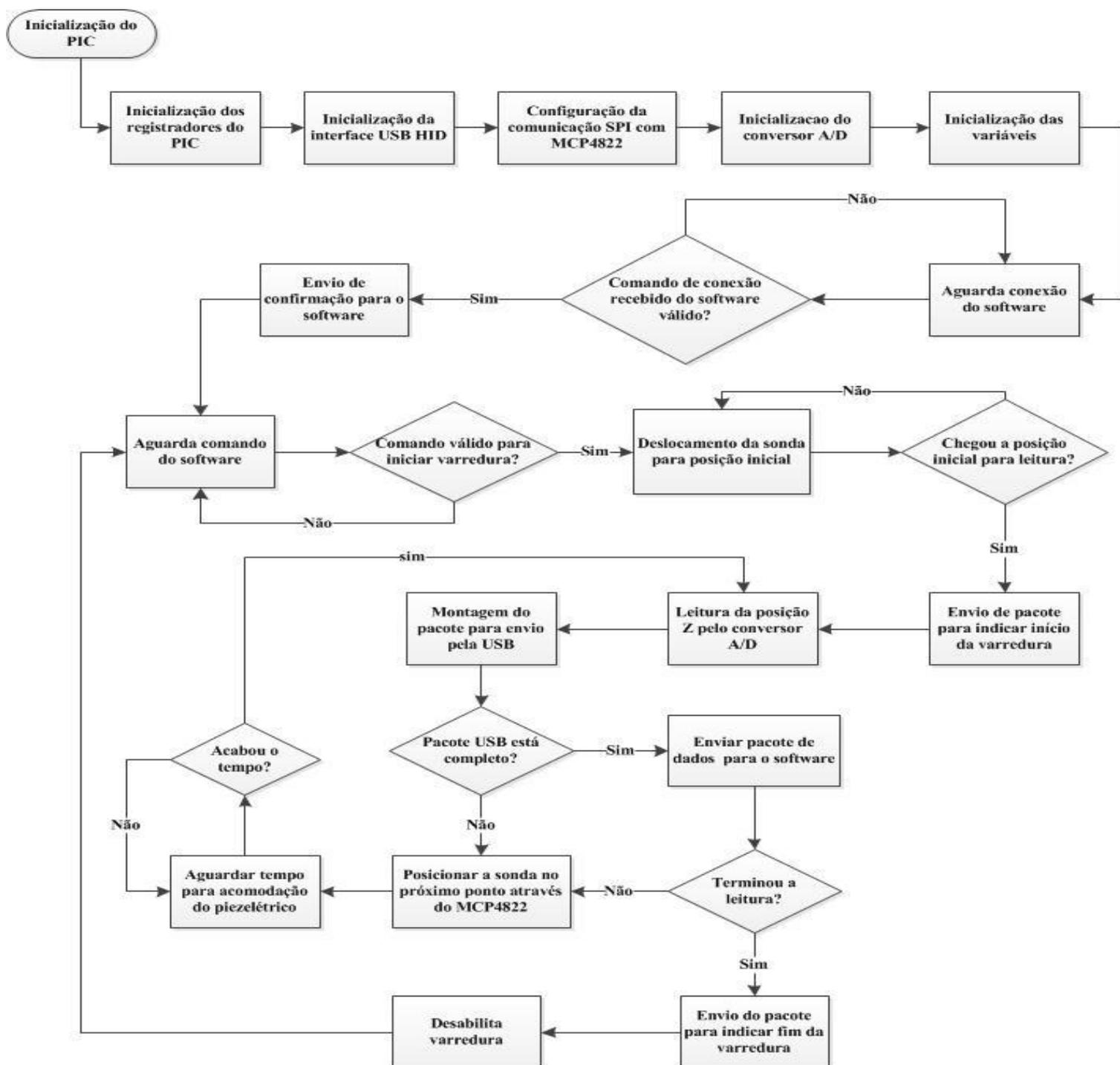


Figura 17- Fluxograma do firmware para o uso da malha de controle analógico

Fonte: Própria

A inicialização do microcontrolador PIC18F4550 foi feita através do *software* de programação MikroC, que ao salvar o novo projeto apresenta uma janela para configuração inicial. Através dela é possível configurar o sinal de *clock*, as formas do *reset*, proteção da memória, etc.

No bloco “INICIALIZAÇÃO DOS REGISTRADORES DO PIC” foram configuradas as portas de entrada e saída, desabilitação dos comparadores e habilitação geral das interrupções.

Para a inicialização da interface USB HID foi gerado um código pelo aplicativo HID terminal do *software* MikroC, com as configurações da USB necessárias para a comunicação. Esse código foi integrado ao *firmware*. Foram definidos dois *buffers* chamados “readbuff” e “writebuff”, um para leitura e outro para escrita respectivamente, com o tamanho de 64 *bytes* cada.

A configuração da comunicação SPI foi estabelecida da seguinte forma: foram ativados os pinos referentes ao sistema e através da função “SPI1_Init()” da biblioteca do MikroC foram salvos os registradores, tudo de maneira transparente ao programador.

Como dito anteriormente, o conversor A/D está integrado com o microcontrolador. Para sua configuração foi utilizada a biblioteca do MikroC, utilizando a função chamada “config_AD()”.

No bloco “Inicialização das variáveis” foram registrados os valores iniciais das variáveis globais criadas no *firmware*.

Em relação à comunicação USB, o microcontrolador sempre trabalha na configuração escravo. Dessa forma o PC é sempre quem inicializa a comunicação enviando um pacote de dados padrão para o microprocessador.

No bloco “aguarda conexão do *software*”, o sistema espera receber um protocolo válido que será discutido posteriormente. Após receber o pacote, o microcontrolador envia uma resposta de confirmação.

A partir de então o sistema está pronto para receber o comando de inicialização da varredura. Junto com este comando podem ser enviados pelo *software* os parâmetros: posição inicial de varredura de X e Y, posição final de varredura x e y, o tempo de amostra do CONTROLEZ, passo entre uma amostra e outra e a quantidade de valores lidos pelo A/D para o cálculo das médias.

O próximo passo é movimentar a sonda para a posição inicial. Após chegar à posição definida, o *firmware* envia um pacote para *software* indicando o início da leitura.

A leitura é feita pelo canal A/D 1 (AN1) do microcontrolador, tendo a possibilidade de ler várias vezes o mesmo ponto e fazendo a média deles. A leitura realizada pelo A/D é a informação do CONTROLEZ. Após esse procedimento, o pacote contendo as informações relativas a posição X, Y e a altura Z é construído no *buffer*. O pacote é capaz de transferir a informação de até 10 pontos diferentes. Como as posições X, Y e Z possuem um tamanho de 2 *bytes* cada (16 *bits* de informação), cada ponto é formado por 6 *bytes*. A estrutura do pacote para o envio via USB é de 60 *bytes*, ou seja, esse pacote contém informações tridimensionais de 10 pontos.

Após a leitura de um ponto, conhecendo sua posição x, y e z, a informação é salva no *buffer*. O *firmware* então envia um comando para o MCP4822 de maneira que posicione a próxima posição da sonda. Logo após o posicionamento, o sistema aguarda um tempo determinado por um parâmetro mencionado anteriormente, para que haja um tempo de acomodação do transdutor piezométrico e resposta do controlador PI. Em seguida uma nova leitura é realizada. Depois de finalizada a leitura de 10 pontos seguidos, o *firmware* finalmente envia o pacote pela USB para o *software*, e retomando todo o processo descrito acima novamente até o último ponto.

4.2.2 MALHA DE CONTROLE DIGITAL

Para o controle da malha de controle digital foram utilizados os seguintes periféricos integrados do microcontrolador:

- USB, para transmitir as informações de posição dos eixos cartesianos X, Y e Z para o *software*.
- SPI, na qual foram utilizados para comunicar com dois conversores MCP4822, com o barramento sendo multiplexado entre os dois *chips*.
- Canal 0 (AN0) do conversor A/D de 10 *bits* para fazer a leitura do sinal SCT.

Na Figura 18 é mostrado o fluxograma do *firmware* para a malha de controle digital:

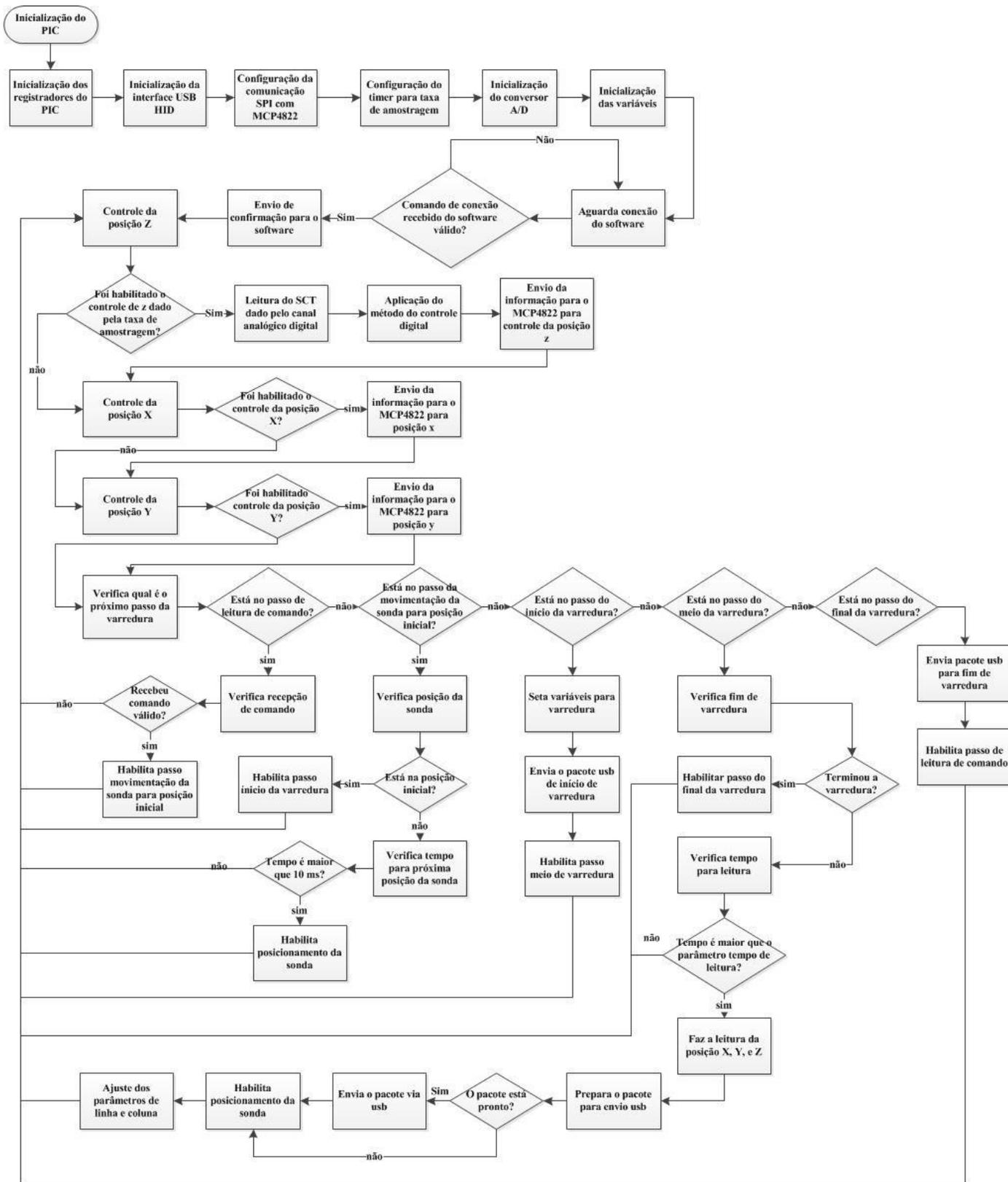


Figura 18- Fluxograma do controle digital

Fonte: Própria

Por razão da limitação imposta pelo período entre uma amostragem e outra, o sistema sofreu algumas alterações em relação ao sistema de controle analógico. A inicialização do microcontrolador foi desenvolvida de maneira semelhante à do controle analógico, com uma diferença na configuração da comunicação SPI. A leitura do SCT, que é feita pelo canal 0 (AN0, pino 2 – ver Figura 13), foi adicionada à função de configuração do *timer 2* e o sinal de *clock* do processador foi alterado para 48 MHz.

A comunicação SPI é multiplexada pelos dois MCP4822, sendo um deles responsável pelo controle da posição nos planos cartesianos X e Y, e o outro utilizado para fazer o controle no plano Z. Para que não houvesse conflito de comunicação entre os dois conversores D/A, o código foi desenvolvido para operar pelo método de *polling*.

Para diminuir o tempo de processamento do *firmware*, o código-fonte foi escrito de forma sequencial, sem a utilização de funções ou estruturas que façam o processador ser obrigado a realizar chaveamentos de contexto – onde se salva os registradores de uso geral e alguns registradores de funções especiais (SFR's – *Special Function Registers*).

A leitura do SCT é feita pelo conversor D/A de 10 *bits*, canal 0 do microcontrolador (ver Figura 13). A configuração é feita na função “*config_AD()*”, e é realizada manualmente para deixar a função de leitura mais rápida quanto possível.

O *timer 2* foi configurado para gerar uma interrupção conforme a taxa de amostragem calculada pela seguinte fórmula:

$$taxa_amostragem = \frac{1 \cdot 4 \cdot PR2 \cdot prescaler \cdot postscaler}{48000000} \quad (24)$$

Onde PR2 é um registrador de carga do *timer* que varia entre 0 e 255, *prescaler* e *postscaler* são constantes de multiplicação.

Para este projeto foi escolhida uma taxa de amostragem de 500 us, com *prescaler* igual a 16 e *postscaler* com valor 2 e PR2 com valor 188. Todas estas configurações foram feitas na função “*config_timer2()*”. O *timer 2* foi configurado para gerar uma interrupção que a cada chamada habilita a malha de controle da altura Z. Como foi dito anteriormente, por razão do código estar construído baseado no método de *polling*, a taxa de amostragem da malha de controle Z pode variar ligeiramente em relação ao definido pela equação 24 dependendo do tempo de processamento de todo o sistema.

O *firmware* pode ser dividido em 8 partes: controle Z, controle X, controle Y, recepção dos dados recebido pelo *software* através da comunicação USB, posicionamento da sonda no ponto inicial de varredura, início da varredura, varredura e final da varredura.

O controle da posição Z é habilitado pela interrupção do *timer 2* dada pela taxa de amostragem. Nela é feita a leitura do sinal SCT pelo canal 0 do conversor D/A. O controle foi desenvolvido a partir do método de discretização bilinear. Ele calcula o valor a ser transferido – pela interface SPI – para o conversor D/A do MCP4822 responsável pelo controle Z.

O controle da posição X e o controle da posição Y são atualizados a partir da comparação com o valor modificado pelo sistema de varredura, e o valor configurado anteriormente. Esta informação é transferida para o conversor D/A do MCP4822 responsável pela varredura nos dois eixos.

O *firmware* aguarda receber um comando dado pelo *software* através da comunicação USB. Ao receber um comando válido, um *flag* é alterado para realizar o posicionamento da sonda no ponto inicial. Isto é feito alterando as variáveis de posição X e Y.

Após o posicionamento da sonda, o *flag* é habilitado para enviar o primeiro pacote para o *software* indicando o início da varredura, e são configuradas as variáveis de controle de comunicação entre *firmware* e *software*. Após esta etapa, o *flag* é configurado para o próximo passo que é a varredura propriamente dita. Esta etapa é semelhante ao controle analógico, com a diferença que a informação no controle da posição Z não é o aquele lido pelo conversor A/D e sim a partir do mesmo registrado pelo sistema de controle digital. Após o término da varredura, o *flag* é configurado para o próximo passo que é o fim da varredura, que envia o pacote indicando o fim da leitura. Após o envio do pacote o *flag* é configurado para aguardar um comando do *software* novamente.

4.3. PROTOCOLO DE COMUNICACAO USB ENTRE *FIRMWARE* E *SOFTWARE*

Para a USB foram definidas o *Vendor ID* como 0x1234 (hexadecimal) e o *Product ID* como 0x0001 (hexadecimal).

Para a comunicação entre *software* e *firmware* foram definidos protocolos de comunicação, explicados no item 4.3.2. Cada caractere representa um *byte* do pacote.

4.3.1 PROTOCOLO DE CONEXÃO

O *software* envia comando `readbuff = [0x73]` formado por apenas 1 *byte*. Enquanto essa informação não for recebida pelo *firmware*, este não inicializa o funcionamento do sistema. Recebendo o dado correto, o *firmware* envia a resposta com a estrutura `writebuff = [73 "B" "I" "O" "T" "A" 73]`, que possui 7 *bytes*.

4.3.2. PROTOCOLO PARA INICIAR A VARREDURA

Para o protocolo de início de varredura existem dois tipos de pacotes. No primeiro caso estão inclusos parâmetros de varredura posição inicial x, posição inicial y, posição final x, posição final y, tempo de leitura entre uma amostra e outra, tamanho do passo, quantidade de amostras da posição z. No segundo caso não é enviado nenhum parâmetro utilizando a configuração da varredura anterior, enviando apenas o comando para iniciar a varredura. Todos os parâmetros possuem 16 *bits*.

O pacote para o primeiro caso com passagens de parâmetro possui 18 *bytes* com o formato:

`readbuff = [0x73 0x0FF A B C D E F G H I J K L M N]`, onde:

- A representa a posição inicial no eixo X (*byte* mais significativo),
- B representa a posição inicial no eixo X (*byte* menos significativo),
- C representa a posição final no eixo X (*byte* mais significativo),
- D representa a posição final no eixo X (*byte* menos significativo),
- E representa a posição inicial no eixo Y (*byte* mais significativo),
- F representa a posição inicial no eixo Y (*byte* menos significativo),
- G representa a posição final no eixo Y (*byte* mais significativo),
- H representa a posição final no eixo Y (*byte* menos significativo),
- I representa o passo a ser aplicado à sonda (*byte* mais significativo),
- J representa o passo a ser aplicado à sonda (*byte* menos significativo),
- K representa o tempo entre amostras (*byte* mais significativo),
- L representa o tempo entre amostras (*byte* menos significativo),
- M representa a quantidade de amostras (*byte* mais significativo),
- N representa a quantidade de amostras (*byte* menos significativo).

O pacote sem passagens de parâmetros possui 2 *bytes* com o formato, onde X é qualquer *byte* diferente de 0xFF:

readbuff = [0x73 X]

4.3.3 PROTOCOLO PARA INDICAR INÍCIO DA VARREDURA

Após a habilitação para iniciar a varredura enviada pelo *software*, o sistema ajusta a sonda para a posição inicial. Depois de estar posicionado, o *firmware* envia um pacote para o *software* indicando o início da varredura. Este pacote possui 64 *bytes*, com a seguinte estrutura:

writebuff = [A A A A]

A representa um *byte* fixado no valor 128 em base decimal.

4.3.4 ESTRUTURAÇÃO DO PACOTE DE DADOS

O pacote possui 64 *bytes* de dados para a geração da imagem e tem a seguinte estrutura:

writebuff = [X₁ Y₁ Z₁ X₂ Y₂ Z₂ X₁₀ Y₁₀ Z₁₀ A B C]

X representa a posição no eixo X (2 *bytes*),

Y representa a posição no eixo Y (2 *bytes*),

Z representa a posição no eixo Z (2 *bytes*),

A informa o número do pacote (23 *bits*),

B informa a inversão de coluna (1 *bit*),

C está reservado para verificação de erro (1 *byte*).

4.3.5 PROTOCOLO PARA INDICAR FINAL DA VARREDURA

Após o término da varredura, o *firmware* envia um pacote para o *software* indicando o final da varredura.

Este pacote possui 64 *bytes* de informação com a seguinte estrutura:

writebuff = [A A A A],

onde, A representa um *byte* fixado no valor 150 em base decimal.

4.4. SOFTWARE

Para se desenvolver a interconectividade com o ambiente externo ao computador – mais especificamente o microcontrolador e por sua vez o sistema eletrônico sensor – foi realizada a pesquisa de diversas tecnologias existentes atualmente que poderiam fornecer soluções úteis ao projeto. Um dos critérios de avaliação das mesmas era que permitissem a troca de informações num ambiente construído a partir de soluções modularizadas. A *Figura 18* mostra esse ambiente dividido em módulos conforme citado anteriormente.

Durante esse processo, chegou-se à conclusão que a tecnologia da plataforma .NET atenderia as necessidades do projeto. Algumas vantagens são: uma extensa documentação disponível, flexibilidade do código para ser integrado com outras linguagens e *softwares* externos, facilidade de uso e suporte *online* por parte da Microsoft. Outro motivo para a escolha dessa plataforma foi a possibilidade da integração com módulos construídos no MATLAB, da MathWorks, compilados como componentes .NET, através do MATLAB Compiler.

Na *Figura 19* é possível observar todos os módulos envolvidos no processamento. Nos próximos parágrafos cada módulo será descrito detalhadamente.

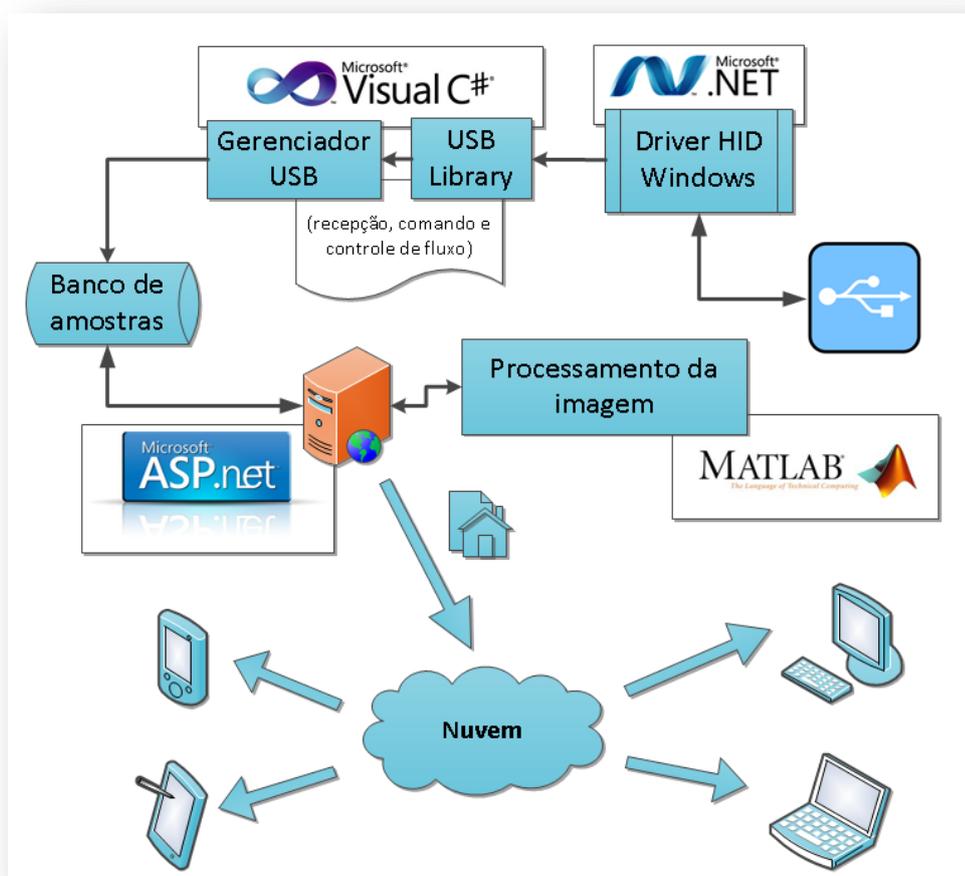


Figura 19 – Visão geral do componente de software

Fonte: Própria

A necessidade de comunicação com o microcontrolador foi atendida com o uso de uma porta USB utilizando o *driver* de interface humana (HID, do inglês *Human Interface Device*). Este tipo de dispositivo faz parte de uma classe genérica que está definido por um conjunto de documentos do *USB Implementers Forum's Device Working Group* (USB-IF, Inc 2001). Para mais detalhes, há uma descrição completa na seção 2.6. A escolha desta classe como interface no projeto se deve à facilidade e ampla disponibilidade de recursos de auxílio ao desenvolvimento de soluções através dela.

A comunicação foi o primeiro passo no desenvolvimento do *software*. Resolver essa etapa permitiria entender e discutir melhor como seria implementado o protocolo de comunicação utilizado para a troca de informações em ambos os sentidos, a saber, do computador para o microcontrolador e o seu inverso.

A biblioteca USB – no diagrama de blocos chamada de “USB Library” – foi disponibilizada através do site Codeproject.com (WIMAR 2007) em um exemplo de implementação da rotina de comunicação entre um PC e um dispositivo HID. Essa biblioteca é responsável por gerenciar toda a interface de controle e estabelecimento da comunicação entre o Driver HID disponível no Windows e o *software* chamado “Gerenciador USB” no diagrama.

O gerenciamento da conexão/desconexão e transferência de dados entre o PC e microcontrolador é executado pelo Gerenciador USB. Dentre as funções deste módulo, estão as descritas no Quadro 1 - Principais funções do módulo de gerenciamento da conexão USB.

Funcionalidade	Descrição
Alterar parâmetros de leitura da amostra	Permite alterar os parâmetros que definem a quantidade de pontos e a resolução de cada amostra.
Salvar última amostra	Salva as informações necessárias para se reproduzir a imagem a partir da interface web.
Conectar/Reconectar	Possibilita a conexão e reconexão do dispositivo ao computador.

Quadro 1 - Principais funções do módulo de gerenciamento da conexão USB

As funcionalidades descritas acima estão dispostas na tela para interação do usuário conforme a tela apresentada nas Figura 20 e Figura 21. O programa possui duas janelas que alternadas permitem uma configuração básica e outra avançada. Tal variação foi pensada para possibilitar um uso adaptado às necessidades do usuário.

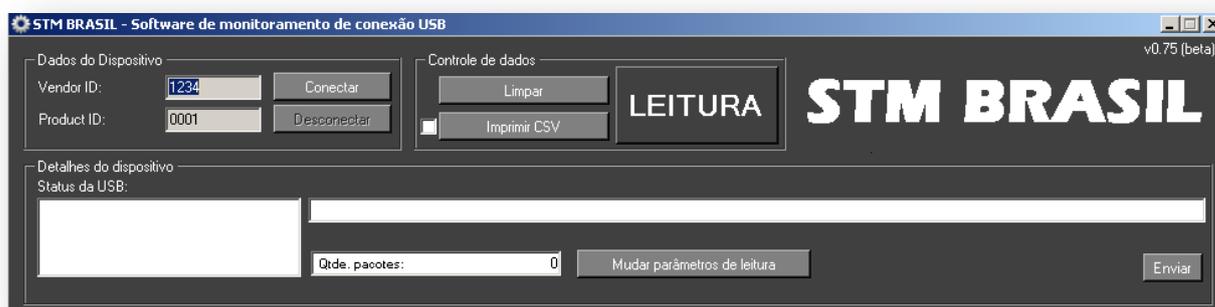


Figura 20 - Tela do software de monitoramento na configuração simplificada

Fonte: Própria

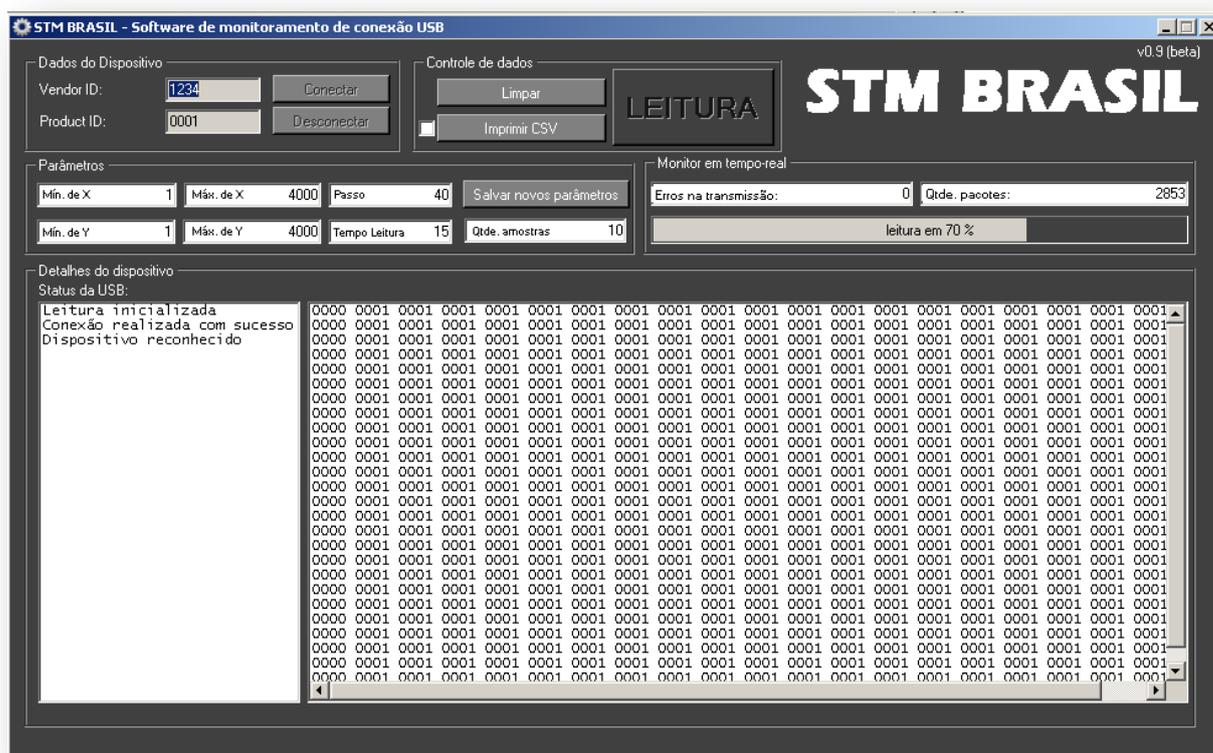


Figura 21 - Tela do software de monitoramento na configuração avançada

Fonte: Própria

Para a conexão com o microcontrolador, utilizou-se o *Vendor ID* padrão “1234” e o *Product ID* padrão “0001”. Esta configuração pode ser alterada na aba *Dados do Dispositivo*, que pode ser visualizada na Figura 22, cujo detalhe em vermelho mostra a configuração padrão.

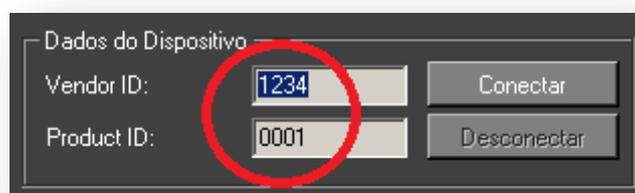


Figura 22 - Dados do Dispositivo

Fonte: Própria

Os parâmetros utilizados na leitura da amostra se resumem aos valores: mínimos e máximos de varredura das componentes horizontal e vertical, passo dado a cada medição, tempo de repouso antes de computar novos valores, e quantidades de amostras por ponto. A interface para alterar tais parâmetros se apresenta na *Figura 23*.



A interface de configuração de parâmetros de varredura, intitulada "Parâmetros", apresenta um formulário com os seguintes campos e valores:

Parâmetro	Valor
Mín. de X	1
Máx. de X	4000
Passo	40
Mín. de Y	1
Máx. de Y	4000
Tempo Leitura	15
Qtde. amostras	10

Um botão "Salvar novos parâmetros" está localizado no canto superior direito da interface.

Figura 23 - Parâmetros de Varredura

Fonte: Própria

Para facilitar o entendimento da utilização desses parâmetros na leitura, apresenta-se a *Figura 24*. Nela é possível visualizar como as informações inseridas na tela acima refletem na configuração da amostragem de pontos do material condutor.

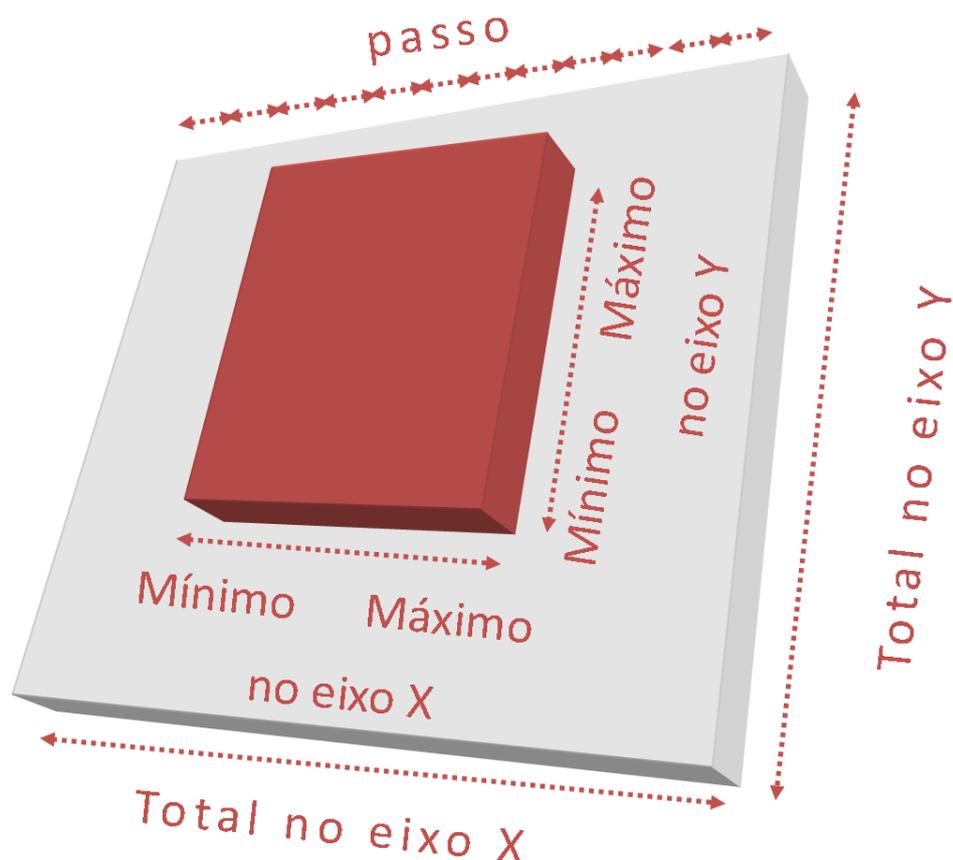


Figura 24 - Representação gráfica dos parâmetros de varredura

Fonte: Própria

Após a recepção de todos os pacotes que compreendem a amostra desejada, o programa executa a funcionalidade de salvar a leitura num arquivo simples de texto, na formatação CSV (*Comma-separated values*) que é prevista na RFC 4180 (SolidMatrix Technologies, Inc. 2005). A informação é armazenada em um arquivo para posterior leitura pelo *software* que faz a exibição das informações em formato gráfico num navegador web como o Firefox.

Toda informação é armazenada num sistema de pastas e arquivos que permitem a fácil organização e localização das amostras aferidas em momentos diferentes. Essa estrutura é semelhante a um banco de dados simplificado.

O acesso às imagens foi desenvolvido tendo como norteador o diagrama de sequência apresentado na *Figura 25*. Nele é possível visualizar a funcionalidade básica do sistema que permite a qualquer usuário conectado a rede local acessar as imagens obtidas pelo STM.

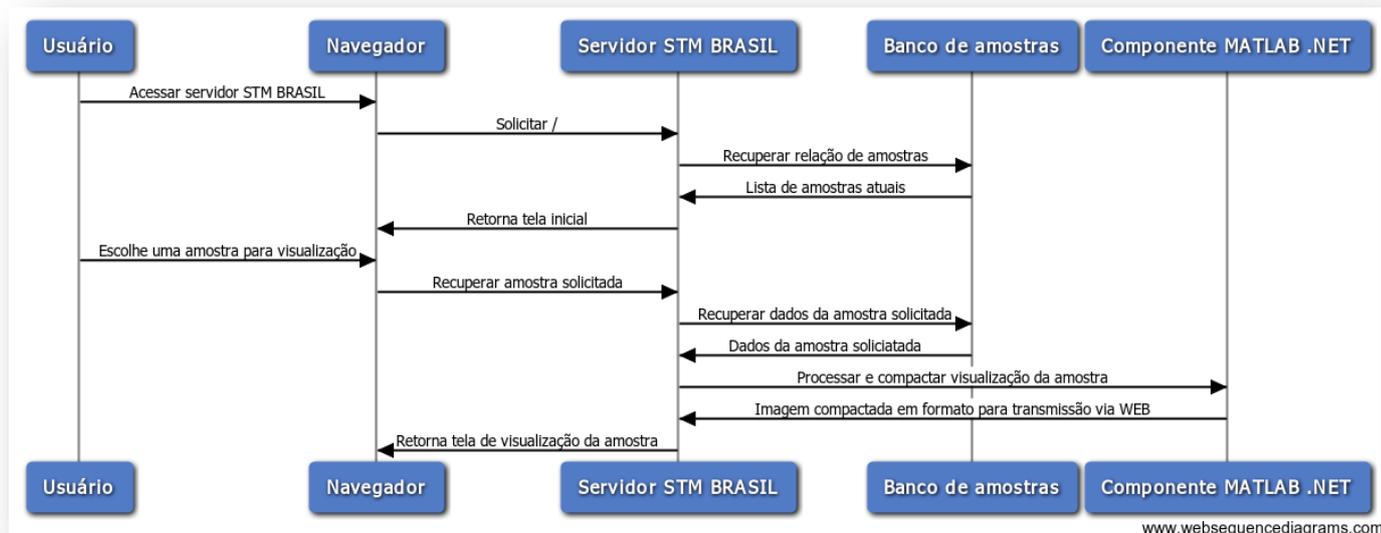


Figura 25 - Diagrama de sequência do acesso de um usuário a Galeria STM

Fonte: Própria

O *software* de processamento e geração de imagem foi escrito em linguagem C# disponível na plataforma Microsoft .NET através do ambiente de desenvolvimento integrado (IDE) Microsoft Visual Studio 2010. A solução conta ainda com a interpretação de página ASP.NET onde a linguagem de programação C# é utilizada para gerar páginas web dinamicamente de acordo com a solicitação do usuário.

O componente MATLAB .NET presente no diagrama de sequência da *Figura 25 - Diagrama de sequência do acesso de um usuário a Galeria STM* realiza o cálculo e posterior reprodução da imagem que representa a amostra a partir da informação armazenada no banco de amostras. O fluxo de informações e o algoritmo desse componente podem ser melhor compreendidos com o auxílio da *Figura 26 - Diagrama do componente MATLAB para .NET*

Foram escolhidos três padrões básicos de amostragem que diferem entre si no tamanho total das áreas observadas no microscópio. Elas variam nas escalas de 200 x 200 pontos, 50 x 200 pontos e 50 x 50 pontos por amostra.

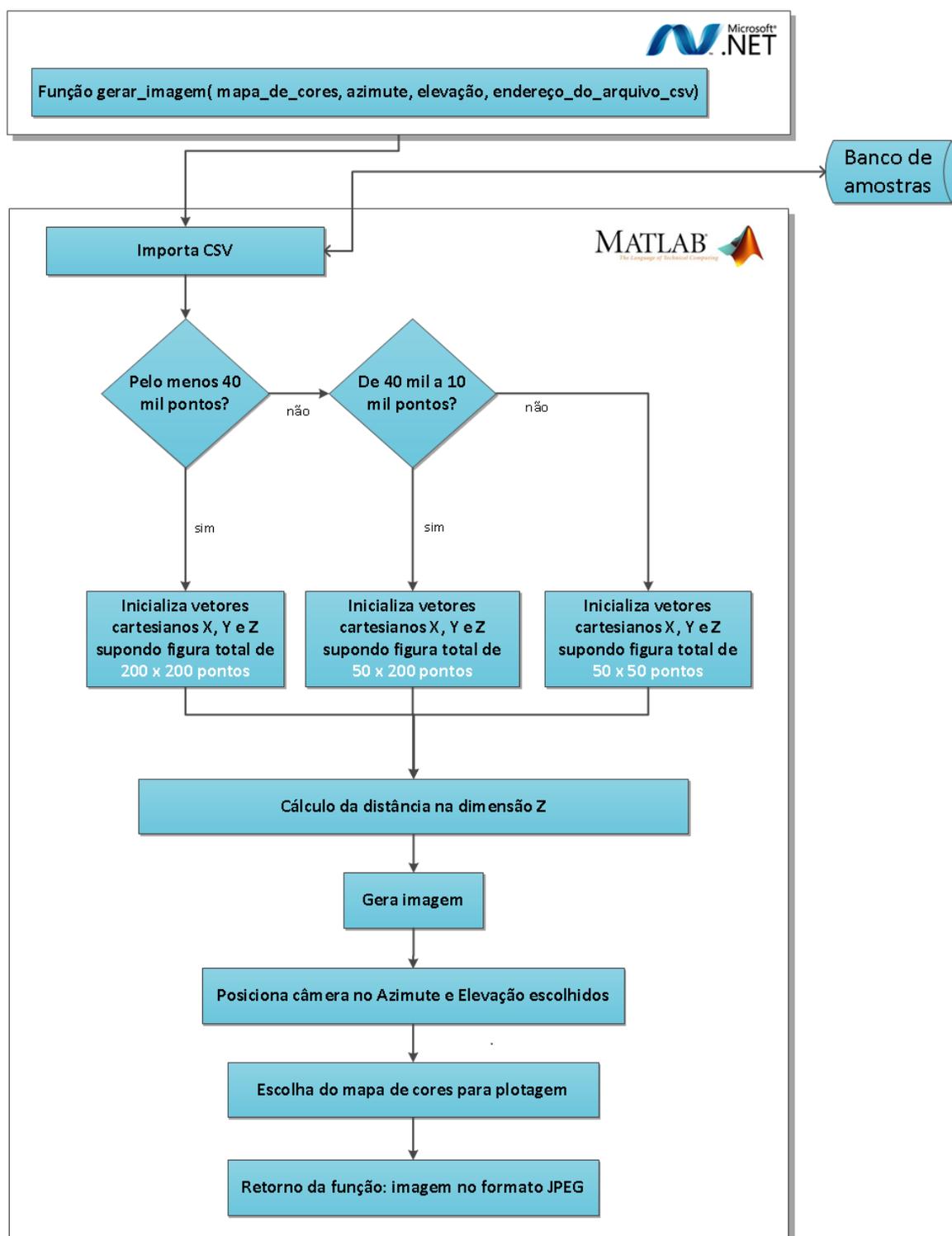


Figura 26 - Diagrama do componente MATLAB para .NET

Fonte: Própria

Na Figura 26, se faz menção ao cálculo da distância na dimensão Z. Este é realizado da seguinte forma: utilizando a equação 1 para o cálculo da altura, foram empregados os valores:

$V_t = 0,07$ V (que é a tensão de polarização entre a sonda e a amostra); $A = 1$; e de acordo com Michaelson, a função trabalho (ϕ) para o material utilizado, a amostra de cobre, é de 4,65 (MICHAELSON 1977).

Para o cálculo de I , utilizado na equação 1 como parâmetro para se recuperar o d , se utilizou a equação abaixo:

$$I = \frac{SCT}{1024 \cdot 1,1 \cdot 10^8} \quad (25)$$

O valor 1024 divide o sinal para transformar o valor recebido em 10 *bits* pela USB num valor entre 0 e 1, que corresponde de 0 % até 100 %. Já o valor $1,1 \cdot 10^8$ corresponde ao ganho do amplificador de transimpedância (ver Figura 8) que divide o valor do sinal SCT, para se obter a corrente de tunelamento. Só então se calcula o valor de ‘ d ’, que é a distância no eixo Z do sistema cartesiano e sua unidade é Angström.

Para a compilação deste código desenvolvimento na plataforma MATLAB numa componente que pudesse ser conectado a um código C#, os seguintes *softwares* foram utilizados: .NET Framework 4.0, MATLAB, MATLAB Compiler, MATLAB Builder for .NET, MATLAB Toolboxes.

A solução através de serviços WEB utilizando a tecnologia ASP.NET foi implementada com objetivo de simplificar o acesso aos dados capturados através do STM. Além disso, essa solução permite o acesso remoto aos resultados gráficos. Tal funcionalidade desenvolvida como um adicional ao projeto original traz uma ampla vantagem no uso laboratorial compartilhado do STM, em que mais de uma equipe trabalha com o equipamento. Enquanto um usuário faz medições locais, diversos usuários remotos podem acompanhar em tempo real os resultados.

Nas figuras abaixo é possível observar essa funcionalidade. Conforme a estrutura construída de pós-processamento e exibição das imagens recuperadas, pôde-se permitir a visualização universal dos resultados. No estado atual do desenvolvimento as superfícies amostradas – que são posteriormente salvas no banco de amostras – podem ser visualizadas em computadores, *notebooks*, *tablets* e *smartphones* com navegadores compatíveis com HTML na sua versão mais atual. A Figura 27 mostra a visualização da Galeria WEB rodando num navegador de um PC comum.

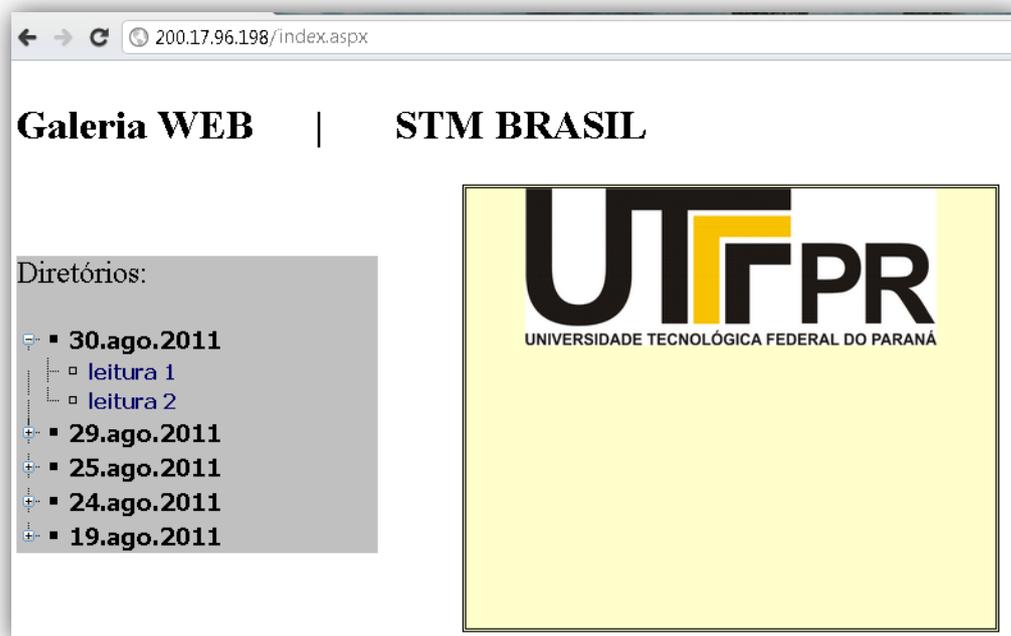


Figura 27 – página inicial da SoluçãoWeb

Fonte: Própria

Já a Figura 28 retrata de maneira intuitiva a rotação da câmera em Azimute e Elevação. Apenas com um clique do *mouse* é possível ao usuário alterar os parâmetros e uma nova imagem aparece na tela, de maneira instantânea. Aqui é utilizado o módulo compilado do MATLAB. E este é chamado de dentro do processador ASP.NET, no servidor de páginas WEB.

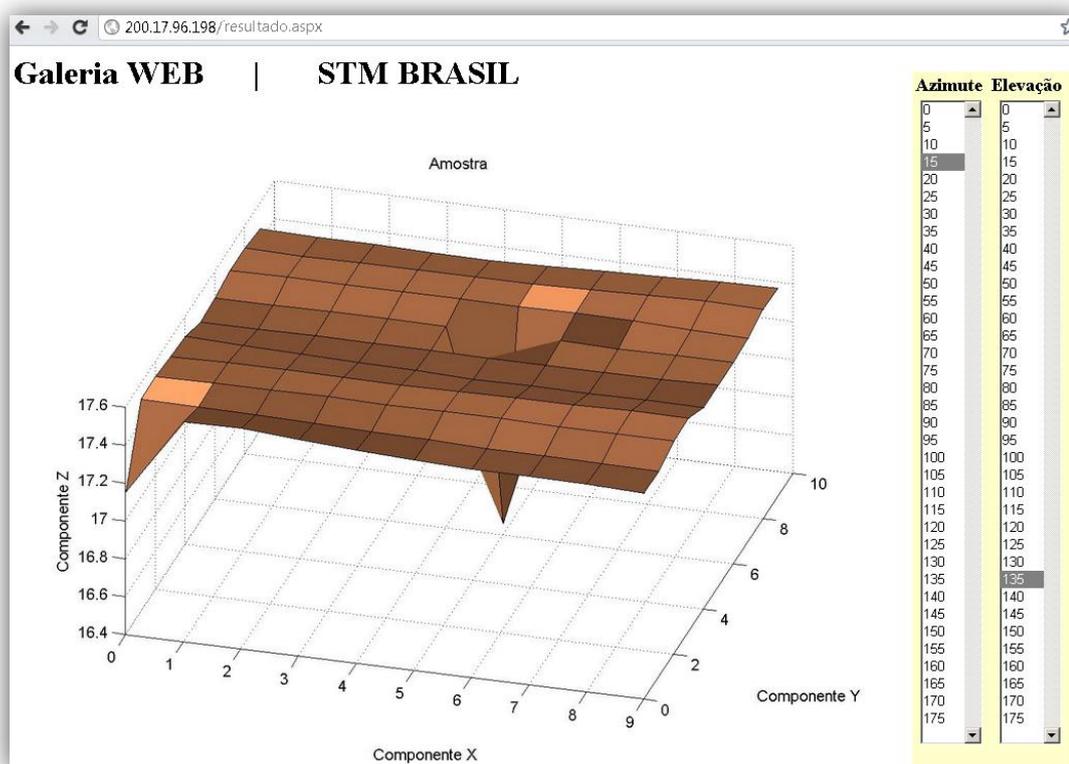


Figura 28 – Print da Solução Web para PC

Fonte: Própria

Na Figura 29, é possível se observar o acesso à Galeria de amostras realizadas pelo microscópio por meio de um *tablet* rodando o Opera Mobile.

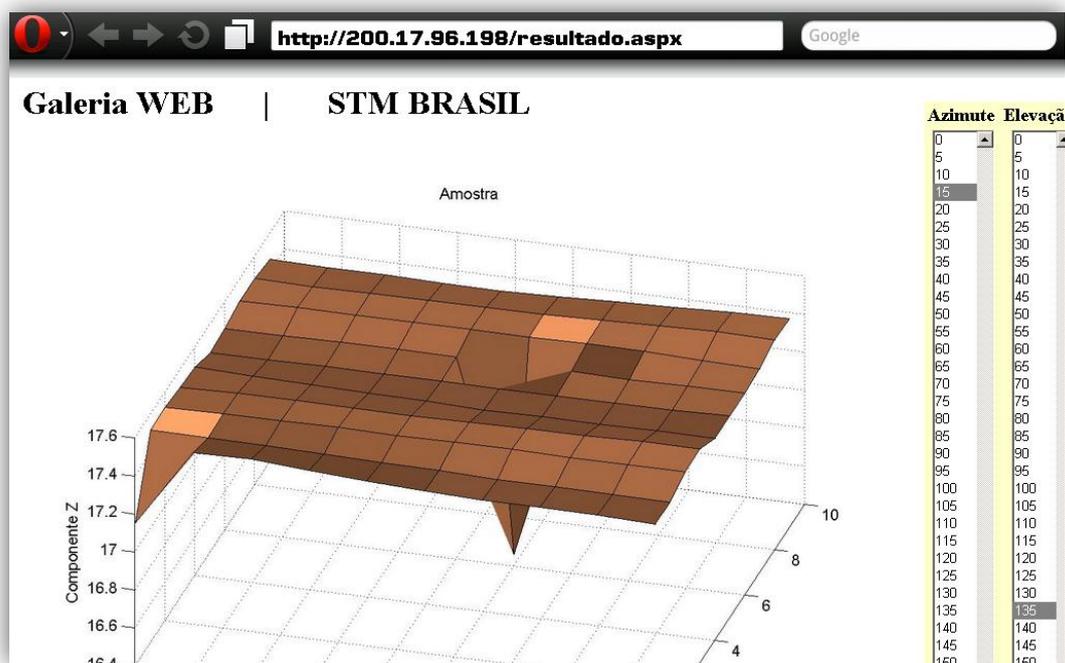


Figura 29 – Print da solução Web para tablets

Fonte: Própria

E a Figura 30, mostra o acesso a mesma galeria, com as mesmas opções de alteração de parâmetros para visualização por meio de um *smartphone* rodando o Android 2.3 e navegador Opera Mobile.

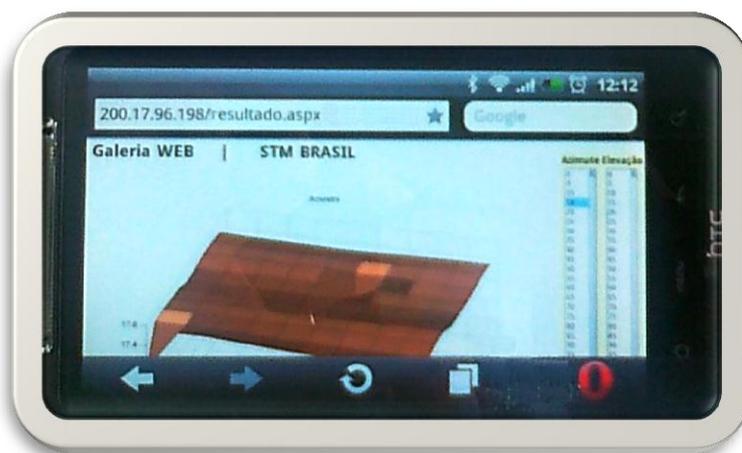


Figura 30 – Fotografia da solução Web acessada em um *smartphone*

Fonte: Própria

4.5. SEGURANÇA DE INFORMAÇÕES E CONTROLE DE VERSÕES

Para garantir a segurança do projeto foi adotada uma solução de segurança de dados através da redundância de cópias de todos os arquivos na nuvem da Amazon S3, disponibilizado pelo serviço conhecido como Drobbox. Através desse serviço, além da versão online cópias redundantes foram feitas nas máquinas de cada integrante da equipe e de uma máquina do laboratório de biotelemetria da UTFPR. Além disso, o controle de versões dos *softwares* e *firmware* foi implementado utilizando a solução Tortoise SVN.

Com o fim de se facilitar o entendimento esse mecanismo, o diagrama abaixo foi desenhado (Figura 31). Nele se demonstra o processo de troca de arquivos entre os servidores da Amazon/Dropbox com as máquinas locais e o funcionamento de maneira simplificada da lógica do SVN.

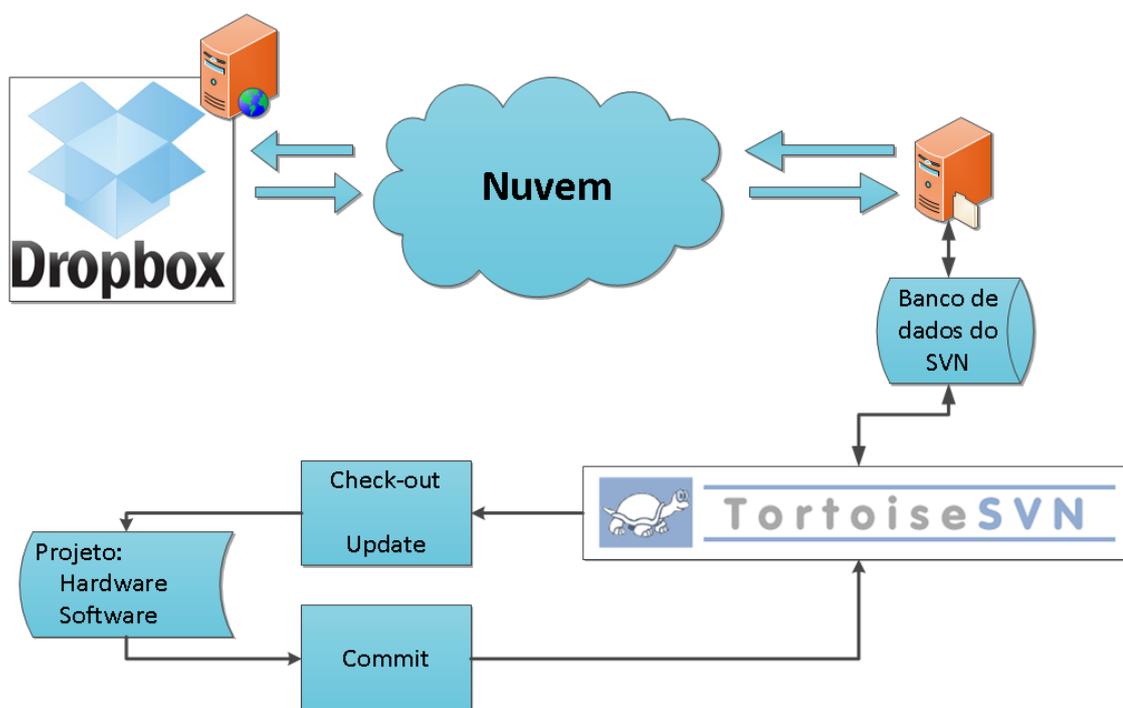


Figura 31 – Diagrama de blocos do sistema de controle de versões

Fonte: Própria

Controle e *backup* dos códigos-fonte de projetos, especialmente os divididos em vários módulos precisam ser planejados para se garantir a consistência dos mesmos. Além disso, o sistema de *backup* e controle de versões permite que sejam executadas restaurações de código-fonte de momentos anteriores do desenvolvimento.

Alguns dos termos utilizados no controle de versão se encontram definidos na tabela abaixo, construída a partir da consulta ao documento citado (WONG 2010). Eles estão aqui dispostos no *Quadro 2 - Definições comuns SVN* para facilitar a compreensão.

Termo SVN	Descrição
<i>Check-out</i>	Utilizado para se obter uma cópia do código salvo no repositório para edição, usualmente se recupera a última versão. Mas há a possibilidade de se obter versões anteriores devido à arquitetura SVN.
<i>Commit</i>	Envia o diferencial dos arquivos alterados para o <i>codebase</i> do SVN. Essas alterações são salvas como uma nova versão do código armazenado, e podem ser recuperadas.
<i>Merge</i>	A utilização deste comando se relaciona à combinação ou “reconciliação” das diferenças entre a última versão do <i>codebase</i> disponível no repositório e o código que foi enviado para o servidor na última interação.
<i>Trunk</i>	A <i>trunk</i> é a sequência principal do código fonte que é atualizada a cada iteração no progresso do projeto.
<i>Branch</i>	Do inglês, “galho ou ramo”, o <i>branch</i> é uma sequência secundária de <i>commits</i> que começa num ponto particular da sequência <i>trunk</i> . Eles são utilizados para se trabalhar numa versão paralela do <i>software</i> em aspectos específicos sem causar distúrbios no desenvolvimento principal.
<i>Tag</i>	Permite tirar uma fotografia da última versão de uma sequência <i>trunk</i> presente no SVN onde <i>commits</i> não serão mais aceitos. Elas são utilizadas para guardar versões específicas do código. Tipicamente, refletem o cronograma de versões marcadas como <i>released</i> de <i>softwares</i> comerciais.

Quadro 2 - Definições comuns SVN

A estratégia adotada tanto no desenvolvimento do *firmware* quanto do *software* foi coerente ao modelo apresentado na tabela acima e de maneira gráfica na *Figura 32*. Ali se pode entender por meio do diagrama a criação de uma *tag* chamada de *Tag A*, que foi estabelecida a partir da versão N+1. O *branch* que possibilita a estratégia de desenvolvimento em paralelo com o *mainstream* foi – no exemplo abaixo – obtido a partir da versão N e reconicionado ao mesmo apenas na versão N+2, por meio de um *merge*. Esse raciocínio pode ser melhor entendido através da observação da *Figura 32 - Sequência de fluxo de informações num repositório SVN*.

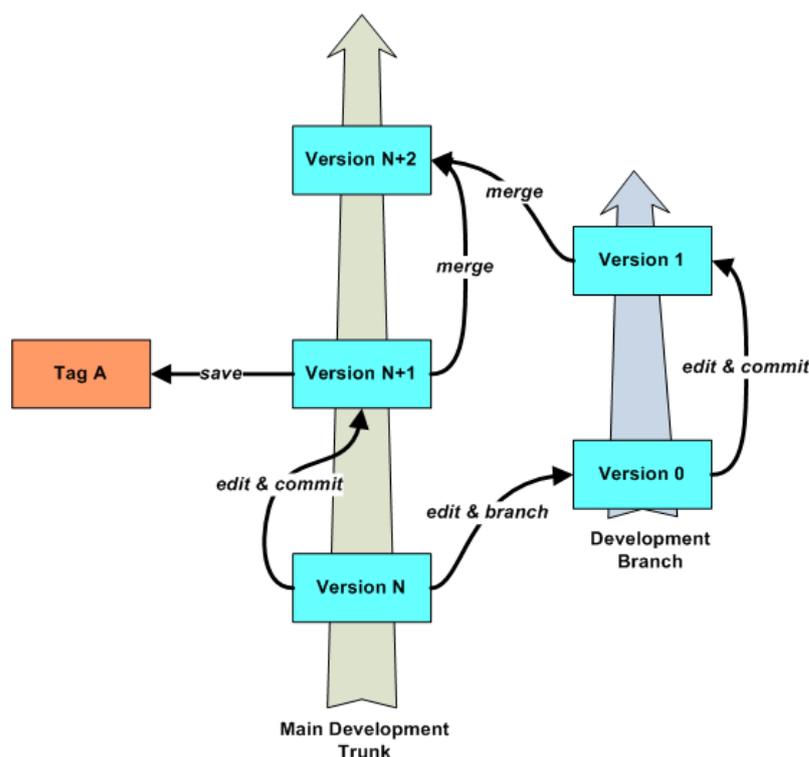


Figura 32 - Sequência de fluxo de informações num repositório SVN (WONG 2010)

Fonte: Própria

No projeto se trabalhou inicialmente com exemplos de códigos de terceiros, que foram salvos como *tags*. Chamadas de *tags* de referência, elas foram utilizados apenas como exemplos para a construção das pastas *trunk*, tanto nos *softwares* quanto no *firmware*.

A lógica de *commits* e *updates* funcionou conforme algumas regras. A primeira garante que toda alteração de código-fonte deveria ser testada em sua funcionalidade básica antes de se realizar uma interação no *codebase* do SVN. Outra regra que se observou na execução do projeto foi sempre se iniciar os trabalhos do dia com um *update* de todo o código disponível na máquina local. E no fim do dia de trabalho, realizar um *commit* do código previamente testado – no mínimo compilável – para a base de dados.

A possibilidade que a estrutura SVN + Dropbox trouxe ao projeto foi útil no desenvolvimento por diversas vezes. O *backup* de versões antigas de código, ou na máquina local no laboratório Biota, ou na máquina de colegas de equipe foi útil para se corrigir problemas em versões mais recentes.

4.6. CONSIDERAÇÕES

Durante a fase de especificação do projeto buscou-se aliar o baixo custo, mote principal do projeto, à agilidade de desenvolvimento. Para tanto foram alvos soluções já conhecidas pelos membros da equipe e que tivessem baixo custo.

Para o desenvolvimento do *hardware* eletrônico, buscou-se componentes já conhecidos de maneira a encurtar o tempo de aprendizado de uma nova tecnologia ou componente, sem deixar de lado o baixo custo. Para o sistema de posicionamento da sonda foi usado um disco piezelétrico, do mesmo tipo usado em *buzzers*, de 35mm, que no mercado custa cerca de R\$ 1,20.

Para todos os circuitos envolvendo amplificadores operacionais, exceto o amplificador de transimpedância, foi usado o circuito integrado TL071, um amplificador operacional muito difundido no meio da engenharia eletrônica, fácil de encontrar no mercado local e com preço de cerca de R\$ 1,00.

Para o amplificador de transimpedância foi usado o circuito integrado LF411N, um amplificador mais robusto e com maior imunidade a ruídos que o anterior. Isso se tornou necessário por conta do baixíssimo nível de corrente a ser adquirida, entre 1 nA e 100 nA. Como para esse nível de sinal, a relação sinal ruído de entrada é muito ruim, então para que o sinal de saída desse amplificador seja o melhor possível, a característica de 100 dB de razão de rejeição de modo comum foi crucial para o sucesso do circuito. Mesmo com todas essas características seu preço pode ser considerado baixo também, cerca de R\$ 2,80.

Para o controle analógico foi feito um sistema com duas fases de controle, uma proporcional e outra integral, ambos utilizando o amplificador operacional TL071. Este sistema já tinha sido implementado no processo de intercâmbio na Alemanha. Um dos grandes problemas deste circuito foi a dependência entre os parâmetros do sistema de controle com o sistema de condicionamento do sinal CONTROLEZ. Este circuito ficou limitado quanto a alteração destes parâmetros (SetPoint do controle proporcional, constante de tempo do integrador, etc). A vantagem do sistema de controle analógico é que o *firmware* ficou mais simples.

Através do controle analógico e o método de discretização bilinear foi possível implementar o sistema de controle digital. Com esta dificuldade do sistema de controle analógico, o controle digital se tornou mais vantajoso, mas com a desvantagem devido à complexidade do *firmware* e a taxa de amostragem. Para tanto, foi necessário adicionar mais um conversor D/A MCP4822. O controle da altura da sonda ficou mais flexível, possibilitando uma melhor sintonia do controlador PI.

Fazendo uma análise dos dois métodos de controle, optou-se pelo método de controle digital.

Para o controle e processamento dos sinais foi escolhido o PIC 18F4550, pela experiência da equipe com ele em outros projetos e por ser um microcontrolador extremamente robusto, com inúmeros recursos e facilidade de compra no mercado local. Seu preço, cerca de R\$ 20,00, não é dos mais baratos contudo ainda se encaixa no baixo custo proposto pelo projeto. Dentre suas funcionalidades é válido ressaltar os *hardwares* integrados para conversão A/D e comunicações USB e SPI.

Para a discretização dos sinais, foi utilizado o conversor A/D integrado do PIC. Ele possui resolução de 10 *bits* com métodos de discretização de aproximações sucessivas.

Para conversão D/A foi utilizado o MCP4822, novamente pela experiência da equipe com esse componente em outros projetos. Ele possui uma resolução de 12 *bits* e sua interface digital é a SPI.

A escolha estratégica do PIC18F4550 e do *software* de programação MikroC facilitou a implementação da comunicação USB. Foi utilizado um aplicativo do MikroC chamado HID terminal que possibilitou a criação do código fonte do *firmware* da comunicação USB.

Foram desenvolvidos dois *firmwares* distintos para o microscópio de tunelamento por varredura, um *firmware* para o controle digital e outro para o controle analógico. Devido à limitação da taxa de amostragem do *firmware* do sistema de controle digital, o código fonte do controle digital ficou mais complexo que do analógico, evitando o uso de chamada de funções, que toma mais tempo de processamento. Apesar desta complexidade, ela se mostrou mais eficiente.

Para a comunicação entre o *firmware* e o *software* instalado no computador foi desenvolvido um pequeno protocolo que permitisse o envio de parâmetros, a troca de informações sobre o início, fim, contagem de pacotes recebidos, dentre outras informações.

Já no quesito inovação, a solução *WEB* desenvolvida permitiu que qualquer pesquisador conectado a uma rede de computadores com acesso ao servidor ao qual o STM está conectado possa acompanhar as medições que estão sendo feitas no equipamento, e verificar através do banco de amostras, todos os outros materiais e experimentos realizados com o dispositivo. Essa solução desenvolvida em .NET, contou com a compilação de um código para processamento e reprodução de imagem desenvolvido em MATLAB e ainda com um *software* para Windows™ que gerencia toda a conexão USB entre o microcontrolador e o computador.

GESTÃO

5.1. CRONOGRAMA E CUSTOS

O cronograma desenvolvido para desenvolvimento do microscópio de tunelamento possui duração total de 11 meses, iniciando em fevereiro de 2011 e terminando em dezembro de 2011. O *Quadro 3* apresenta o cronograma detalhado do projeto.

Ação	2011											
	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out	Nov	Dez
1. Pesquisar e estudar princípio físico do efeito túnel e tecnologias que podem ser aplicados no projeto		X	X									
1.1. Pesquisar e estudar o efeito túnel		X										
1.2. Estudar tecnologias que podem ser aplicados no projeto			X									
2. Documentar a defesa da disciplina de TCC1			X	X	X	X						
3. Defender o projeto de TCC1							X					
4. Construir a estrutura mecânica						X	X					
5. Desenvolver hardware						X	X	X				
5.1. Projetar dos circuitos de polarização, amplificação, varredura, controle, digitalização e comunicação						X						
5.2. Especificar componentes e verificar sua disponibilidade no mercado						X						
5.3. Fazer a aquisição dos componentes, confeccionar as placas de circuito impresso, montar das placas e integrar do hardware com a estrutura mecânica						X	X	X				
6. Executar testes							X	X	X			
7. Desenvolver o firmware para digitalização, pré-processamento dos sinais e comunicação com o PC								X	X	X		
7.1. Desenvolver firmware para digitalização								X	X			
7.2. Desenvolver firmware para pré-processamento								X	X			
7.3. Desenvolver firmware para comunicação com o PC									X	X		
8. Reproduzir a imagem no PC a partir dos dados obtidos									X	X	X	
9. Documentar o projeto											X	
10. Apresentar o projeto para a banca examinadora												X

Quadro 3 - Cronograma detalhado

*Nota explicativa: O preenchimento das lacunas acima está apresentado de tal forma que as atividades principais e seu período de execução receberam a cor cinza escuro, e os subitens das mesmas a cinza claro.

Para uma melhor controle do andamento do projeto, foram especificadas algumas datas importantes que dizem respeito ao início e ao término de cada uma das 10 grandes etapas. Essas datas são mostradas no *Quadro 4*.

ID	Ação	Data de Início	Data de Conclusão	Horas
1	Pesquisar e estudar princípio físico do efeito túnel e tecnologias que podem ser aplicados no projeto	01/02/2011	31/03/2011	150
2	Documentar a defesa da disciplina de TCC1	05/03/2011	15/06/2011	150
3	Defender o projeto de TCC1	09/07/2011	09/07/2011	1
4	Construir a estrutura mecânica	01/06/2011	31/07/2011	60
5	Desenvolver o <i>hardware</i>	01/06/2011	31/08/2011	300
6	Executar testes	01/07/2011	15/09/2011	240
7	Desenvolver o <i>firmware</i> para digitalização, pré-processamento dos sinais e comunicação com o PC	15/08/2011	15/10/2011	180
8	Reproduzir a imagem no PC a partir dos dados obtidos	16/09/2011	15/11/2011	180
9	Documentar o projeto	01/11/2011	30/11/2011	150
10	Apresentar o projeto para a banca examinadora	15/12/2011	15/12/2011	2
	Microscópio de tunelamento – STM (total de horas)			1413

Quadro 4 - Datas importantes

Para a gestão do projeto foi definida também a responsabilidade das ações através da definição das horas trabalhadas por cada integrante da equipe e o custo em cada uma das áreas do desenvolvimento do projeto. O total de horas necessárias para a conclusão do projeto foi de 1413 horas. O *Quadro 5* mostra responsabilidade das ações, a distribuição das horas por integrante da equipe e o custo de cada atividade. O custo total do projeto foi estimado em R\$1500,00, como visto no *Quadro 5*. Parte desse custo será pago através da universidade (via CAPES), diminuindo o custo aos membros da equipe.

Ação	Responsável/Horas			Custos/ (R\$)
	Airton	Felipe	Luiz Davi	
1. Pesquisar e estudar princípio físico do efeito túnel e tecnologias que podem ser aplicados no projeto	150			

1.1. Pesquisar e estudar o efeito túnel	30	30	30	90
1.2. Estudar tecnologias que podem ser aplicados no projeto	20	20	20	60
2. Documentar a defesa da disciplina de TCC1	150			
2.1 Documentar a defesa da disciplina de TCC1	50	50	50	150
3. Defender o projeto de TCC1	1			1
4. Construir a estrutura mecânica	60			R\$500,00
4.1 Construir a estrutura mecânica	20	20	20	60
5. Desenvolver <i>hardware</i>	300			R\$1000,00
5.1. Projetar dos circuitos de polarização, amplificação, varredura, controle, digitalização e comunicação	0	50	0	50
5.2. Especificar componentes e verificar sua disponibilidade no mercado	20	0	20	40
5.3. Fazer a aquisição dos componentes, confeccionar as placas de circuito impresso, montar as placas e integrar do <i>hardware</i> com a estrutura mecânica	70	70	70	210
6. Executar testes	240			
6.1 Executar testes	80	80	80	240
7. Desenvolver o <i>firmware</i> para digitalização, pré-processamento dos sinais e comunicação com o PC	180			
7.1. Desenvolver <i>firmware</i> para digitalização	40	0	0	40
7.2. Desenvolver <i>firmware</i> para pré-processamento	40	0	0	40
7.3. Desenvolver <i>firmware</i> para comunicação com o PC	40	60	0	100
8. Reproduzir a imagem no PC a partir dos dados obtidos	180			
8.1 Reproduzir a imagem no PC a partir dos dados obtidos	0	60	120	180
9. Documentar o projeto	150			
9.1 Documentar o projeto	50	50	50	150
10. Apresentar o projeto para a banca examinadora	2			2
Total de horas geral	1413			
Total de horas por pessoa	461	491	461	1413/ R\$1500,00

Quadro 5- Responsabilidade das ações e custo

5.2. RISCOS

Foi realizado um levantamento dos riscos que poderiam afetar a execução do projeto. Uma vez levantados, os riscos foram classificados segundo sua probabilidade de ocorrência e o

impacto que causariam no projeto. A probabilidade de ocorrência varia de 0 (representando 0%) a 1 (representando 100%) e o impacto tem variação de 1 a 5 sendo 1 para muito baixo, 2 para baixo, 3 para médio, 4 para alto e 5 para muito alto. Multiplicando ambos os índices obtém-se o grau do risco. Foram determinadas então as ações que devem ser tomadas caso o risco em questão venha a realmente ocorrer. O *Quadro 6* mostra a análise de riscos realizada.

Grau de Risco	Descrição do Risco	Descrição do Impacto	Probabilidade (P)	Impacto (I)	Grau (PxI)	Ação
Alto	Funcionamento incorreto da estrutura mecânica.	Dificulta o funcionamento eletromecânico do sistema	0,7	5	3,5	Eliminar: refazer o projeto mecânico e reconstruir protótipo
Médio	Falha no desenvolvimento do <i>hardware</i>	Mau funcionamento do sensor de corrente	0,6	4	2,4	Eliminar: encontrar e solucionar o problema
Médio	Componentes danificados por má utilização	Atraso no desenvolvimento	0,8	2	1,6	Eliminar: ter estoques dos componentes
Médio	Atraso na entrega de componentes	Atraso no desenvolvimento	0,5	3	1,5	Mitigar: adiantar outras partes do projeto
Médio	Atraso na fabricação da placa de circuito impresso	Atraso no desenvolvimento	0,5	3	1,5	Mitigar: adiantar outras partes do projeto
Baixo	Aumento no preço dos componentes	Aumento do custo do projeto	0,5	2	1	Conviver
Baixo	Falha no desenvolvimento do <i>software</i>	Mau funcionamento do sistema	0,3	2	0,6	Eliminar: testar partes problemáticas separadamente.

Quadro 6- Análise de riscos

P = probabilidade de Ocorrência (variação de 0 a 1)

I = Impacto (variação de 1 a 5)

5.3. CONSIDERAÇÕES

Analisando a gestão do projeto é possível afirmar que foram trabalhadas 1427 horas no total de 1413 horas estimadas. Esse valor corresponde a 100% do projeto

concluído. Foram gastos R\$750,00 dos R\$1500,00 previsto, que corresponde a 50% do total. O projeto foi concluído em tempo hábil e não apresentou atraso. O cronograma foi adiantado em duas semanas uma vez que a defesa foi marcada para o dia 30 de novembro – 15 dias antes do planejado.

6. RESULTADOS

6.1. ESTRUTURA FÍSICA

6.1.1. VIBRACOES MECÂNICAS

Pelo fato de se trabalhar com distâncias muito pequenas, um dos grandes desafios encontrados foram as vibrações mecânicas. Podem ser citadas como fonte de vibrações: correntes de ar, máquinas ligadas, motores, aparelhos de ar condicionado, fluxo de pessoas nas intermediações próximas ao sistema, e até ondas mecânicas propagadas pelo ar causadas pela fala de pessoas no laboratório. Pelo fato da universidade estar situada no centro da cidade, o tráfego de veículos, principalmente ônibus, faz com que todo o prédio onde está o laboratório vibre frequentemente.

Para minimizar este efeito, o projeto foi colocado sobre uma estrutura numa mesa plana horizontal, sem contato com as paredes. Esse sistema anti-vibração é composto por (da camada mais inferior até a mais alta, onde está a estrutura apresentada na *Figura 34*): duas camadas de espuma, em cima destas uma pedra de cimento plana horizontal de tamanho 45x45x5 cm, que recebeu duas câmaras de ar de bicicleta (uma circunscrita na outra) e no topo outra pedra de mesmas dimensões que a anterior. A parte mecânica do microscópio de tunelamento (*Figura 34*) foi colocada em cima dessa estrutura.

Este sistema de amortecimento minimizou bastante as vibrações mecânicas. Mas elas não foram eliminadas completamente por se tratar de uma solução de baixo custo.

6.1.2. CONSTRUÇÃO DA ESTRUTURA MECÂNICA

Conforme indicado no escopo do trabalho, a estrutura física foi construída nos moldes apresentados no capítulo 2, e pode ser vista na *Figura 33* e na *Figura 34*. Nessas imagens estão visíveis os detalhes da construção, como a colocação dos dois micrômetros nas duas extremidades opostas do microscópio, juntamente com a fixação por meio de parafusos do suporte que evita vibrações no disco piezelétrico. Além desses, existem outros dois parafusos visíveis nas mesmas figuras citadas neste parágrafo que também auxiliam na fixação da placa metálica em formato V.

Na *Figura 34* é possível se ver com detalhes a fixação do disco piezelétrico logo abaixo do suporte (branco) que o mantém na mesma posição durante as amostragens, sem atrapalhar a deflexão. Tal fixação só é possível porque o suporte segura apenas as bordas do disco, deixando a região central onde está o material piezelétrico livre.

O papel alumínio que pode ser visto nas figuras 33 e 34 foi usado para envolver a estrutura mecânica, onde a corrente de tunelamento é gerada, formando assim uma gaiola de Faraday. Para melhor performance da gaiola, ela foi conectada ao terra do circuito.



Figura 34 - Microscópio completo

Fonte: Própria

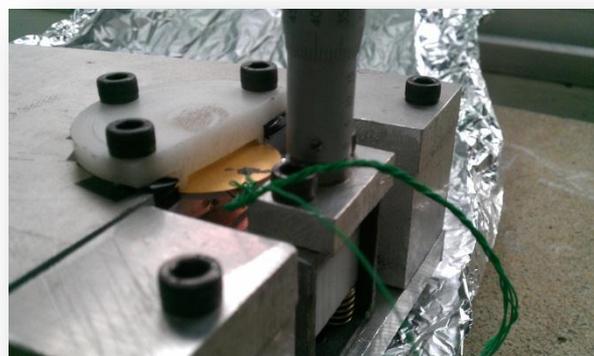


Figura 33 - Sonda e sua fixação no detalhe

Fonte: Própria

6.2. *HARDWARE* ELETRÔNICO

O *hardware* eletrônico foi feito em placas do tipo universal. A ideia inicial era montar em uma placa industrializada de circuito impresso, mas não foi possível devido a questões financeiras e tempo de projeto. Foram montados dois protótipos diferentes: a primeira com alimentação direto da rede elétrica, com um circuito de retificação e geração de alimentação +5 V, -12 V e +12 V inclusos, como pode ser visto na *Figura 35* e a segunda sem esses circuitos que foi a placa usada no projeto, como pode ser visto na *Figura 36*. O protótipo com o circuito de alimentação apresentou problemas devido à interferência da tensão da rede, apresentando instabilidade na aquisição da corrente de tunelamento. Como resolução para tal dificuldade foi utilizada uma fonte externa para fornecer +12 V e -12 V para alimentação da segunda placa implementada. A geração de 5 V continuou na placa, sendo gerado a partir de +12 V.

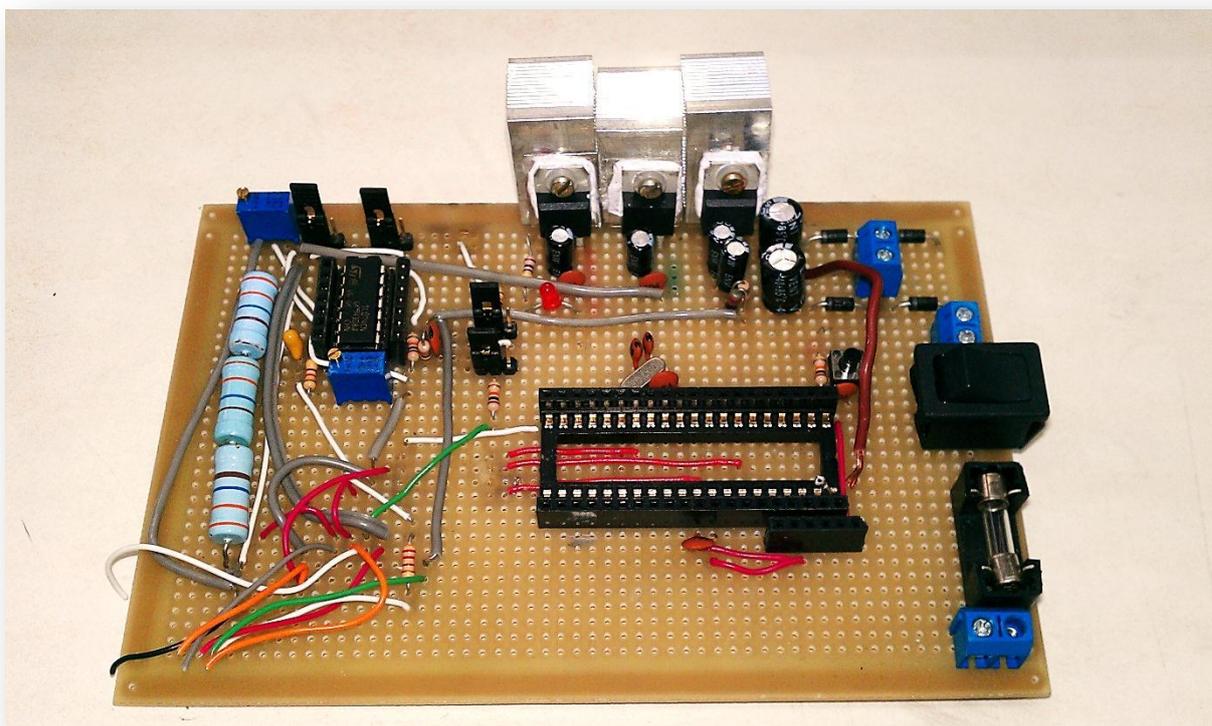


Figura 35 - Placa com alimentação da rede

Fonte: Própria

Junto ao segundo protótipo foram implementadas duas placas adicionais que são conectadas a este protótipo (*Figura 36*). Uma é responsável pelo condicionamento do sinal **CONTROLEZ** para o controle analógico, e outra processa a entrada SCT e o controle digital do piezelétrico.

Na *Figura 36* a placa no canto superior esquerdo é responsável por conectar o circuito principal ao *hardware* propriamente do microscópio, conforme o capítulo 4.

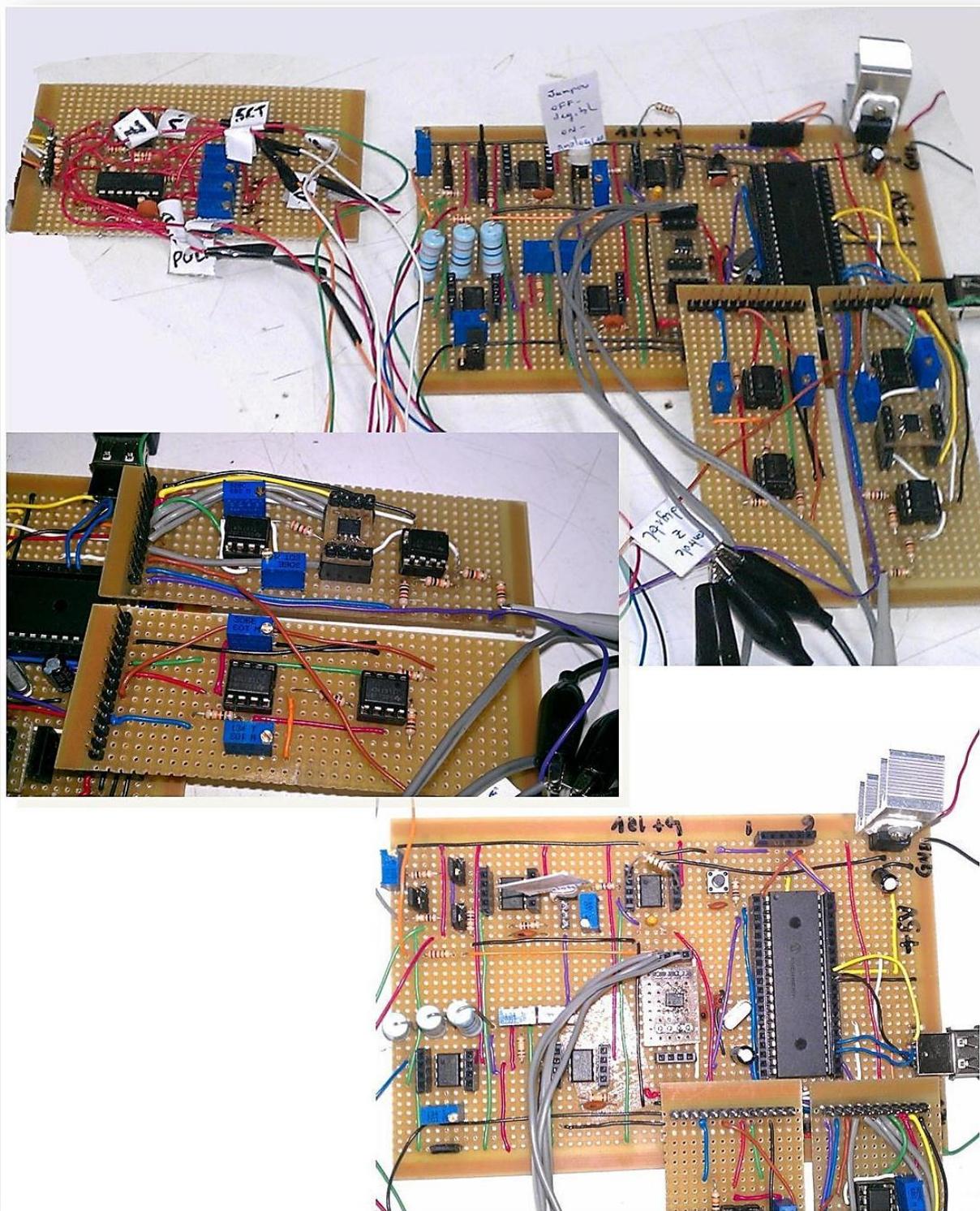


Figura 36 - Circuitos do prototipo final

Fonte: Própria

6.3. DEFLEXÃO DO DISCO PIEZELÉTRICO

Para se descobrir qual a deflexão do piezolétrico na direção do eixo Z em relação à tensão aplicada foi proposto um experimento, composto por um espelho na mesma direção do eixo apoiado por um tripé móvel, sendo uma das pernas apoiadas no centro do disco piezolétrico. Neste espelho foi aplicada uma luz (laser) perpendicularmente, sendo refletida ao anteparo a certa distância. Com a aplicação de uma onda triangular com um gerador de funções no disco piezolétrico, foi possível calcular através de relações trigonométricas a relação da deflexão no eixo Z com a amplitude da tensão aplicada ao sensor. É possível observar o sistema na Figura 37:

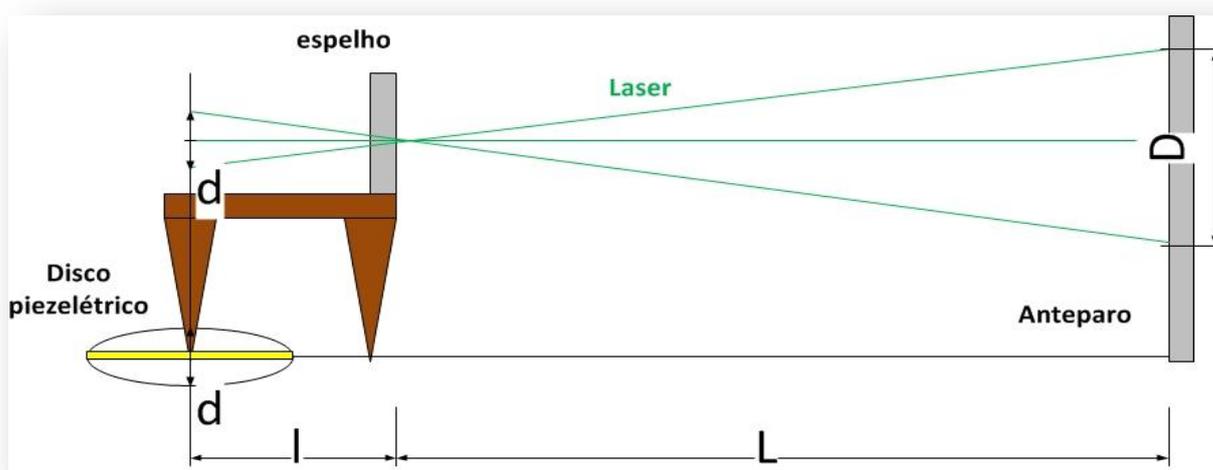


Figura 37 - Montagem da experiência para verificação da deflexão do disco piezolétrico

Fonte: Própria

onde:

d é a deflexão do movimento do disco piezolétrico no eixo Z;

D é a reflexão do laser no anteparo;

l é a distância do espelho ao centro do disco piezolétrico;

L é a distância do espelho ao anteparo.

Através da relação trigonométrica deduz:

$$d = \frac{l \cdot D}{L} \quad (26)$$

De acordo com o arranjo do experimento, o valor medido para a distância do espelho ao anteparo (**L**) foi de 20 m, a distância do espelho ao centro do disco piezelétrico (**l**) foi de 32,5 mm, a reflexão do laser ao anteparo (**D**) foi de 2,75 mm. Através destes dados foi possível calcular a deflexão do movimento do disco piezelétrico no eixo **Z**(**d**), que foi de 4,45 μm .

Com a informação da deflexão de 4,45 μm e a tensão pico-a-pico aplicado ao piezelétrico de 20,4 V:

$$B = \frac{d}{V_{pp}} \quad (27)$$

onde:

d é a deflexão do movimento do disco piezelétrico no eixo **Z**;

B é a distância do piezelétrico proporcional a tensão aplicada;

V_{pp} é a tensão pico-a-pico da onda triangular;

O valor da distância do piezelétrico proporcional à tensão aplicada é de 0,21 $\mu\text{m}/\text{V}$.

Com este experimento foi possível implementar a malha de controle analógica e digital e auxiliou na visualização da relação da polaridade da tensão aplicada com o sentido do movimento do piezelétrico (tensão negativa afasta a sonda da amostra e tensão positiva aproxima). Essa experiência deve ser repetida para cada novo disco piezelétrico usado, pois pode ser que ele se movimente no sentido inverso à tensão aplicada.

6.4. CORRENTE DE TUNELAMENTO

Um passo importante para o progresso do projeto foi a verificação que a corrente lida pelo amplificador de transimpedância (vide Figura 8) realmente surge por efeito túnel. Para tanto o procedimento adotado foi: aplicar ao piezelétrico um sinal triangular com 20 Vpp e frequência de 1 Hz, e então amostrar a variação de corrente proveniente da sonda.

Ao aplicar um sinal de tensão triangular no piezelétrico, produz-se uma variação linear na posição do eixo cartesiano Z da sonda.

Como visto no capítulo 2, a corrente de tunelamento é exponencialmente proporcional à barreira de potencial (distância entre amostra e sonda). Portanto deve-se observar uma variação de corrente exponencial quando é aplicada uma variação linear (onda triangular). De fato isso foi observado, como pode ser visto na *Figura 38*, em que o canal 1 do osciloscópio (linha amarela) mostra o sinal SCT (vide Figura 8) quando aplicado ao piezelétrico à onda triangular. Essa variação da corrente de tunelamento ocorre muito rapidamente, nesse caso aproximadamente de 200 μ s, como pode ser visto na *Figura 38*, e por isso não é possível visualizar corretamente a onda triangular na *Figura 38*, entretanto ela corresponde ao canal 2 do osciloscópio (linha verde).

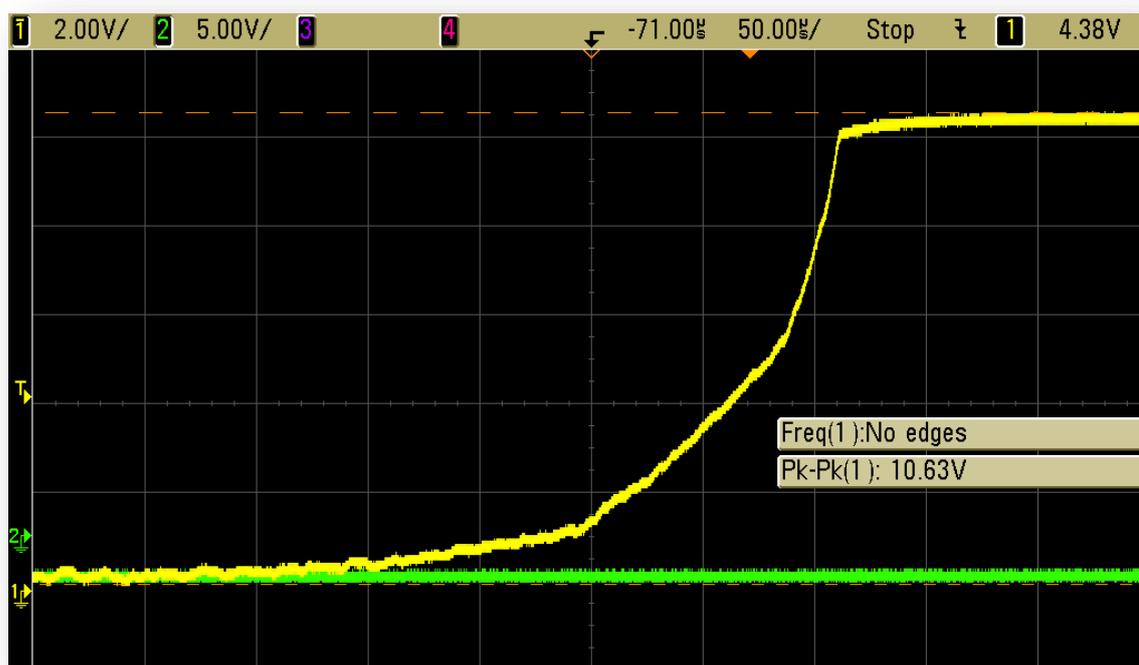


Figura 38 - Visualização da dependência exponencial da corrente de tunelamento

Fonte: Própria

Para comprovar matematicamente que esse sinal realmente se aproxima de uma curva exponencial, os pontos do osciloscópio foram salvos em um arquivo tipo CSV. Então os valores correspondentes ao tempo foram usados para formar o eixo das abscissas do gráfico da e os valores referentes a amplitude do canal 1(sinal SCT) constituem o eixo das ordenadas. Com esses dados, a curva em azul foi construída. A curva em preto é uma curva de tendência exponencial.

Outro teste se dá através da linearização da variação amostrada. Aos valores relativos ao sinal SCT foi aplicado o logaritmo natural, também conhecido como logaritmo neperiano. Os resultados foram usados para formar o eixo das ordenadas e dessa forma construir a curva em azul da *Figura 40*. O gráfico em preto representa uma curva de tendência linearização da curva LN(SCT). Como é possível observar, ambas as curvas são muito próximas de suas respectivas curvas de tendência, o que prova que a corrente obtida pelo sistema é predominantemente gerada por efeito túnel.

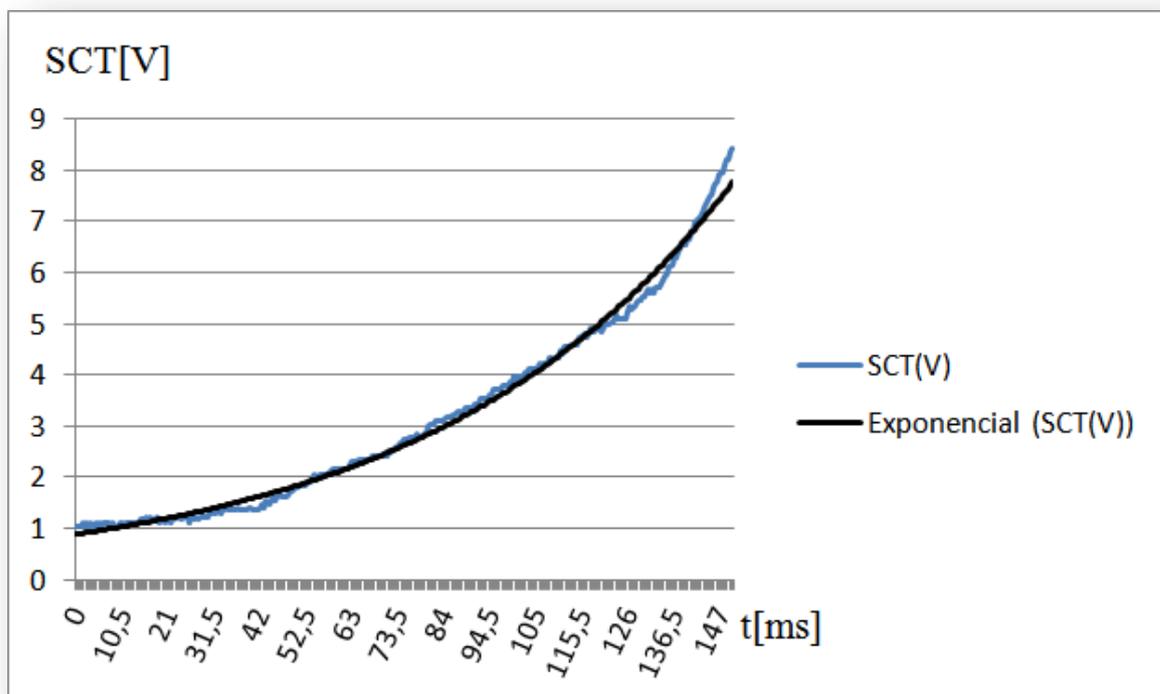


Figura 39 - Gráfico SCT vs Tempo e linha de tendência exponencial vs Tempo

Fonte: Própria

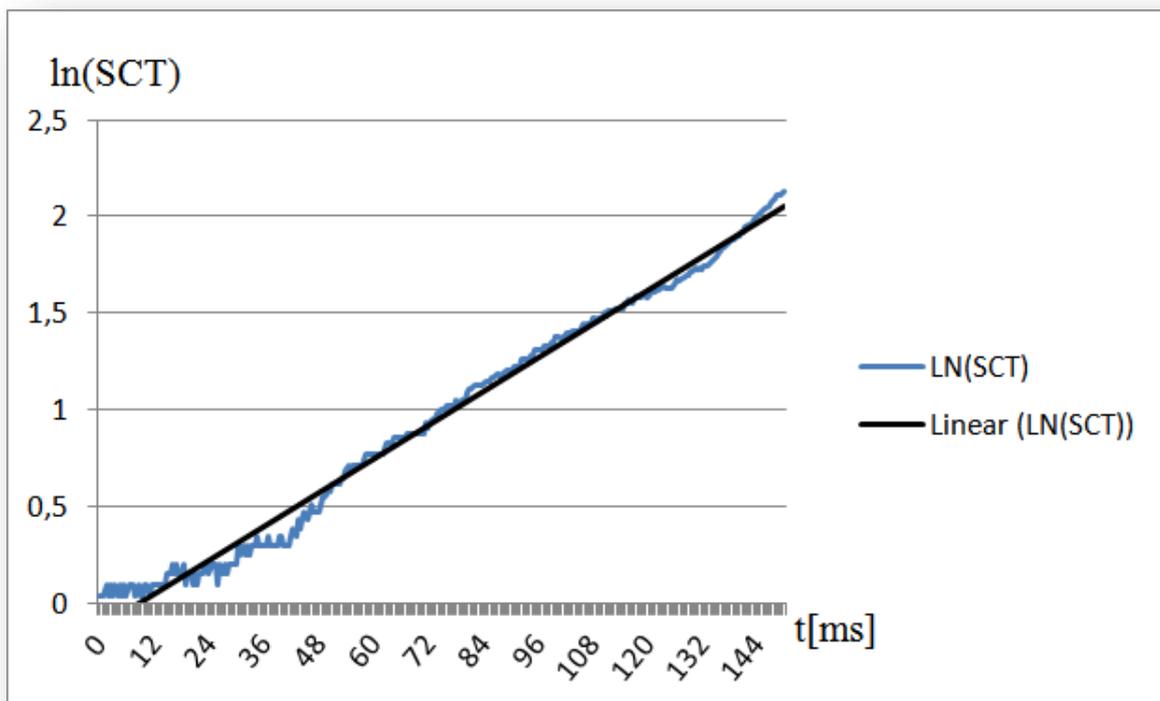


Figura 40 - Gráfico $\ln(\text{SCT})$ vs Tempo e linha de tendência vs Tempo

Fonte: Própria

6.5. COMPROVAÇÃO POR REPETITIVIDADE

Dois testes realizados com sucesso para a verificação da repetitividade e validação das imagens obtidas está apresentado nas figuras que seguem neste subitem.

6.5.1. PRIMEIRO TESTE DA REPETIÇÃO DE PADRÕES ENTRE MEDIÇÕES

O primeiro teste utilizado como parte da comprovação compreende da figura 41 até a figura 45. Seu princípio é simples. Na primeira imagem (figura 41) foi realizada uma leitura em formato retangular que apresenta 50 x 200 pontos. A superfície observada faz parte de uma placa de cobre, previamente lixada e tratada para a leitura.

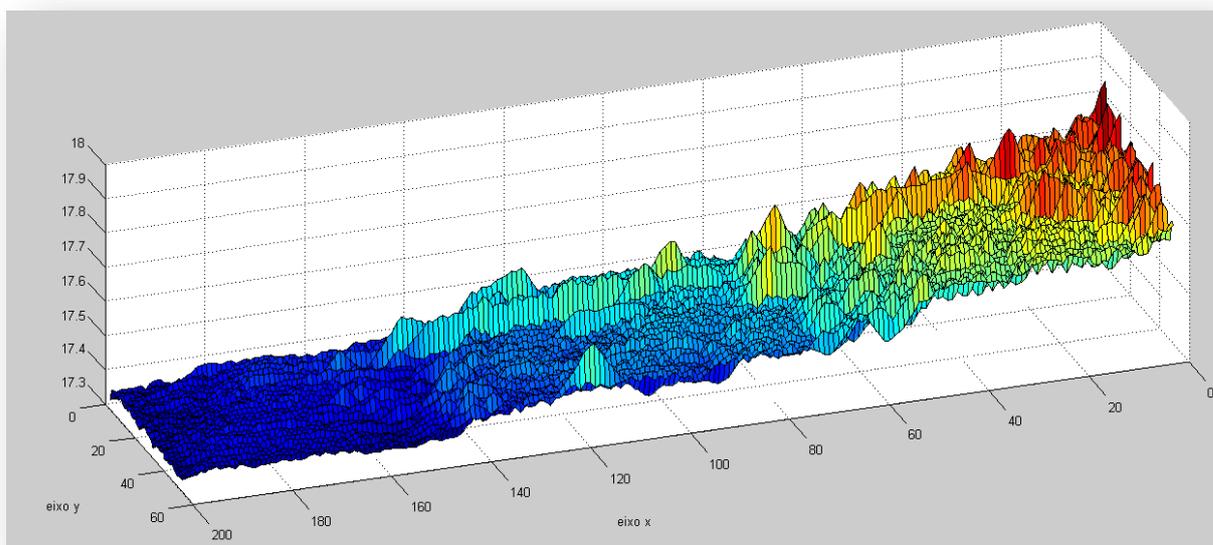


Figura 41 - amostra um – completa

Fonte: Própria

Para que se pudesse comprovar que a imagem observada não provém de um padrão de erro nas leituras, ou algum outro fenômeno que não a superfície de cobre, adotou-se a seguinte estratégia: a região observada na figura 41 foi amostrada novamente utilizando novo padrão de leitura que a dividiu em 4 partes iguais de 50 x 50 pontos. A figura 42 está limitada aos últimos 50 valores do eixo x (ponto 200 ao 150) e o eixo y completo da figura 42.

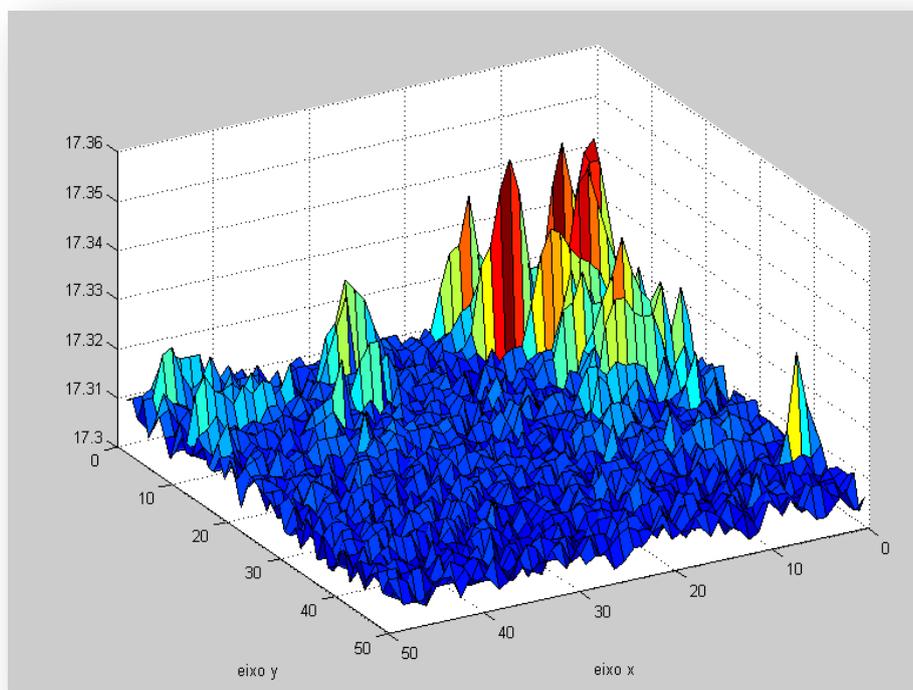


Figura 42 - amostra um - primeira fatia

Fonte: Própria

É possível observar de maneira bastante clara a correlação existente entre a imagem apresentada na figura 42 com a porção da imagem equivalente a figura 41. De forma semelhante, a Figura 43 está limitada dos pontos 150 ao 100 do eixo x e o eixo y completo da figura 41. A mesma correlação está presente embora menor que na primeira imagem (figura 41).

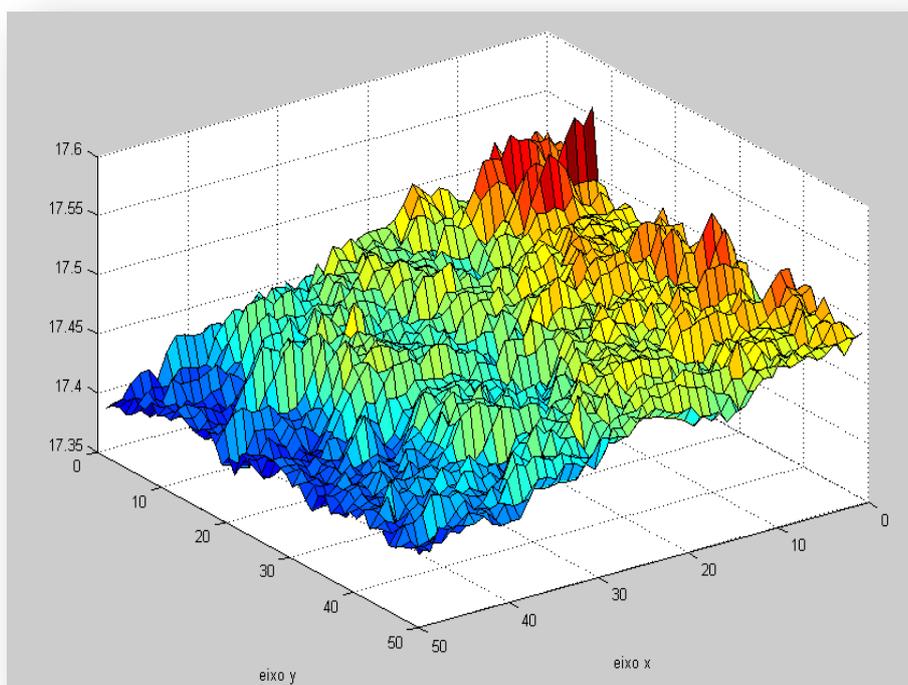


Figura 43 - amostra um - segunda fatia

Fonte: Própria

As imagens representadas pela Figura 44 e pela Figura 45, são, respectivamente, retratos das regiões entre os pontos 100 – 50 e 50 – 0 do eixo x e o eixo y completo da figura 41. Nessas últimas imagens, a correlação também está presente.

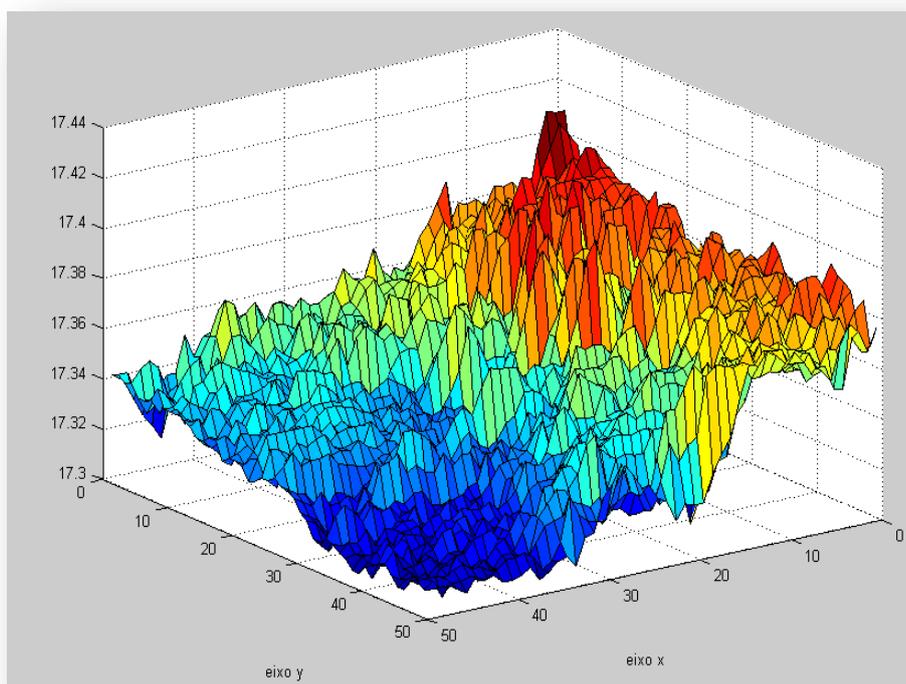


Figura 44 - amostra um - terceira fatia

Fonte: Própria

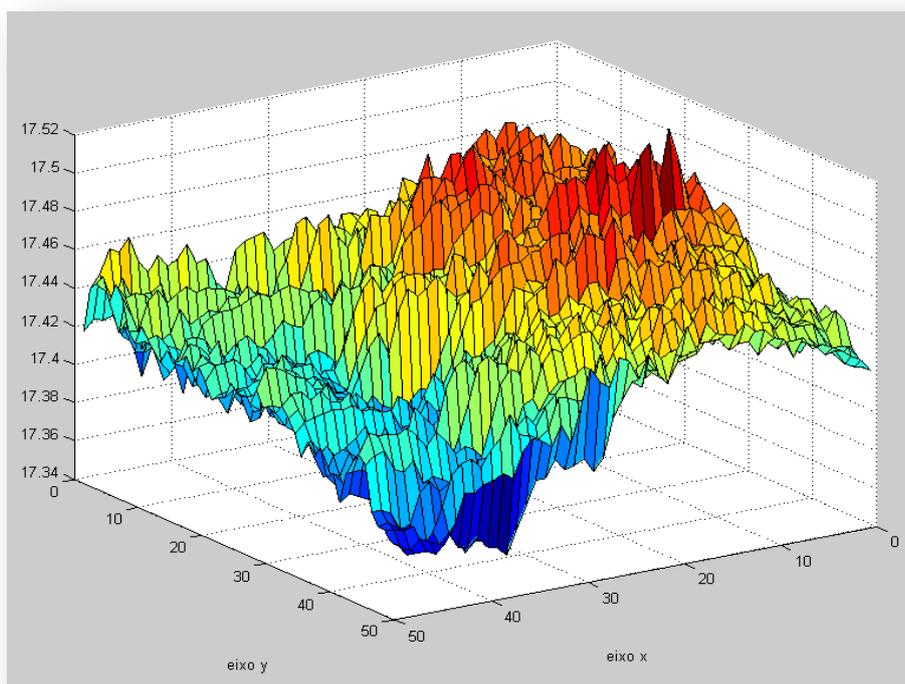


Figura 45 - amostra um - quarta fatia

Fonte: Própria

6.5.2. SEGUNDO TESTE DA REPETIÇÃO DE PADRÕES ENTRE MEDIÇÕES

Neste segundo teste, foi amostrada uma superfície de cobre com as mesmas características (no tipo de material e preparo antes da leitura) da utilizada no primeiro teste. A *Figura 46* representa uma amostra realizada no período da noite, por volta das 23 horas, para minimizar possíveis vibrações. É possível perceber uma estrutura parecida com um vale geográfico.

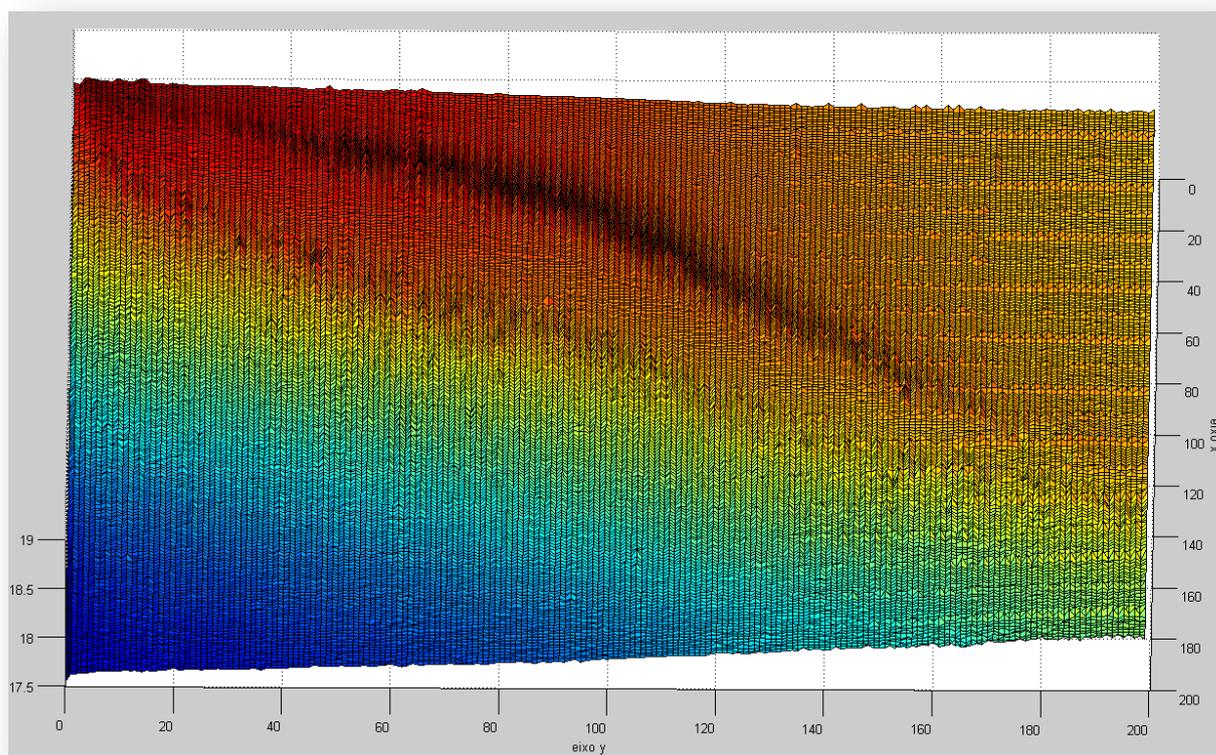


Figura 46 – amostra dois - dia anterior

Fonte: Própria

Após a aquisição dessa imagem, foram desligados os equipamentos. No dia seguinte, por volta das 8 horas da manhã, uma segunda amostra foi realizada, da mesma superfície sem qualquer alteração na configuração mecânica e elétrica do STM. A representação dessa imagem está na Figura 47.

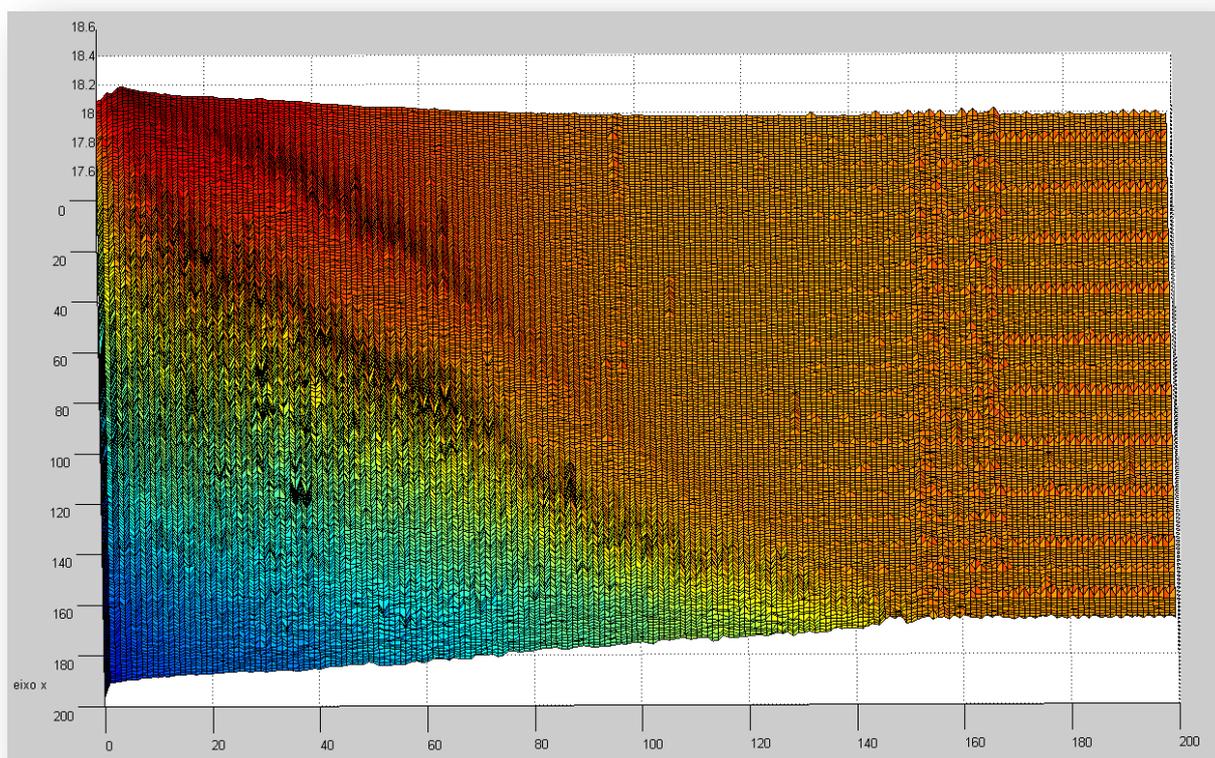


Figura 47 - amostra dois - dia posterior

Fonte: Própria

A correlação entre as duas é visivelmente perceptível. O vale presente na *Figura 46* reaparece levemente deslocado na *Figura 47*. O interesse desse teste é que existe um intervalo de pelo menos 8 horas entre as duas medições, o que demonstra o princípio da invariância no tempo aplicado a essas medições.

A *Figura 48* consiste na *Figura 47* deslocada na angulação da câmera no *software* reconstrutor da imagem. O objetivo é facilitar a visualização e comparação com as três figuras: a *Figura 49*, *Figura 50* e *Figura 51*. Elas estão na sequência, supondo uma leitura em fatias da imagem total, e também comprovam mais uma vez uma repetição de padrão superficial entre as amostras.

Pode-se ver claramente que o mesmo vale presente nas figuras 200 x 200 aparece recortado em posições correlacionadas com a imagem total. Ou seja, a primeira fatia mostra a parte mais elevada, a segunda uma região intermediária e a última, o final do “vale”.

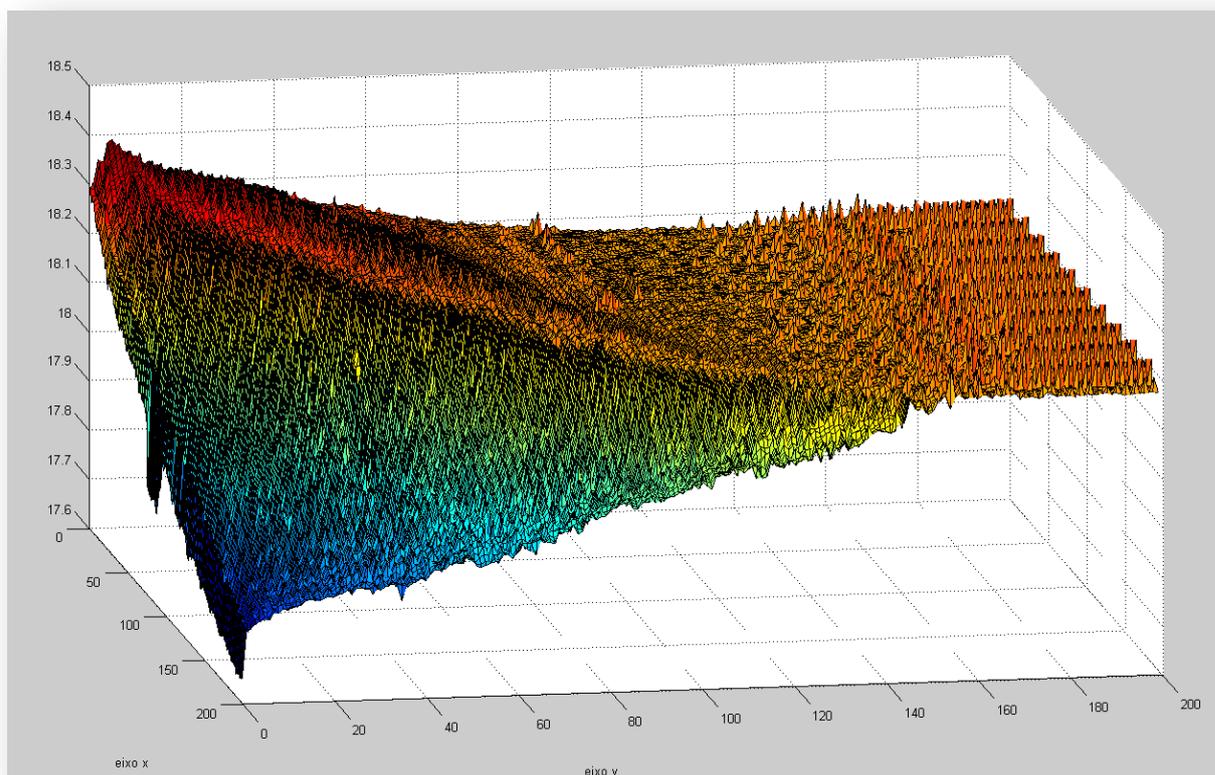


Figura 48 - amostra dois - vista inclinada

Fonte: Própria

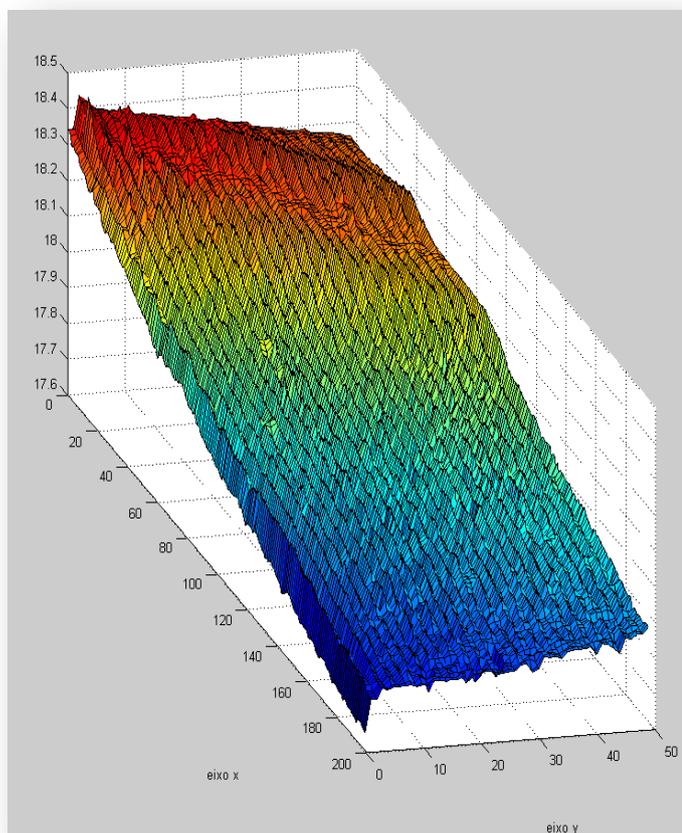


Figura 49- amostra dois - fatia da esquerda

Fonte: Própria

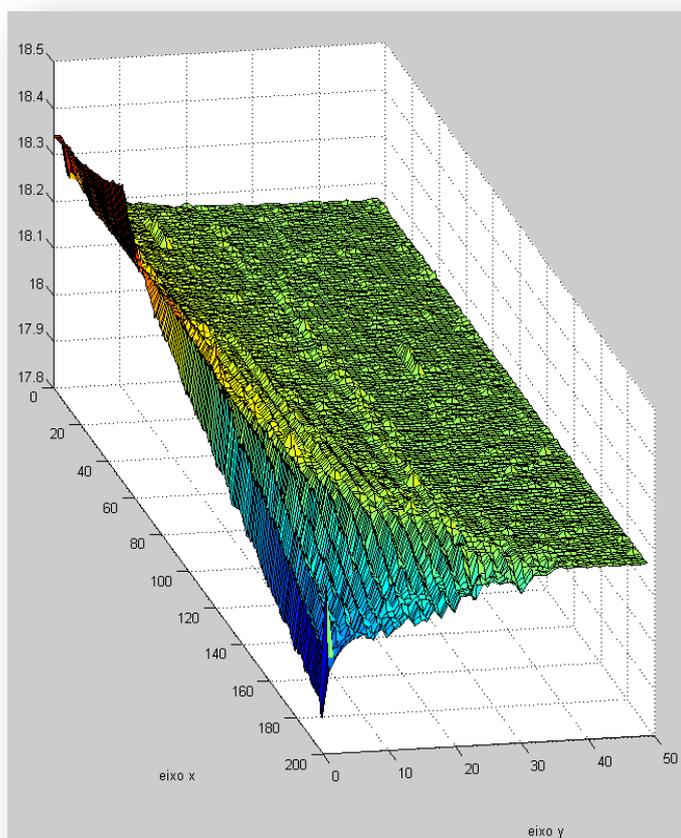


Figura 50 - amostra dois - fatia central

Fonte: Própria

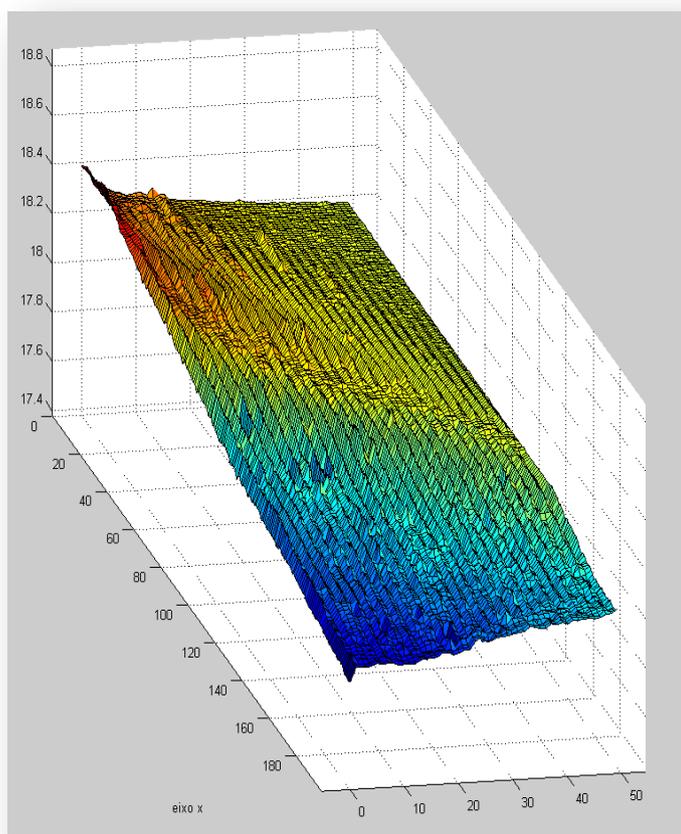


Figura 51 - amostra dois – fatia da direita

Fonte: Própria

6.6. DISCUSSÃO

Uma das grandes dificuldades encontradas foi a comprovação das imagens obtidas através do microscópio de tunelamento. A solução ideal seria realizar a varredura de uma amostra em que o padrão superficial fosse conhecido. Um exemplo é o HOPG (*Highly ordered pyrolytic graphite*), que possui uma superfície bem definida e plana, utilizada para calibração desse tipo de microscópio. Entretanto dois problemas impediram o uso do HOPG para comprovação das imagens: a questão financeira, e a questão do tempo de entrega. Foi encontrado HOPG apenas em sites chineses com prazos de entrega de cerca de um mês, o que inviabilizou sua compra por conta do cronograma adotado no projeto.

Outra possível solução é a utilização de outro STM para fazer a leitura de uma superfície qualquer e gerar uma imagem que pudesse ser comparada com a gerada pelo STM construído nesse projeto. Entretanto não foi encontrado um STM disponível para que se pudesse utilizar para comprovação.

Com o auxílio de vários profissionais como físicos, engenheiros mecânicos e engenheiros eletrônicos foram desenvolvidas teorias e elaboradas possíveis experiências para comprovação dessas imagens.

Algumas das possibilidades escolhidas como tentativas de comprovação da imagem foram as seguintes:

- Depósito de solução aquosa contendo nanopartículas de ouro em mica.

Uma solução de ouro e água preparada pelo Prof. Arandi Ginane Bezerra Junior foi depositada em um pedaço de mica. Foram depositadas várias camadas da solução na mica, com um tempo de espera de 4 horas entre uma camada e outra para que a água da solução secasse e ficassem depositadas apenas partículas de ouro. O sistema não apresentou resultados satisfatórios devido a problema de condutividade da superfície construída.

- Depósito de solução de ouro e água em cobre.

Outra tentativa para se resolver o problema de condutividade citado acima foi a solução de nanopartículas de ouro em água depositada em uma superfície de cobre. O sistema apresentou falhas devido à oxidação gerada pela água existente na solução, em pouco tempo aparecendo uma camada isolante entre partes da amostra e a sonda. Isso confunde o sistema de controle, pois quando a sonda está sobre uma superfície oxidada a corrente de tunelamento tende a zero, e a malha de controle tende a abaixar mais a sonda, o que pode causar colisão da

sonda com a amostra, fato que, ao final do processo, gera uma imagem que não corresponde à realidade.

- Superfície de aço preparada pelo Prof. Julio Cesar Klein das Neves do grupo de materiais do departamento de mecânica da UTFPR. E superfície de placa virgem de cobre para montagem de circuitos eletrônicos.

Foram utilizadas duas superfícies que geraram imagens distintas. Essas superfícies de cobre e aço foram as mais utilizadas para gerar imagens, entretanto como ambas não tem padrões anteriormente conhecidos, essas imagens não puderam ser comprovadas no quesito resolução (dimensões ou padrões conhecidos).

Reconhece-se que existe uma dificuldade metodológica na prova das imagens geradas pelo resultado deste trabalho na questão da precisão da escala de medição. Em parte por se trabalhar numa região incomum tanto para microscópios óticos quanto eletrônicos convencionais. Há necessidade de se pesquisar através de trabalhos futuros uma maneira de comprovação de tais imagens seja com estruturas nanométricas (ou padrões nanométricos) previamente conhecidos utilizados para comparações.

Outros possíveis trabalhos futuros:

- Construção de estrutura mecânica mais robusta para o microscópio de tunelamento quanto à praticidade para colocação/retirada de substrato em análise;
- Construção de aparelhagem anti-vibração que permita um maior isolamento da estrutura mecânica do equipamento do ambiente do laboratório;
- Aprimoramento do sistema de controle para que se possa reduzir ainda mais o ruído mecânico e eletromagnético que de maneira esporádica aparece durante algumas medições.

6.7. CONSIDERAÇÕES

Dentre os assuntos abordados neste capítulo de resultados, pode-se concluir que a construção do *hardware* tanto físico quanto eletrônico foi executada com sucesso. E que nenhuma grande adaptação ou mudança precisou ser realizada para que se pudesse atingir o funcionamento planejado do sistema.

Também é possível afirmar que a comprovação da corrente por tunelamento foi satisfatória no momento em que se pôde verificar a sua característica exponencial, e a partir desse ponto se realizar a sua linearização e comprovação de fato que se trata de uma corrente com este perfil.

Através dos testes da deflexão do piezelétrico pode-se calcular de maneira aproximada a variação no eixo cartesiano Z do disco. A partir desse cálculo foi possível conhecer as deformações superficiais presentes na amostra de cobre, dentre outras amostras utilizadas, com uma precisão razoável.

Finalmente, na comprovação e validação da imagem da superfície amostrada não foi possível encontrar um padrão em tempo hábil que pudesse ser utilizado como prova. Portanto ainda existe um desconhecimento da precisão da medição executada pelo microscópio construído neste trabalho.

7. PLANO DE NEGÓCIOS

7.1. INTRODUÇÃO

Durante a construção deste plano de negócios, teve-se como objetivo uma prova de conceito para se verificar a viabilidade comercial do produto proposto. A prova de conceito é necessária pela dificuldade de se prever o comportamento do mercado nacional. Não existem indicadores seguros da real demanda deste produto nem concorrentes que pudessem ter a quantidade vendas unitárias utilizadas como padrão para o planejamento da STM Brasil.

Para que haja uma boa homogeneização entre este documento e o relatório de projeto final foi utilizado o mesmo padrão de conclusões parciais (comentários adicionais) após alguns capítulos do mesmo, sendo essa a única adaptação realizada.

7.2. RESUMO EXECUTIVO

7.2.1. DEFINIÇÃO DO NEGÓCIO

DESCRIÇÃO DO NEGÓCIO
A STM Brasil é a empresa criada para a realização de teste de conceito na área de instrumentação eletrônica voltada para nanotecnologia, com atuação nacional e o diferencial de fornecer produtos com alta qualidade e com baixo custo, adequados as necessidades de universidades e laboratórios de pesquisa. O carro chefe da empresa será um microscópio eletrônico de tunelamento por varredura e um <i>software</i> licenciado para geração da imagem. Esse equipamento tem o diferencial de possuir um custo baixo, se comparado aos concorrentes estrangeiros. A empresa tem o intuito de atender as necessidades de universidades e laboratórios envolvidos em pesquisas de superfícies condutoras.
VISÃO 2014
Tornar-se líder no mercado nacional, como referência em qualidade, baixo custo e satisfação do cliente. Com a meta de atender 100 laboratórios brasileiros até 2014.
MISSÃO
A STM Brasil é uma empresa que atua na área de instrumentação eletrônica voltada para nanotecnologia, com atuação nacional, com o diferencial de fornecer produtos com alta qualidade e baixo custo adequados as necessidades de universidades e laboratórios de pesquisa. Trabalhando sempre para buscar respeito do mercado e agregar valores aos funcionários, clientes e fornecedores com o intuito de contribuir para o avanço tecnológico brasileiro e tornar o mundo menos hostil.

Quadro 7 - Definição do negócio

7.2.2. DADOS DA EMPRESA

DADOS DA EMPRESA	
Razão social: STM Brasil Participações	
Nome fantasia: STM Brasil	
Endereço: Av. Sete de Setembro, 3165 – Rebouças	
CEP 80230-901 - Curitiba - PR – Brasil	
Telefone: (41) 3310 – 4696	
Email: stmutfpr@gmail.com	Home-page: http://stmutfpr.wordpress.com

Quadro 8 - Dados da empresa

7.2.3. PRODUTOS E SERVIÇOS

PRODUTOS E SERVIÇOS	
Produto	Microscópio Eletrônico de Baixo Custo de Varredura por Corrente de Tunelamento e <i>software</i> para geração de imagens
Serviços	Assistência técnica e <i>software</i> licenciado
Experiência	Know-How em e sistemas de detecção de corrente de tunelamento

Quadro 9 - Produtos e Serviços

7.2.4. ANÁLISE DE MERCADO

ANÁLISE DE MERCADO	
O mercado nacional de microscópios eletrônicos de baixo custo simplesmente não possui solução nacional para atender as demandas de mercado. Através de análises preliminares é possível observar que existe um grande interesse por produtos que ofereçam soluções na área de nanotecnologia e comprovado baixo custo. A expectativa futura com o lançamento desse produto é de um avanço expressivo sobre o <i>market-share</i> de microscópios em geral no mercado nacional.	

Quadro 10 - Análise de mercado

7.2.5. PERFIL DO CLIENTE

Perfil do Cliente	
Cliente Universitário	Para entender melhor o cliente se faz necessário a divisão em dois tipos principais de clientes: a universidade pública e a universidade privada. Vale lembrar que pelo perfil e dinâmicas diferentes, ambos demandam atenções e ações diferentes.
Características	<ul style="list-style-type: none"> • Universidade Pública: compõe-se de uma estrutura complexa onde os principais integrantes são quatro. Como as compras são feitas através de licitação, os integrantes são: o professor pesquisador interessado no produto, a pessoa responsável pelo departamento de compras, o chefe de departamento e a pessoa responsável pelo processo licitatório. • Universidade Privada: compõe-se de uma estrutura menor que a da universidade pública. A compra é feita de maneira mais direta e os principais integrantes do processo de compra são o professor pesquisador, e o reitor da universidade. <p>Em ambos os casos se faz necessário observar e analisar a questão financeira da instituição. A disponibilidade de verbas é essencial para o processo de compra.</p>
Localização	Universidades em grandes centros urbanos pelo Brasil

Quadro 11 - Perfil do cliente

7.2.6. PLANO DE MARKETING E VENDAS

ESTRATÉGIAS DE MARKETING
<ul style="list-style-type: none"> - Produto: através da análise do resultado do teste de conceito será possível se obter uma melhor avaliação do mercado. - Preço: a partir da pesquisa de mercado será feito o cálculo do custo fixo, variável e lucro pretendido que compõem o preço, que tem seu valor inicial de 7 mil reais. - Lugar: parceria com a UTFPR e com universidades do Brasil para iniciar as vendas. - Promoção: Por meio de congressos, feiras e mídias eletrônicas. - Pessoas: contato próximo com os clientes através das vendas que serão realizadas pelos próprios sócios da empresa.

Quadro 12 - Estratégias de Marketing

META DE VENDAS DO MICROSCÓPIO DE TUNELAMENTO	
Ano	Unidades vendidas por ano
2012	4
2013	20
2014	100

Quadro 13 - Metas de Vendas

7.2.7. PLANO FINANCEIRO

ANO	2012	2013	2014
Receitas (mil R\$)	32	150	580
Despesas (mil R\$)	11,6	55,375	232,25
Lucro (mil R\$)	20,4	94,625	347,75

Quadro 14 - Demonstração resumida de lucro

7.3. VISÃO E MISSÃO

7.3.1. MARCOS RELEVANTES

MARCOS RELEVANTES	
Data	Fato
04/2010	Intercâmbio do Felipe Ribeiro na Alemanha onde houve a transferência de know-how da confecção e operação de sistemas de precisão nanométrica
02/2011	Proposta de desenvolvimento no Brasil de protótipo do microscópio eletrônico com apoio da CAPES
03/2011	Formação de equipe de projeto final para o desenvolvimento do produto
07/11	Defesa da proposta e relatório parcial de projeto final.

Quadro 15 - Marcos Relevantes

7.3.2. VISÃO

VISÃO 2014
Tornar-se líder no mercado nacional, como referência em qualidade, baixo custo e satisfação do cliente. Com a meta de atender 100 laboratórios brasileiros até 2014.

Quadro 16 – Visão

7.3.3. MISSÃO

MISSÃO
A STM Brasil é uma empresa que atua na área de instrumentação eletrônica voltada para nanotecnologia, com atuação nacional, com o diferencial de fornecer produtos com alta qualidade e baixo custo adequados as necessidades de universidades e laboratórios de pesquisa. Trabalhando sempre para buscar respeito do mercado e agregar valores aos funcionários, clientes e fornecedores com o intuito de contribuir para o avanço tecnológico brasileiro e tornar o mundo menos hostil.

Quadro 17 – Missão

7.3.4. GESTÃO DE QUALIDADE

GESTÃO DE QUALIDADE
<ul style="list-style-type: none"> • A empresa está engajada na obtenção da certificação ISO 9000; • O sistema de gestão da qualidade da STM Brasil está dividido entre procedimentos, instruções e controles para garantir a qualidade do processo/produto;

Quadro 18 - Gestão de Qualidade

7.3.5. VALORES

VALORES**Comprometimento**

Para se obter o crescimento como um todo é necessário um comprometimento de todos e principalmente da empresa para manter a harmonia e alcançar os objetivos do coletivo e individual.

Ética e Transparência

Respeitando os clientes, funcionários, fornecedores e concorrentes. Deixando claro o que se pensa e como agir. Sempre deixando transparente o objetivo a qual a empresa.

Inovação

Por ser uma empresa de tecnologia, melhorias contínuas são metas permanentes em todos os setores para fornecer ao cliente final um produto inovador.

Qualidade

A qualidade dos produtos e serviços oferecidos está relacionada ao bem estar e à qualidade de vida das pessoas ligadas direta ou indiretamente com a empresa.

Respeito

As diferentes visões de mundo são respeitadas e a principal crença da empresa é de que a diversidade é a força que promove a inovação empresarial.

Quadro 19 - Valores

7.4. DADOS DA EMPRESA

7.4.1. ASPECTOS JURÍDICOS

ASPECTOS JURÍDICOS	
Razão social:	STM Brasil fabrica de microscópios LTDA.
Nome fantasia:	STM Brasil
Endereço sede:	Av. Sete de Setembro, 3165 – Rebouças.
CEP/Cidade/UF/País	80230-901 Curitiba/Paraná/Brasil.
Telefone:	3310-4696
Home-page e E-mail	http://stmutfpr.wordpress.com stmutfpr@gmail.com

Quadro 20 - Aspectos Jurídicos

COMPOSIÇÃO DO CAPITAL			
Sócio	Quotas	Cargo	Competências
Airton Hitoshi Mitsuyasu	33,3 %	Diretor de Marketing e Projetos	Engenheiro Eletrônico formado na UTFPR, especialista em desenvolvimento de sistemas micro controlados, telecomunicações e automação residencial.
Felipe da Silva Ribeiro	33,3 %	Diretor de P&D	Engenheiro Eletrônico formado na UTFPR, especialista em desenvolvimento de sistemas micro controlados, engenharia biomédica e automação residencial.
Luiz Davi Leitão Martins	33,3 %	Diretor Financeiro e Outsourcing	Engenheiro Eletrônico formado na UTFPR, especialista em desenvolvimento de <i>softwares</i> , telecomunicações e engenharia biomédica.

Quadro 21 - Composição de Capital

7.4.2. CONSELHEIROS

CONSELHEIROS
Pessoas que fazem parte do conselho administrativo da empresa. <ul style="list-style-type: none"> • Fábio Schneider; • Francisco Pichorim;

Quadro 22 – Conselheiros

7.4.3. RESPONSABILIDADE GERENCIAL EQUIPE DE GERENTES

Cargo	Nome	Responsabilidade
Diretor de Marketing e Projetos	Airton Mitsuyasu Hitoshi	Planejamento e comercialização
Diretor de P&D	Felipe da Silva Ribeiro	Pesquisa, inovação e desenvolvimento de projetos.
Diretor Financeiro e Outsourcing	Luiz Davi Leitão Martins	Administração, finanças e RH

Quadro 23 - Equipe gerencial

7.4.4. ADVOGADO E CONTADOR

A empresa conta com serviços terceirizados de empresas especializadas nessas áreas.

7.5. PRODUTOS E SERVIÇOS

7.5.1. PRODUTOS E SERVIÇOS DISPONÍVEIS

7.5.1.1. DESCRIÇÃO DA LINHA DE PRODUTOS

Microscópio de Tunelamento	
Aplicação	Análise de superfícies de materiais condutores
Características, Vantagens e benefícios	<ul style="list-style-type: none"> • Análise de superfícies condutoras com boa relação “resolução X custo”. • Tamanho reduzido, do tamanho de uma caixa de sapatos. • Custo reduzido em relação aos concorrentes estrangeiros, tornando mais acessível aos clientes como universidades públicas e privadas. • Produto de fabricação nacional, facilitando vários aspectos da relação cliente-empresa: a manutenção e suporte técnico dos produtos e a interação com o cliente.
Reprodução, empacotamento, entrega	Fabricação, calibração e testes feitos em um mês. Embalagem feita e caixa de papelão com logotipo da empresa, preenchida com isopor. Entrega feita pessoalmente por responsável técnico.
Garantia e suporte	O produto tem um garantia de três anos com suporte técnico fornecido diretamente pela STM Brasil.

Quadro 24 - Descrição STM

STM Software	
Aplicação	Licença de <i>software</i> de interface com o microscópio e geração de imagens.
Características Vantagens e benefícios	<ul style="list-style-type: none"> • <i>Software</i> de geração de imagem em três dimensões. • Possibilidade de rotacionar a imagem nos três eixos. • Controle do status da USB diretamente no <i>software</i>, facilitando para o usuário perceber conexão e desconexão do equipamento. • <i>Software</i> em apenas um arquivo executável, simplificando instalação por parte do usuário. • Visualização da imagem em qualquer browser moderno.

Quadro 25 - Descrição STM Software

7.5.1.2. CICLO DE VIDA DO PRODUTO

Características	Introdução	Crescimento	Maturidade	Declínio
Vendas	Vendas Baixas	Vendas em crescimento contínuo	Máximo de vendas	Vendas em queda
Custos	Custo médio por consumidor	Custo baixo por consumidor	Custo baixo por consumidor	Custo baixo por consumidor
Desenvolvimento	Muito Ativo	Ativo	Pouco Ativo	Nenhum
Lucros	Negativos	Em crescimento	Altos	Em queda
Consumidores	Inovadores	Adotantes Iniciais	Maioria intermediária	Retardatários
Concorrentes	Poucos e estrangeiros	Aumentando em número	Estável	Em queda
Produto	2011	2013-2015	2016-2020	2021

Quadro 26 - Ciclo de vida dos produtos

7.5.1.3. PRODUTOS E SERVIÇOS PLANEJADOS

Produtos e Serviços Planejados	
Produtos – Acessórios	Kit de sondas e amostras.
Serviços – <i>Software</i>	STM <i>Software</i> Linux.
Serviços- <i>Software</i>	STM <i>Software</i> Mobile.

Quadro 27 - Produtos e Serviços Planejados

7.6. ANÁLISE DE MERCADO

7.6.1. DEFINIÇÃO DE MERCADO

Microscópio de Tunelamento	
Mercado atual	Microscópios de tunelamento de baixo custo praticamente não são comercializados no país
Mercado-alvo nacional	Inicialmente domínio de mercado nos estados do Paraná e Santa Catarina, por serem estados com cerca de 25 universidades potenciais clientes e pela proximidade com a sede da empresa. Então expandir para todo o território nacional
Mercado Potencial	Laboratórios, universidades públicas e privadas com linhas de pesquisa em análise de superfícies e nanotecnologia.
Localização	Grandes centros urbanos brasileiros que possuam universidade com cursos nas áreas de física, engenharia mecânica e/ou eletrônica.
Sazonalidade	Liberação de verbas por parte do governo (CAPES, CNPQ, etc.), que acontece anualmente no mês de agosto. Período pré-eleitoral (aumento das vendas), que acontece a cada 2 anos, mas nos anos em ocorre eleições estaduais e federais(a cada 4 anos) as vendas tendem a aumentar ainda mais. Período eleitoral (queda nas vendas), que acontece a cada dois anos e durante a época eleitoral é proibida liberação de verbas públicas.

Quadro 28 - Definição de mercado

7.6.2. SEGMENTO DE MERCADO

Segmento de Mercado	
Microscópio de Tunelamento	<ul style="list-style-type: none"> Laboratórios de pesquisa e Universidades públicas e privadas com linhas de pesquisa na área de nanotecnologia (principalmente na área de análise de superfícies condutoras) ou experiências com corrente por efeito túnel.

Quadro 29 - Segmento de Mercado

7.6.3. PONTOS FORTES E FRACOS

Pontos Fortes e Fracos		
Fatores	Pontos Fortes	Pontos Fracos
Confiabilidade	→ Empresa pioneira no país → Baixo numero de concorrentes (todos internacionais)	→ Empresa de pequeno porte Transformando a empresa em sinônimo de confiabilidade, a empresa aumenta a carteira de clientes, vai crescendo e se consolidando no mercado.
Credibilidade	→ Baseado em equipamentos similares, como o de John Alexander, expoente em STMs	→ Pouco conhecido no mercado. Ações de marketing junto a potenciais clientes tornarão a empresa mais conhecida
Preço	→ Melhor relação custo benefício do mercado	→ Conhecimento parcial dos preços dos concorrentes
Tecnologia	→ Uso de tecnologias avançadas e consolidadas no mercado, no desenvolvimento de produtos	→ O <i>software</i> roda apenas na plataforma .NET Windows O <i>software</i> será adaptado a outros sistemas operacionais.
Qualidade	→ Utilização de produtos de marcas confiáveis.	→ Não possui certificado de qualidade. Busca de qualificação do tipo ISO 9000

Quadro 30 - Pontos Fortes e Fracos

7.6.4. OPORTUNIDADES E AMEAÇAS EXISTENTES

Oportunidades e Ameças existentes		
Fatores	Oportunidades	Ameças
→ Opções de mercado	→ Possibilidade de atuação em nível nacional	→ Empresa estrangeira se instalar no país com preço competitivo
→ Linha de produtos	→ Expandir a linha de produtos para atender outras demandas dos clientes na área de instrumentação eletrônica	→ Propriedade industrial (pagamento de royalties) → Saturação de determinado produto no mercado
→ Tecnologia	→ Criação de suporte técnico com atendimento online	→ Demanda incompatível com a oferta do serviço

Quadro 31 - Oportunidades e ameaças

7.6.5. PERFIL DO CLIENTE

Perfil do Cliente	
Cliente Universitário	Para entender melhor o cliente se faz necessário a divisão em dois tipos principais de clientes: a universidade pública e a universidade privada. Vale lembrar que pelo perfil e dinâmicas diferentes, ambos demandam atenções e ações diferentes.
Características	<ul style="list-style-type: none"> • Universidade Pública: compõe-se de uma estrutura complexa onde os principais integrantes são quatro. Como as compras são feitas através de licitação, os integrantes são: o professor pesquisador interessado no produto, a pessoa responsável pelo departamento de compras, o chefe de departamento e a pessoa responsável pelo processo licitatório. • Universidade Privada: compõe-se de uma estrutura menor que a da universidade pública. A compra é feita de maneira mais direta e os principais integrantes do processo de compra são o professor pesquisador, e o reitor da universidade. <p>Em ambos os casos se faz necessário observar e analisar a questão financeira da instituição. A disponibilidade de verbas é essencial para o processo de compra.</p>
Localização	Universidades em grandes centros urbanos pelo Brasil

Quadro 32 - Perfil do cliente

7.6.6. CARACTERIZAÇÃO DA CONCORRÊNCIA

Concorrência	
Principais Concorrentes	<ul style="list-style-type: none"> • PHYWE (Alemanha) • Nanosurf (Suíça, Alemanha, China e EUA)
Região de atuação	<ul style="list-style-type: none"> • América do Norte • Europa • Ásia
Posicionamento	É difícil fazer um levantamento concreto sem estar dentro do mercado, dada a peculiaridade do produto.
Pontos Fortes	<p>Maior Experiência na área e maior conhecimento no mercado.</p> <p>Com produtos de qualidade, e ações de marketing a empresa ganha reconhecimento do mercado, aumenta a fatia de mercado e por consequência ganha experiência.</p>

Quadro 33 – Concorrência

7.6.7. RISCOS

Riscos	
Tecnológicos	Riscos: É preciso estar atento a vida útil do microcontrolador, amplificadores operacionais e canal de comunicação. Ações da Empresa: Foram escolhidos chips, tecnologias e fabricantes consolidados no mercado. Mas ha um monitoramento constante da vida útil desses componentes.
Mercadológicos	Riscos: Surgimento de empresas nacionais que possam competir em relação a preço e qualidade. Aumento dos impostos para importação de componentes eletrônicos. Ações da empresa: Procurar sempre maneiras de manter a interatividade para fidelizar o cliente. Sempre buscando alternativas de fabricação nacional.
Financeiros	Riscos: Fluxo de caixa negativo do primeiro ano pode afastar possíveis investidores. Ações da empresa: trabalho junto aos possíveis investidores no sentido de mostrar a qualidade, robustez e confiabilidade do produto, aliado a aceitação do produto no mercado.
Da Especialidade	Pela falta de experiência na área e falta de um mercado consolidado no país, há um desconhecimento do potencial de vendas Brasil.

Quadro 34 - Riscos

7.7. PLANO DE MARKETING E VENDAS

7.7.1. OBJETIVOS E METAS QUANTO ÀS VENDAS E AOS CLIENTES

Objetivo		Prazo
Faturamento	Atingir faturamento de R\$ 250.000,00	2014
Cobertura de mercado	Atingir 80% do mercado nacional	Até 2015
Atualização tecnológica e Qualidade	Manter a qualidade e atualização dos produtos e serviços usando os padrões internacionais	Contínuo

Quadro 35 - Objetivos

Microscópio de tunelamento			
Ano	Vendas anuais	Faturamento (R\$)	% mercado nacional
2012	4	32.000,00	3
2013	12	150.000,00	9
2014	100	580.000,00	75

Quadro 36 - Metas de vendas

7.7.2. ESTRATÉGIAS DE MARKETING

A estratégia de criação da empresa se dará no conceito de teste de conceito, pois como não há o fácil acesso a produtos similares no país, imagina-se que há um grande mercado potencial, mas não há certeza. Portanto todas as estratégias aqui demonstradas foram definidas baseadas no conceito de teste de conceito.

7.7.2.1. PARA O MERCADO NACIONAL

1. Desenvolver novos produtos
Linhas de ação/atividades
1.1 – Pesquisa de Mercado: → A pesquisa de mercado não será feita de maneira direta, mas será parte da análise a ser feita após o término do teste de conceito proposto.
1.2 – Teste de mercado: → A própria abertura da empresa é o teste de mercado pretendido.

Quadro 37 - Estratégia de desenvolvimento de produtos

2. Preparar Material de Divulgação
Linhas de ação/atividades
2.1 – Elaborar material de comunicação: → Serão desenvolvidos um <i>Folder</i> e um <i>Flyer</i> com informações da empresa e do STM para distribuição em feiras e congressos. → Blog para os interessados acompanharem as novidades da empresa. → Criação de um perfil na rede social <i>Linked-In</i> para abordagem de possíveis clientes

Quadro 38 - Material de divulgação

3. Equipe de Vendas
Linhas de ação/atividades
3.1 – Designar responsável pela equipe de vendas: → As vendas serão conduzidas pelos próprios sócios da empresa.
3.2 – Estabelecer metas de vendas: As metas de vendas foram estabelecidas como pode ser visto no tópico 6.1 “OBJETIVOS E METAS QUANTO ÀS VENDAS E AOS CLIENTES”
3.3– Acompanhamento de resultados: → Os resultados serão acompanhados mês a mês com geração de relatórios parciais.
3.4 – Feedback da equipe de vendas: → A partir dos relatórios parciais serão realizadas reuniões bimestrais para avaliar e possivelmente corrigir as estratégias de vendas e pós-vendas

Quadro 39 - Equipe de Vendas

4. Desenvolver canais de comercialização
Linhas de ação/atividades
4.1 – Identificar os canais potenciais → Os canais definidos são: <ul style="list-style-type: none"> ○ Congressos e feiras de área ○ Internet → Serviços necessários para a comercialização (produção local, suporte técnico) → Elaboração dos “fact-sheets”: informações relevantes para o canal sobre a empresa e o produto.
4.2 – Definir os serviços requeridos dos canais: → Stands de divulgação nas feiras especializadas em nanotecnologia. → O blog e o perfil na rede social <i>Linked-In</i> manterão o contato com os clientes trazendo novidades, promoções e trarão informações relevantes aos interessados em se tornarem novos clientes.
4.5 – Suporte aos canais contratados: → O blog deve ser atualizado semanalmente no mínimo, enquanto o perfil do <i>Linked-In</i> deve ser acessado diariamente buscando potenciais clientes.

Quadro 40 - Canais de comercialização

5. Estabelecer parcerias estratégicas
Linhas de ação/atividades
5.1 – Identificação de parceiros: → UTFPR cedeu espaço, equipamentos, e acesso aos conhecimentos necessários ao desenvolvimento do projeto. Mais especificamente o parceiro é o programa de pós-graduação de engenharia elétrica e informática industrial (CPGEI).
5.2 – Formalizar parceria: → A parceria foi formalizada de maneira verbal entre as partes envolvidas. → O CPGEI forneceu o laboratório, equipamentos e parte da verba gasta no projeto → A equipe forneceu mão de obra técnica e intelectual, e a outra parte da verba necessária ao projeto.
5.3 – Motivar parceiro → A motivação da UTFPR se dá pelas possibilidades de pesquisas abertas pela posse de um STM.

Quadro 41 – Parcerias

6. Estabelecer política de preços
Linhas de ação/atividades
6.1 – Pesquisa de mercado: Preços dos concorrentes foram pesquisados
6.2 – Determinar preço de venda: Analisando os custos fixos, variáveis e a margem de lucro pretendida, o preço do produto inicial é R\$ 7000,00 mas com o aumento da demanda esse preço tende a cair entre cerca de 30% a 40%

Quadro 42 – Políticas de preços

7.8. PLANO FINANCEIRO

7.8.1. INVESTIMENTOS

O investimento inicial fixo será basicamente para instalação de um escritório e equipamentos para desenvolvimento. Não há investimentos para implantação da área física e compras de máquinas para produção, pois esta parte será toda terceirizada.

Investimentos	Reais (R\$)
Obras civis	
Construção	-
Projeto	-
Instalações	
Custo de compra/aluguel	R\$ 3.000,00
Melhorias/reformas	R\$ 8.000,00
Outros	-
Equipamentos	
Móveis, estantes, equipamentos, gôndolas	R\$ 5.000,00
Máquinas e equipamentos	R\$ 10.000,00
Veículos	-
Informática	R\$ 7.500,00
Outros	R\$ 2.000,00
Total	R\$ 35.500,00

Quadro 43 - Investimentos iniciais

7.8.2. CUSTOS FIXOS MENSAIS

Devido à demora do tempo de retorno, os custos fixos mensais são um dos maiores problemas do controle financeiro do projeto.

Custos Fixos	R\$
Retirada dos sócios	R\$ 4.500,00
Encargos sociais	R\$ 1.530,00
Salários	-
Encargos sociais sobre salários	-
Seguros	R\$ 50,00
Despesas bancárias	R\$ 40,00
Juros	-
Honorários contábeis	R\$ 40,00
Material de expediente	R\$ 200,00
Aluguel	R\$ 500,00
Despesas de viagens	R\$ 300,00
Água	R\$ 20,00
Luz	R\$ 50,00
Telefone	R\$ 50,00
Manutenção e Conservação	R\$ 100,00
Propaganda	R\$ 200,00
Depreciação	R\$ 200,00
Ônibus, taxi e correios	R\$ 150,00
Financiamentos existentes	-
Total	R\$ 7.930,00

Quadro 44 - Custos fixos mensais

7.8.3. IMPOSTOS

Impostos	%	
ICMS	12,00%	Valor da venda
ISS	5,00%	Valor da venda
CONFINS	7,60%	Valor da venda
PIS	1,65%	Valor da venda
Contribuição social sobre o lucro líquido	9,00%	Sobre o lucro líquido
Imposto de renda	15,00%	Sobre o lucro líquido
Simplex	-	
Comissões	-	
Total sobre o valor da venda	26,25%	Valor da venda
Total sobre o lucro líquido	24,00%	Sobre o lucro líquido

Quadro 45 – Impostos

7.8.4. FATURAMENTO

Devido à necessidade da implantação de um novo mercado no cenário brasileiro, o faturamento nos dois primeiros anos será baixo. A empresa estará focada em difundir e divulgar este novo ramo no Brasil.

Discriminação	2012	2013	2014
Unidades vendidas	4	20	100
Preço por unidade	R\$ 8.000,00	R\$ 7.500,00	R\$ 5.800,00
Faturamento	R\$ 32.000,00	R\$ 150.000,00	R\$ 580.000,00
Custo da mercadoria por unidade	R\$ 400,00	R\$ 400,00	R\$ 400,00
Custo da mercadoria vendida	R\$ 1.600,00	R\$ 8.000,00	R\$ 40.000,00
Custo do serviço prestado por unidade	R\$ 400,00	R\$ 400,00	R\$ 400,00
Custo do serviço prestado	R\$ 1.600,00	R\$ 8.000,00	R\$ 40.000,00
Impostos+comissões: (% de impostos x faturamento)/100 sobre o valor da venda	R\$ 8.400,00	R\$ 39.375,00	R\$ 152.250,00

Quadro 46 – Faturamento

7.8.5. MARGEM DE CONTRIBUIÇÃO

Discriminação	2012	2013	2014
Faturamento	R\$ 32.000,00	R\$ 150.000,00	R\$ 580.000,00
(-) custo da mercadoria vendida	R\$ 1.600,00	R\$ 8.000,00	R\$ 40.000,00
(-) custo do serviço prestado	R\$ 1.600,00	R\$ 8.000,00	R\$ 40.000,00
(-) impostos + comissões	R\$ 8.400,00	R\$ 39.375,00	R\$ 152.250,00
(=) margem de contribuição	R\$ 20.400,00	R\$ 94.625,00	R\$ 347.750,00
Índice da margem de contribuição: (margem de contribuição/faturamento) x100	63,75%	63,08%	59,96%

Quadro 47 - Margem de contribuicao

7.8.6. PONTO DE EQUILÍBRIO

Discriminação	2012	2013	2014
Despesas fixas	R\$ 95.160,00	R\$ 95.160,00	R\$ 95.160,00
Índice da margem de contribuição	63,75%	63,08%	59,96%
Ponto de equilíbrio: (despesas fixas/índice da margem de contribuição) x 100	R\$ 149.270,59	R\$ 150.848,08	R\$ 158.714,02

Quadro 48 - Ponto de equilibrio

7.8.7. LUCRO ESPERADO PARA O FATURAMENTO ESTIMADO

É possível observar na tabela abaixo que os dois primeiros anos a empresa terá um lucro negativo.

Discriminação	2012	2013	2014
Faturamento estimado	R\$ 32.000,00	R\$ 150.000,00	R\$ 580.000,00
Despesas fixas	R\$ 95.160,00	R\$ 95.160,00	R\$ 95.160,00
Lucro: ((faturamento estimado x índice da margem de contribuição)/100) - despesas fixas	-R\$ 74.760,00	-R\$ 535,00	R\$ 252.590,00

Quadro 49 - Lucro estimado

7.8.8. ESTOQUE INICIAL

Estoque inicial	R\$ 10.000,00
------------------------	---------------

Quadro 50 - Estoque inicial

7.8.9. NECESSIDADE DE CAPITAL DE GIRO

ANO 2012

Discriminação	1 mês	2 mês	3 mês
Unidades vendidas	0	0	0
Faturamento	R\$ -	R\$ -	R\$ -
Custo da mercadoria vendida	R\$ -	R\$ -	R\$ -
Custo do serviço prestado	R\$ -	R\$ -	R\$ -
Despesas fixas	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00
Impostos	R\$ -	R\$ -	R\$ -
Saldo final	-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 7.930,00
Saldo acumulado	-R\$ 7.930,00	-R\$ 15.860,00	-R\$ 23.790,00

4 mês	5 mês	6 mês	7 mês	8 mês
0	0	0	0	1
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 8.000,00
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 400,00
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 400,00
R\$ 7.930,00				
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 2.100,00
-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 2.830,00
-R\$ 31.720,00	-R\$ 39.650,00	-R\$ 47.580,00	-R\$ 55.510,00	-R\$ 58.340,00

9 mês	10 mês	11 mês	12 mês	total
2	1	0	0	4
R\$ 16.000,00	R\$ 8.000,00	R\$ -	R\$ -	R\$ 32.000,00
R\$ 800,00	R\$ 400,00	R\$ -	R\$ -	R\$ 1.600,00
R\$ 800,00	R\$ 400,00	R\$ -	R\$ -	R\$ 1.600,00
R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 95.160,00
R\$ 4.200,00	R\$ 2.100,00	R\$ -	R\$ -	R\$ 8.400,00
R\$ 2.270,00	-R\$ 2.830,00	-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 74.760,00
-R\$ 56.070,00	-R\$ 58.900,00	-R\$ 66.830,00	-R\$ 74.760,00	

Quadro 51 - Fluxo de caixa ano de 2012

ANO 2013

Discriminação	1 mês	2 mês	3 mês
Unidades vendidas	0	1	2
Faturamento	R\$ -	R\$ 7.500,00	R\$ 15.000,00
Custo da mercadoria vendida	R\$ -	R\$ 400,00	R\$ 800,00
Custo do serviço prestado	R\$ -	R\$ 400,00	R\$ 800,00
Despesas fixas	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00
Impostos	R\$ -	R\$ 1.968,75	R\$ 3.937,50
Saldo final	-R\$ 7.930,00	-R\$ 3.198,75	R\$ 1.532,50
Saldo acumulado	-R\$ 82.690,00	-R\$ 85.888,75	-R\$ 84.356,25

4 mês	5 mês	6 mês	7 mês	8 mês
0	0	0	0	6
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 45.000,00
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 2.400,00
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 2.400,00
R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00
R\$ -	R\$ -	R\$ -	R\$ -	R\$ 11.812,50
-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 7.930,00	-R\$ 7.930,00	R\$ 20.457,50
-R\$ 92.286,25	-R\$ 100.216,25	-R\$ 108.146,25	-R\$ 116.076,25	-R\$ 95.618,75

9 mês	10 mês	11 mês	12 mês	total
9	1	1	0	20
R\$ 67.500,00	R\$ 7.500,00	R\$ 7.500,00	R\$ -	R\$ 150.000,00
R\$ 3.600,00	R\$ 400,00	R\$ 400,00	R\$ -	R\$ 8.000,00
R\$ 3.600,00	R\$ 400,00	R\$ 400,00	R\$ -	R\$ 8.000,00
R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 95.160,00
R\$ 17.718,75	R\$ 1.968,75	R\$ 1.968,75	R\$ -	R\$ 39.375,00
R\$ 34.651,25	-R\$ 3.198,75	-R\$ 3.198,75	-R\$ 7.930,00	-R\$ 535,00
-R\$ 60.967,50	-R\$ 64.166,25	-R\$ 67.365,00	-R\$ 75.295,00	

Quadro 52 - Fluxo de caixa ano de 2013

ANO 2014

discriminação	1 mês	2 mês	3 mês
Unidades vendidas	3	6	22
Faturamento	R\$ 17.400,00	R\$ 34.800,00	R\$ 127.600,00
Custo da mercadoria vendida	R\$ 1.200,00	R\$ 2.400,00	R\$ 8.800,00
Custo do serviço prestado	R\$ 1.200,00	R\$ 2.400,00	R\$ 8.800,00
Despesas fixas	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00
Impostos	R\$ 4.567,50	R\$ 9.135,00	R\$ 33.495,00
Saldo final	R\$ 2.502,50	R\$ 12.935,00	R\$ 68.575,00
Saldo acumulado	-R\$ 72.792,50	-R\$ 59.857,50	R\$ 8.717,50

4 mês	5 mês	6 mês	7 mês	8 mês
25	25	0	0	8
R\$ 145.000,00	R\$ 145.000,00	R\$ -	R\$ -	R\$ 46.400,00
R\$ 10.000,00	R\$ 10.000,00	R\$ -	R\$ -	R\$ 3.200,00
R\$ 10.000,00	R\$ 10.000,00	R\$ -	R\$ -	R\$ 3.200,00
R\$ 7.930,00				
R\$ 38.062,50	R\$ 38.062,50	R\$ -	R\$ -	R\$ 12.180,00
R\$ 79.007,50	R\$ 79.007,50	-R\$ 7.930,00	-R\$ 7.930,00	R\$ 19.890,00
R\$ 87.725,00	R\$ 166.732,50	R\$ 158.802,50	R\$ 150.872,50	R\$ 170.762,50

9 mês	10 mês	11 mês	12 mês	total
8	2	1	0	100
R\$ 46.400,00	R\$ 11.600,00	R\$ 5.800,00	R\$ -	R\$ 580.000,00
R\$ 3.200,00	R\$ 800,00	R\$ 400,00	R\$ -	R\$ 40.000,00
R\$ 3.200,00	R\$ 800,00	R\$ 400,00	R\$ -	R\$ 40.000,00
R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 7.930,00	R\$ 95.160,00
R\$ 12.180,00	R\$ 3.045,00	R\$ 1.522,50	R\$ -	R\$ 152.250,00
R\$ 19.890,00	-R\$ 975,00	-R\$ 4.452,50	-R\$ 7.930,00	R\$ 252.590,00
R\$ 190.652,50	R\$ 189.677,50	R\$ 185.225,00	R\$ 177.295,00	

Quadro 53 - Fluxo de caixa ano de 2013

7.8.10. INVESTIMENTO NECESSÁRIO

Discriminação	Reais (R\$)
Investimento	R\$ 35.500,00
Capital de giro	R\$ 116.076,25
Investimento+capital de giro	R\$ 151.576,25

Quadro 54 - Investimento necessario

7.8.11. TAXA DE RENTABILIDADE

Discriminação	Reais (R\$)
Investimento	R\$ 151.576,25
Lucro	R\$ 177.295,00
Taxa de rentabilidade no prazo de 3 anos (lucro/investimento*100)	16,97%

Quadro 55 – Rentabilidade

7.8.12. PRAZO DE RETORNO

Discriminação	Reais (R\$)
Investimento	R\$ 151.576,25
Lucro	R\$ 177.295,00
Prazo de retorno	2,5648
Prazo de retorno real	2 anos e 4 meses

Quadro 56 - Prazo de retorno

7.8.13. ANÁLISE PELO VPL (VALOR PRESENTE LÍQUIDO)

Discriminação	Reais (R\$)
Investimento necessário	R\$ 151.576,25
Taxa de juros anuais	11,00%
Taxa de juros mensais	0,87%
Valor Presente Líquido	R\$ 125.338,84

Quadro 57 - VPL

7.9. CONSIDERAÇÕES

Pelo fato de ser um mercado novo no Brasil, analisando para os três primeiros anos, a criação de uma nova empresa neste ramo não é viável economicamente e financeiramente, pelo fato de que os dois primeiros anos o objetivo é consolidar este novo tipo de mercado sem ter muita preocupação com a venda. É possível ver a inviabilidade pela taxa de rentabilidade está em torno de 16,97% em 3 anos, valor abaixo da taxa Selic e até mesmo do juros da poupança. Pode-se verificar também pela análise do VPL, que é de R\$ 125.338,84, abaixo do valor de investimento. O ponto crítico do projeto neste período é meados do segundo ano (julho de 2013) em que o saldo acumulado é de -R\$ 116.076,25. No entanto se as estimativas do terceiro ano se mantiverem nos próximos dois, a empresa se tornará viável. Pensando em um ciclo de vida para este produto de 7 anos e trabalhando para a criação de novos produtos baseado na experiência adquirida no STM, é possível concluir que este mercado é um ramo promissor que tem um alto potencial de crescimento e rentabilidade. É um investimento de alto risco, por isso a empresa se encaixa na criação através do teste de conceito.

8. CONCLUSÃO FINAL

Em 1981, o microscópio de tunelamento por varredura (STM) foi inventado na IBM em Zurique na Suíça. Invenção que rendeu o prêmio Nobel ao seus inventores em 1986. Esse aparelho deu origem a um novo segmento de aparelhos de manipulação e instrumentos de visualização em escala nanométrica. Segundo informações da Agência Brasil, o mercado brasileiro de produtos com base em nanotecnologias desenvolvidas originalmente no país, somou no ano passado cerca de R\$ 115 milhões, posicionando o Brasil como responsável por apenas 0,03% da produção mundial de nanotecnologia (AGÊNCIA BRASIL 2011).

Baseado nesses dados e utilizando o conhecimento adquirido no intercâmbio na universidade *Beuth Hochschule für Technik* em Berlim, verificou-se a possibilidade e a necessidade do desenvolvimento de um STM de baixo custo na UTFPR. O custo desse microscópio ficou cerca de 50% mais barato que os equipamentos concorrentes pesquisados. Dessa forma, abrirá a possibilidade de serem criadas linhas de pesquisa na área de nanotecnologia em laboratórios da universidade, que antes não o teriam por questões econômicas, além de contribuir para o aumento da produção nacional de nanotecnologias.

Com o estudo dos efeitos físicos envolvidos e características de funcionamento de um STM, como o efeito túnel e as maneiras de realizar varreduras, foi possível o estabelecimento de parâmetros de funcionamento que o STM deveria ter. Como por exemplo, ficou escolhido que o modo de operação a ser utilizado no projeto é o modo de corrente constante.

Durante a pesquisa de patentes foram encontradas inúmeras patentes que são compatíveis com nosso projeto. Foram selecionadas quatro patentes relevantes, por serem semelhantes ao projeto que estamos desenvolvendo, e as informações contidas nessas patentes podem auxiliar no desenvolvimento do projeto.

A análise da concorrência mostrou que a maioria das empresas fabricantes de microscópios está localizada na Europa, em países como Alemanha, Suíça e Dinamarca. Já a pesquisa de patentes mostrou que os inventores são em sua maioria orientais ou descendentes. Também foi constatado que não existem fabricantes nacionais de microscópios de tunelamento. Percebe-se que este produto possui um nicho de mercado, e sendo de fabricação nacional o produto consegue ser mais barato e por consequência mais atrativo aos interessados.

A especificação do projeto uniu o baixo custo à agilidade de desenvolvimento, através de soluções. Para facilitar e agilizar o desenvolvimento do projeto, uma vez que o cronograma foi planejado para menos de um ano.

Para o desenvolvimento do *hardware* eletrônico, buscou-se componentes já conhecidos de maneira a encurtar o tempo de aprendizado de uma nova tecnologia ou componente, sem deixar de lado o baixo custo.

Foram implementados dois métodos de controle, um analógico do tipo PI e um digital baseado no analógico pelo método bilinear. Depois da comparação entre os dois métodos optou-se pelo método de controle digital.

Foram desenvolvidos dois *firmwares* distintos para o microscópio de tunelamento por varredura, um *firmware* para o controle digital e outro para o controle analógico. Apesar desta complexidade, ela se mostrou mais eficiente.

Para a comunicação entre o *firmware* e o *software* instalado no computador foi desenvolvido um pequeno protocolo que permitisse o envio de parâmetros, a troca de informações sobre o início, fim, contagem de pacotes recebidos, dentre outras informações.

Em relação ao projeto de *software*, foi desenvolvida uma solução baseada em *software* para PC e serviço WEB para a recepção e reprodução das imagens amostradas pelo STM, respectivamente. Este segundo contou com a compilação de módulo customizado a partir de um algoritmo desenvolvido nesse projeto para o processamento de imagem escrito em linguagem MATLAB.

Analisando a gestão do programa é possível afirmar que foram trabalhadas 1427 horas no total de 1413 horas estimadas. Esse valor corresponde a 100% do projeto concluído. Tempo esse correspondente às seguintes atividades: pesquisa e o estudo do efeito túnel e tecnologias que podem ser aplicados no projeto, documentação e relatório para defesa do TCC e as etapas da construção mecânica e desenvolvimento do *hardware*, *firmware* e *software*.

Foram gastos R\$750 dos R\$1500,00 previstos, que correspondem a 50% do total. O cronograma foi concluído em adiantamento de duas semanas do prazo de conclusão final.

Através do plano de negócios, se manterem as expectativas para os próximos 3 anos, o projeto é inviável financeiramente e economicamente pelo fato de ser um novo mercado no Brasil. No entanto se as estimativas do terceiro ano se mantiverem nos próximos dois, a empresa se tornará viável. Pensando em um ciclo de vida para este produto de sete anos e

trabalhando para a criação de novos produtos baseado na experiência adquirida no STM, é possível concluir que este mercado é um ramo promissor que tem um alto potencial de crescimento e rentabilidade. É um investimento de alto risco, por isso a empresa se encaixa na criação através do teste de conceito.

Pode-se dizer que as realizações de parte mecânica, do circuito eletrônico e *software* foram executadas com sucesso. E não foram necessárias adaptações ou mudanças para atingir o funcionamento planejado.

A comprovação da corrente por tunelamento foi o resultado que demonstrou a característica exponencial da corrente lida pelo equipamento, e deu a certeza de que o projeto estava no rumo correto. Através dos testes da deflexão do piezolétrico foi possível perceber sua utilidade na varredura do sistema com precisão razoável.

A validação da imagem da superfície amostrada não foi possível encontrar um padrão em tempo hábil que pudesse ser utilizado como prova. Contudo, verificou-se repetitividade nas amostras feitas em curto período de tempo, uma vez que a vibração mecânica modifica o arranjo da estrutura do microscópio e a área de varredura se altera.

Esse projeto foi de extrema importância na formação acadêmica dos integrantes da equipe, principalmente por envolver tantas áreas distintas da engenharia eletrônica, sedimentando grande parte dos conceitos vistos durante o curso.

É importante ressaltar que o projeto conciliou o lado acadêmico ao lado empreendedor de maneira sólida. Fato que mostrou ao grupo a importância da preocupação com essas duas abordagens no desenvolvimento de projetos.

9. REFERÊNCIAS BIBLIOGRÁFICAS

AGÊNCIA BRASIL. *Brasil produz 0,03% da nanotecnologia mundial*. 21. Fevereiro 2011. Disponível em:

<<http://www.inovacaotecnologica.com.br/noticias/noticia.php?artigo=nanotecnologia-no-brasil>> (Acesso em 29. Abril 2011).

ATCP Engenharia Física. „Apostila 3 da fase 1 Materiais Piezoelétricos.“ *Apostila do Curso Materiais e Dispositivos Piezoelétricos: Fundamentos e Desenvolvimento, São Carlos*. 2004. Disponível em: <http://www.atcp.com.br/images/stories/products/ceramicas/Apostila3_curso-piezo-ATCP.pdf> (Acesso em 18. Novembro 2011).

AXELSON, Jan. *USB Complete The Developer's Guide Fourth Edition*. Madison: Lakeview Research LLC, 2009.

BINNIG, G., und H ROHRER. „Scanning Tunneling Microscopy.“ *Surface Science* 126, 1983: 236-244.

CATSOULIS, John. *Designing Embedded Hardware*. Sebastopol: O'Reilly, 2005.

DUARTE, Fabiano Carvalho. „Microscópio de Tunelamento com Varredura (STM) e Microscópio de Força Atômica.“ *Seminário de Medidas para Caracterização e Análise de Materiais do Prof. Furio Damiani e Peter Jurgen Tatsch da Universidade de Campinas (UNICAMP)*. 2000. Disponível em: <http://www.dsif.fee.unicamp.br/~furio/IE607A/STM_AFM.pdf> (Acesso em 18. Novembro 2011).

FONSECA, José A. „Interfaces e Periféricos.“ *Apostila da aula teórica da disciplina de Interfaces e Periféricos - Departamento de Eletrônica da Universidade de Aveiro, Portugal*. 2003. Disponível em: <http://www.ieeta.pt/~jaf/Apoio_IP/Apontamentos_JAF/USB.pdf> (Acesso em 18. Novembro 2011).

GORTAN, Antônio. „Arq. e Org. Comp - Prof. Gortan - 11. Aula – USB.“ *Apresentação de slides da disciplina Arquitetura de Computadores – Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba*. 2010. Disponível em: <<http://pessoal.utfpr.edu.br/gortan/Arquitetura%20e%20Organiza%e7%e3o%20de%20Computadores/Transpar%eancias/USB/USB.pdf>> (Acesso em 18. Novembro 2011).

JANECZKO, César. „Transformação Bilinear.“ *Material de apoio da disciplina Processamento Digital de Sinais (curso de Engenharia Eletrônica) – Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba*. sem data.

<http://www.pessoal.utfpr.edu.br/janeczko/index_files/pds_ee/transformacao_bilinear.pdf>
(Acesso em 18. Novembro 2011).

LAMAS, T.E. „Espectroscopia a Nível Atômico Usando um Microscópio de Tunelamento (STM).“ *Dissertação (Mestrado em Ciências) - Instituto de Física da Universidade de São Paulo*. São Paulo, 1999.

LIMA, Carlos Raimundo Erig. „Métodos de Discretização.“ *Material de apoio da disciplina de Controle e Servomecanismos II - Universidade Tecnológica Federal do Paraná (UTFPR), Curitiba*. kein Datum. Disponível em: <<http://pessoal.utfpr.edu.br/erig/CTRL02/2-%20%20m%e9todos%20de%20discretiza%e7%e3o.pdf>> (Acesso em 18. Novembro 2011).

LINDSAY, S.M. *Introduction to Nanoscience*. Arizona State University: Oxford University Press, 2008.

MICROCHIP Technology Inc. kein Datum. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/spi.pdf>> (Acesso em 18. Novembro 2011).

—. „MCP4822/4812/4822 8/10/12-Bit Dual Voltage Output Digital-to-Analog Converter with Internal VREF and SPI Interface Datasheet.“ 2010. Disponível em: <<http://ww1.microchip.com/downloads/en/devicedoc/22249a.pdf>> (Acesso em 18. Novembro 2011).

—. „PIC18F2455/2550/4455/4550 Data Sheet 28/40/44-Pin High-Performance, Enhanced Flash USB Microcontrollers with nanoWatt Technology.“ 2004. Disponível em: <<http://www.create.ucsb.edu/~dano/CUI/PIC18F4550datasheet.pdf>> (Zugriff am 18. Novembro 2011).

NATIONAL Semiconductor Corporation. „LF411 Low Offset, Low Drift JFET Input Operational Amplifier Datasheet.“ Agosto 2000. <<http://www.national.com/ds/LF/LF411.pdf>> (Acesso em 20. Novembro 2011).

PARK SYSTEMS. *Scanning Tunneling Microscopy (STM)*. kein Datum. Disponível em: <http://www.nanowerk.com/nanobusiness/showroom/Park_Systems/downloads/Scanning_Tunneling_Microscopy.pdf> (Acesso em 2011. Novembro 2011).

PHYWE physics. *Compact-Scanning Tunneling Microscope, complete set incl. tools, sample kit and consumables, in aluminium case*. sem data. Disponível em: <<http://www.phywe.com/461/pid/27088/Kompakt-Rastertunnelmikroskop,-Komplettset-inkl->

Werkzeug,-Probenset-und-Verbrauchsmaterial,-im-Aluminiumkoffer.htm> (Acesso em 29. Abril 2011).

PINO, Alexandre Visintainer, und Marcio Nogueira de SOUZA. „Transdutores Piezeelétricos.“ *Apostila (Medição de Fenômenos Biológicos) - Programa de Engenharia Biomédica (PEB)*. 2011. Disponível em <http://www.peb.ufrj.br/cursos/COB783/Cob783_Apostila7.pdf> (Acesso em 18. Novembro 2011).

SolidMatrix Technologies, Inc. 2005. Disponível em: <<http://tools.ietf.org/html/rfc4180>> (Acesso em 18. Novembro 2011).

USB-IF, Inc. „ Universal Serial Bus (USB) Device Class Definition for Human Interface Devices (HID Specification).“ 2001. Disponível em: <http://www.usb.org/developers/devclass_docs/HID1_11.pdf> (Acesso em 18. Novembro 2011).

WIMAR. *The Code Project*. 22. Março 2007. <http://www.codeproject.com/KB/cs/USB_HID.aspx> (Acesso em 18. Novembro 2011).

WONG, Stephen. *Subversion Source Control General Information*. 2010. <<http://www.clear.rice.edu/comp310/Eclipse/Subclipse/subversion.html>> (Acesso em 19. Novembro 2011).

Anexo A

Hardware do microscópio de tunelamento por varredura.

HARDWARE STM

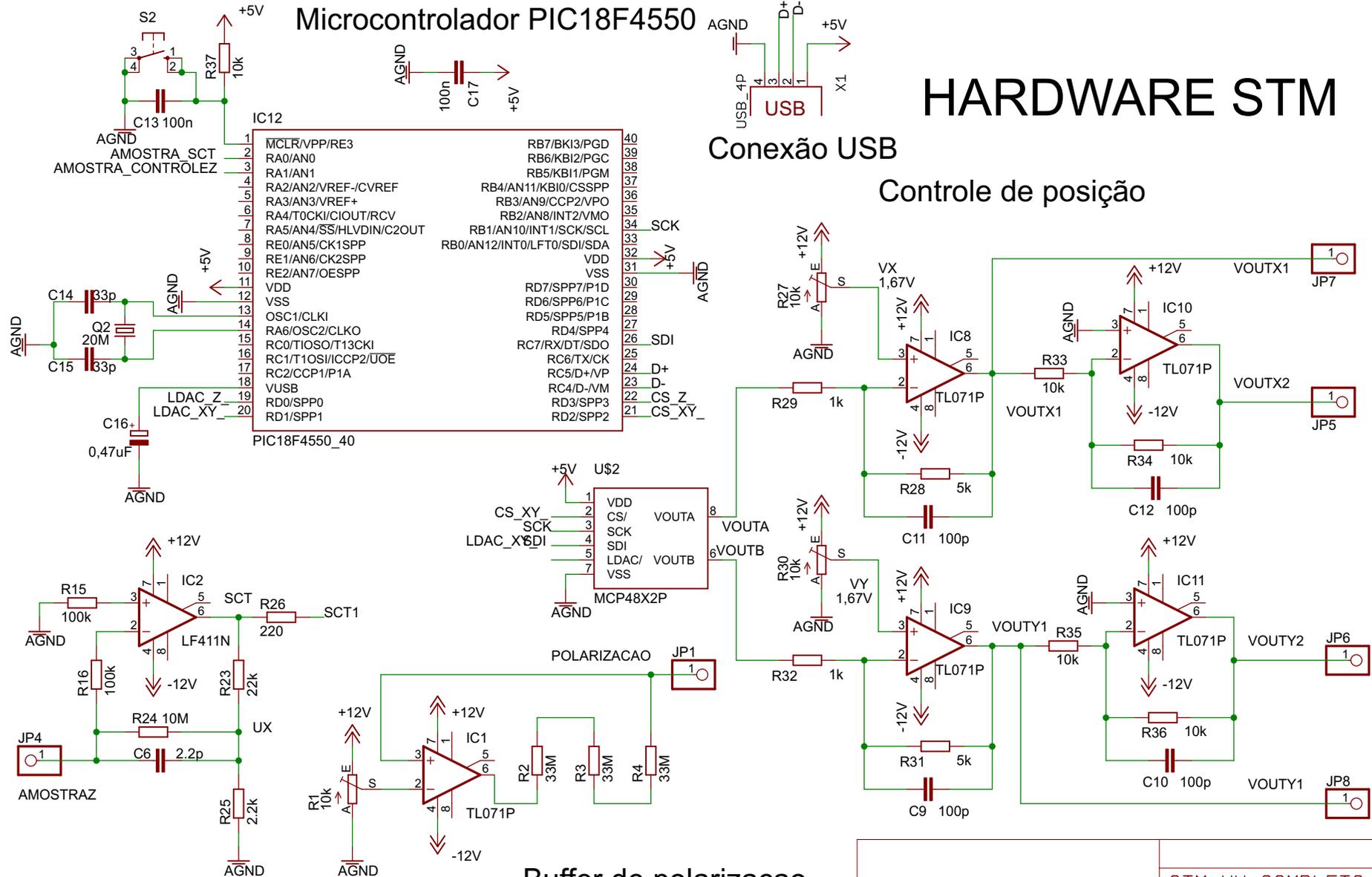
Microcontrolador PIC18F4550

Conexão USB

Controle de posição

Buffer de polarização

Amplificador de Transimpedância



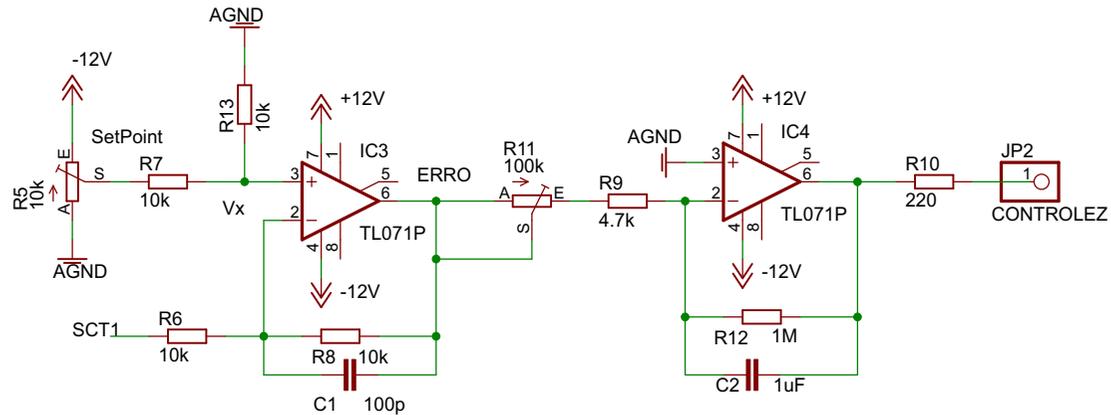
STM HW COMPLETO
20/11/2011 15:10:42
Sheet: 1/3

Anexo B

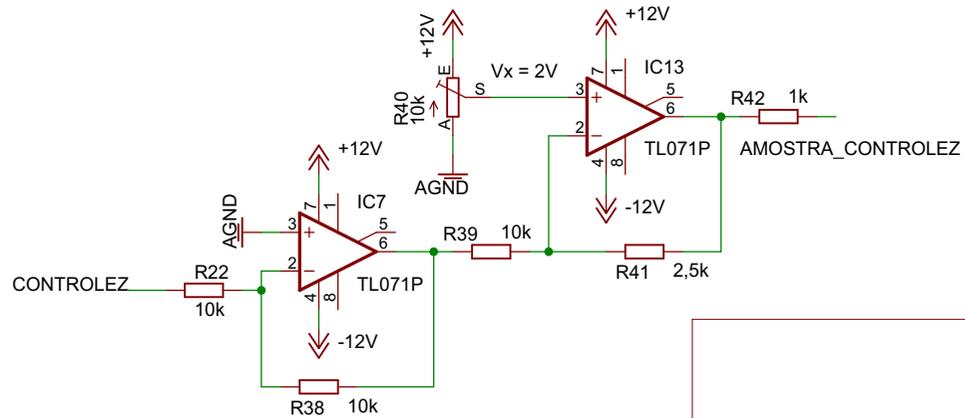
Hardware do controle analógico

HARDWARE CONTROLE ANALÓGICO

Malha de controle PI analógica



Condicionamento do sinal CONTROLEZ



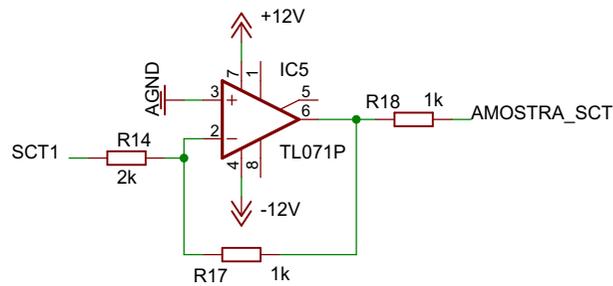
STM HW COMPLETO
20/11/2011 15:10:42
Sheet: 2/3

Anexo C

Hardware do controle digital.

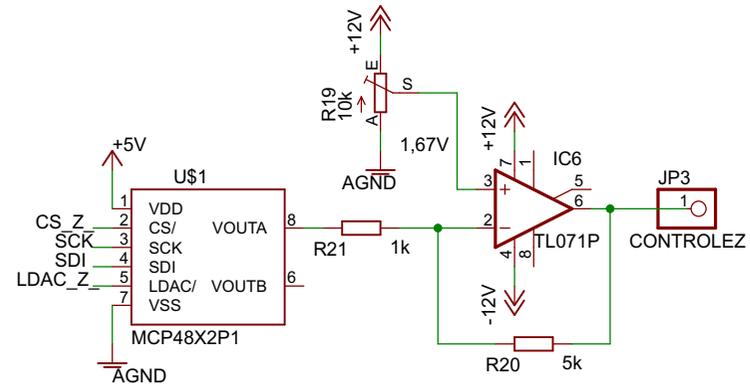
HARDWARE CONTROLE DIGITAL

Aquisição do sinal SCT1



(a)

Entrega do sinal processado



(b)

Anexo D

Firmware para o uso do controle analógico.

```

1 void config_DA_mcp4822();
2 unsigned short escreve_SPI(unsigned short dado);
3 void escreve_mcp4822(unsigned char eixo_xy, unsigned int valor);
4
5 //=====
6 //          CONFIGURA PINOS DO DAC (MCP4822)
7 //=====
8 void config_DA_mcp4822(){
9     TRIS_CS_ = 0;
10    TRIS_LDAC_ = 0;
11    CS_ = 1;
12    LDAC_ = 1;
13    SPI1_Init();
14 }
15
16 //=====
17 //          ENVIA DADOS SPI
18 //=====
19 unsigned short escreve_SPI(unsigned short dado){
20     SSPBUF = dado;
21     do {;}while(!BF_bit);
22     return SSPBUF;
23 }
24 //=====
25 //          ESCREVE NO DAC (MCP4822)
26 // tempo da funcao: 19 us
27 //=====
28 void escreve_mcp4822(unsigned char eixo_xy, unsigned int valor){
29     char temp1 = 0;
30
31     CS_ = 0; //habilita para escrita
32
33     //----- BITS DE CONTROLE -----
34     /*
35     bit 7 - VoutA ou VoutB
36     bit 6 -
37     bit 5 - Vout = 2*2.048*D/4096
38     bit 4 - modo ativo
39     bit 3,2,1,0 - bits de dados
40     */
41     if(!eixo_xy)
42         temp1 = 0b11010000;
43     else
44         temp1 = 0b01010000;
45
46     //----- BITS DE DADOS -----
47     temp1 = temp1 + ((valor>>8) & 0x000F);
48     escreve_SPI(temp1); // envia byte mais significativo
49     //SPI1_Write(temp1);
50
51     temp1 = valor;
52     escreve_SPI(temp1); // envia byte menos significativo
53     //SPI1_Write(temp1);

```

```
54
55     CS_ = 1; //desabilita para escrita
56     LDAC_ = 0; //libera valor na saida (sincronização)
57     LDAC_ = 1;
58 }
```

```
1 //=====
2 // DEFINIÇÃO DOS PINOS DE I/O
3 //=====
4 sbit USB_Dpos    at    RC5_bit;
5 sbit USB_Dneg   at    RC4_bit;
6
7 sbit CS_        at    RD2_bit;
8 sbit TRIS_CS_   at    TRISD2_bit;
9 sbit SCK        at    RB1_bit;
10 sbit TRIS_SCK   at    TRISB1_bit;
11 sbit SDO        at    RC7_bit;
12 sbit TRIS_SDO   at    TRISC7_bit;
13 sbit LDAC_      at    RD1_bit;
14 sbit TRIS_LDAC_ at    TRISD1_bit;
15
16 sbit TRIS_botao at    TRISC2_bit;
17 sbit botao_teste at    RC2_bit;
18 sbit TRIS_led   at    TRISD5_bit;
19 sbit led_teste  at    RD5_bit;
20
21 //=====
22 // DEFINIÇÃO DOS BITS
23 //=====
24
25 #define eixo_x    0
26 #define eixo_y    1
27
28 //=====
29 // DECLARACAO DAS VARIAVEIS
30 //=====
31 unsigned char readbuff[64] absolute 0x500; // Buffers should be in USB RAM, please
consult datasheet
32 unsigned char writebuff[64] absolute 0x540;
33
34 unsigned int pos_x, pos_y, pos_z, num_amostra, pos_y_anterior;
35 unsigned int min_x, max_x, min_y, max_y, passo, temp_leitura, qntd_amostra;
36 unsigned char en_varredura;
37
38 //=====
39 // DECLARAÇÃO DAS FUNCOES
40 //=====
41 void init_PIC();
42 void config_AD();
43 void init_variaveis();
44 void habilita_STM();
45 unsigned char habilita_varredura_usb();
46 unsigned char habilita_varredura_botao();
47
48 void teste_usb(unsigned char Enable);
49 void varredura_amostra(unsigned char Enable);
50
51 #include "conversor_DA.h"
52
```

```
53 //=====
54 // INTERRUPT
55 //=====
56 void interrupt () {
57     USB_Interrupt_Proc();           // USB servicing is done inside the interrupt
58 }
59
60 //=====
61 // FUNCAO PRINCIPAL
62 //=====
63 void main(void) {
64
65     init_PIC();
66     HID_Enable(&readbuff, &writebuff);    // Enable HID communication
67
68     habilita_STM();
69     //configuração dos periféricos
70     config_DA_mcp4822();
71     config_AD ();
72     ADC_Init();
73     init_variaveis();
74
75
76     while(1) {
77         en_varredura = habilita_varredura_usb();
78         if(en_varredura==0) {en_varredura = habilita_varredura_botao();}
79         //teste_usb(en_varredura);
80         varredura_amostra(en_varredura);
81     }
82 }
83
84 //=====
85 // OUTRAS FUNCOES
86 //=====
87 void init_PIC() {
88     CMCON |= 7;           // Disable comparators
89     TRIS_botao = 1;
90     TRIS_led = 0;
91     PORTA = 0; PORTB = 0; PORTC = 0; PORTD = 0;
92 }
93 //=====
94 void config_AD() {
95     TRISA = 0x0F; //AN0 a AN3 como entrada
96     ADCON1 = 0b00001011;
97 }
98 //=====
99 void init_variaveis() {
100     min_x = 1;
101     max_x = 4000;
102     min_y = 1;
103     max_y = 4000;
104     passo = 16;
105     temp_leitura = 5;
```

```

106     qntd_amostra = 10;
107     pos_x = min_x;
108     pos_y = min_y;
109     escreve_mcp4822(eixo_x, pos_x);
110     escreve_mcp4822(eixo_y, pos_y);
111 }
112
113 //=====
114 void habilita_STM(){
115     unsigned char cont;
116     //comunicação USB começa sempre recebendo informação do software
117     for(cont=0;cont<64;cont++){readbuff[cont] = 0;}
118     while(readbuff[0]!=73){
119         while(!HID_Read()){;}
120         if(readbuff[0]!=73){
121             led_teste = 1; delay_ms(500); led_teste = 0; delay_ms(500);
122             led_teste = 1; delay_ms(500); led_teste = 0; delay_ms(500);
123             led_teste = 1; delay_ms(500); led_teste = 0; delay_ms(500);
124             for(cont=0;cont<64;cont++){readbuff[cont] = 0;}
125         }
126     }
127     for(cont=0;cont<64;cont++){
128         writebuff[cont]=readbuff[cont];
129         readbuff[cont] = 0;
130     }
131     //envia informacao ao software
132     while(!HID_Write(&writebuff,64));
133     writebuff[0]=73; writebuff[1]='B'; writebuff[2]='I';
134     writebuff[3]='O'; writebuff[4]='T'; writebuff[5]='A'; writebuff[6]=73;
135     for(cont=7;cont<64;cont++){writebuff[cont]=0;}
136     while(!HID_Write(&writebuff,64));
137     led_teste = 1;
138     delay_ms(2000);
139     led_teste = 0;
140 }
141 //=====
142
143 unsigned char habilita_varredura_usb(){
144     unsigned char cont, Enable;
145
146     Enable = 0;
147     if(!HID_Read()){
148         if(readbuff[0]==73){
149             if(readbuff[1]==255){
150                 //variavel min_x
151                 min_x = readbuff[2]*256+readbuff[3];
152                 if(min_x>4000){min_x = 4000;}
153                 //variavel max_x
154                 max_x = readbuff[4]*256+readbuff[5];
155                 if(max_x<=min_x){max_x = min_x+1;}
156                 //variavel min_y
157                 min_y = readbuff[6]*256+readbuff[7];
158                 if(min_y>4000){min_y = 4000;}

```

```

159         //variavel max_y
160         max_y = readbuff[8]*256+readbuff[9];
161         if(max_y<=min_y){max_y = min_y+1;}
162         //variavel passo
163         passo = readbuff[10]*256+readbuff[11];
164         if(passo==0){passo = 1;}
165         //variavel temp_leitura
166         temp_leitura = readbuff[12]*256+readbuff[13];
167         //variavel qntd_amostra
168         qntd_amostra = readbuff[14]*256+readbuff[15];
169         if(qntd_amostra==0){qntd_amostra = 1;}
170     }
171     Enable = 1;
172 }
173 //zera o buffer da recepcao usb
174 for(cont=0;cont<64;cont++){readbuff[cont] = 0;}
175 }
176 return Enable;
177 }
178
179 //=====
180 unsigned char habilita_varredura_botao(){
181     unsigned char Enable;
182
183     Enable = 0;
184     if(!botao_teste){
185         delay_ms(1);
186         if(!botao_teste){Enable = 1;}
187     }
188     return Enable;
189 }
190 //=====
191 void teste_usb(unsigned char Enable){
192     unsigned long numero_pacote;
193     unsigned int dado_usb;
194     unsigned int x, y, z;
195     unsigned char cont;
196
197     if(Enable!=0){
198         led_teste = 1;
199         numero_pacote = 1;
200         dado_usb = 0;
201         //envio do pacote de inicio da varredura
202         for(cont=0;cont<64;cont++){writebuff[cont] = 128;}
203         while(!HID_Write(&writebuff,64));
204         Vdelay_ms(temp_leitura);
205         //inicia a varredura
206         for(y=1;y<=100;y++){
207             for(x=1;x<=100;x++){
208                 //----- posicao x -----
209                 writebuff[dado_usb] = (x>>8) & 0xFF;
210                 dado_usb++;
211                 writebuff[dado_usb] = x & 0xFF;

```

```

212     dado_usb++;
213     //----- posicao y -----
214     writebuff[dado_usb] = (y>>8) & 0xFF;
215     dado_usb++;
216     writebuff[dado_usb] = y & 0xFF;
217     dado_usb++;
218     //----- posicao z -----
219     pos_z = y;
220     writebuff[dado_usb] = (pos_z>>8) & 0xFF;
221     dado_usb++;
222     writebuff[dado_usb] = pos_z & 0xFF;
223     dado_usb++;
224     if(dado_usb>=60){
225         writebuff[60] = (numero_pacote>>14) & 0xFF;
226         writebuff[61] = (numero_pacote>>6) & 0xFF;
227         writebuff[62] = ((numero_pacote<<2) & 0b11111100);
228         if(y%2==0){writebuff[62] = writebuff[62]+2;}
229         else if(y%2==1){writebuff[62] = writebuff[62]+1;}
230         writebuff[63] = 0;
231         while(!HID_Write(&writebuff,64));
232         Vdelay_ms(temp_leitura);
233         numero_pacote++;
234         dado_usb = 0;
235     }
236 }
237 }
238 //envio do pacote de fim de varredura
239 for(cont=0;cont<64;cont++){writebuff[cont] = 150;}
240 while(!HID_Write(&writebuff,64));
241 Vdelay_ms(temp_leitura);
242 led_teste = 0;
243 }
244 }
245
246 //=====
247 void varredura_amostra(unsigned char Enable){
248     unsigned long numero_pacote, media_z;
249     unsigned char cont, amostra_z;
250     unsigned int dado_usb;
251     unsigned char inverte_coluna;
252
253     if(Enable){
254         //seta a posicao inicial
255         while(pos_x!=min_x || pos_y!=min_y){
256             //seta posicao x
257             if(pos_x>min_x){pos_x = pos_x - passo;}
258             else if(pos_x<min_x){pos_x++;}
259             //seta posicao y
260             if(pos_y>min_y){pos_y = pos_y - passo;}
261             else if(pos_y<min_y){pos_y++;}
262             escreve_mcp4822(eixo_x, pos_x);
263             escreve_mcp4822(eixo_y, pos_y);
264             Vdelay_ms(temp_leitura);

```

```

265     }
266     led_teste = 1;
267     numero_pacote = 1;
268     dado_usb = 0;
269     inverte_coluna = 1;
270     //envio do pacote de inicio da varredura
271     for(cont=0;cont<64;cont++){writebuff[cont] = 128;}
272     while(!HID_Write(&writebuff,64));
273     Vdelay_ms(temp_leitura);
274     //inicio da varredura
275     while(pos_y>=min_y && pos_y<max_y){
276         escreve_mcp4822(eixo_y, pos_y);
277         while(pos_x>=min_x && pos_x<max_x){
278             escreve_mcp4822(eixo_x, pos_x);
279             Vdelay_ms(temp_leitura);
280             //----- posicao x -----
281             writebuff[dado_usb] = (pos_x>>8) & 0xFF;
282             dado_usb++;
283             writebuff[dado_usb] = pos_x & 0xFF;
284             dado_usb++;
285             //----- posicao y -----
286             writebuff[dado_usb] = (pos_y>>8) & 0xFF;
287             dado_usb++;
288             writebuff[dado_usb] = pos_y & 0xFF;
289             dado_usb++;
290             //----- posicao z -----
291             //faz a media das amostras
292             media_z = 0;
293             for(amostra_z=1;amostra_z<=qntd_amostra;amostra_z++){
294                 pos_z = ADC_Read(1);
295                 //pos_z = pos_y;
296                 media_z = media_z+pos_z;
297             }
298             media_z = media_z/qntd_amostra;
299             pos_z = media_z;
300             writebuff[dado_usb] = (pos_z>>8) & 0xFF;
301             dado_usb++;
302             writebuff[dado_usb] = pos_z & 0xFF;
303             dado_usb++;
304             //envia dado pela usb se fez a leitura de 10 pontos
305             if(dado_usb>=60){
306                 writebuff[60] = (numero_pacote>>14) & 0xFF;
307                 writebuff[61] = (numero_pacote>>6) & 0xFF;
308                 writebuff[62] = ((numero_pacote<<2) & 0b11111100);
309                 writebuff[62] = writebuff[62] + inverte_coluna;
310                 writebuff[63] = 0;
311                 while(!HID_Write(&writebuff,64));
312                 numero_pacote++;
313                 dado_usb = 0;
314             }
315             switch (inverte_coluna){
316                 case 1:
317                     pos_x = pos_x+passo;

```

```
318         break;
319     case 2:
320         pos_x = pos_x-passo;
321         break;
322     }
323 }
324 //faz a leitura da coluna invertida
325 switch (inverte_coluna){
326     case 1:
327         pos_x = pos_x-passo;
328         inverte_coluna = 2;
329         break;
330     case 2:
331         pos_x = pos_x+passo;
332         inverte_coluna = 1;
333         break;
334     }
335     pos_y = pos_y+passo;
336 }
337 //envio do pacote de fim de varredura
338 Vdelay_ms(temp_leitura);
339 for(cont=0;cont<64;cont++){writebuff[cont] = 150;}
340 while(!HID_Write(&writebuff,64));
341 Vdelay_ms(temp_leitura);
342 led_teste = 0;
343 }
344
345 }
```

Anexo E

Firmware para o uso do controle digital.

```

1 //=====
2 // DEFINIÇÃO DOS PINOS DE I/O
3 //=====
4 sbit USB_Dpos      at      RC5_bit;
5 sbit USB_Dneg      at      RC4_bit;
6
7 sbit CS_XY_        at      RD2_bit;
8 sbit TRIS_CS_XY_   at      TRISD2_bit;
9 sbit CS_Z_         at      RD3_bit;
10 sbit TRIS_CS_Z_    at      TRISD3_bit;
11 sbit SCK           at      RB1_bit;
12 sbit TRIS_SCK      at      TRISB1_bit;
13 sbit SDO           at      RC7_bit;
14 sbit TRIS_SDO      at      TRISC7_bit;
15 sbit LDAC_XY_      at      RD1_bit;
16 sbit TRIS_LDAC_XY_ at      TRISD1_bit;
17 sbit LDAC_Z_       at      RD0_bit;
18 sbit TRIS_LDAC_Z_ at      TRISD0_bit;
19
20 sbit TRIS_botao    at      TRISC2_bit;
21 sbit botao_teste  at      RC2_bit;
22 sbit TRIS_led      at      TRISD5_bit;
23 sbit led_teste     at      RD5_bit;
24
25 //=====
26 // DEFINIÇÃO DOS BITS
27 //=====
28 #define flag_timer2      PIR1.TMR2IF
29 #define en_timer2        T2CON.TMR2ON
30 #define flag_AD          PIR1.ADIF
31 #define en_AD            ADCON0.ADON
32 #define start_AD        ADCON0.GO_DONE
33
34 #define SetPoint          512
35
36 #define varredura_desabilitada  0
37 #define posicionamento_sonda    1
38 #define inicio_varredura_amostra  2
39 #define meio_varredura_amostra    3
40 #define fim_varredura_amostra     4
41
42 //=====
43 // DECLARACAO DAS VARIAVEIS
44 //=====
45 unsigned char readbuff[64] absolute 0x500; // Buffers should be in USB RAM, please
consult datasheet
46 unsigned char writebuff[64] absolute 0x540;
47
48 //parametros recebidos pelo software
49 unsigned int parametro_min_x, parametro_max_x;
50 unsigned int parametro_min_y, parametro_max_y;
51 unsigned int parametro_passo, parametro_tempo_pos_xy, parametro_qntde_amostra;
52

```

```
53 unsigned int pos_x, pos_x_anterior;
54 unsigned int pos_y, pos_y_anterior;
55
56 unsigned long media_z;
57 unsigned char en_controle_z;
58 signed int SCT, pos_z, SCT_anterior, erro;
59
60 unsigned char dado_usb, inverte_coluna, cont, en_varredura, cont_aux;
61 unsigned int num_amostra, cont_interrupcao;
62 unsigned long numero_pacote;
63
64 //=====
65 // DECLARAÇÃO DAS FUNCOES
66 //=====
67 void init_PIC();
68 void config_timer2();
69 void config_DA_mcp4822();
70 void config_AD();
71 void init_variaveis();
72 void habilita_STM();
73
74 //=====
75 // INTERRUPCAO
76 //=====
77 void interrupt(){
78
79     if(flag_timer2){
80         flag_timer2 = 0;
81         TMR2 = 0;
82         en_controle_z = 1;
83         cont_interrupcao++;
84     }
85     else{
86         USB_Interrupt_Proc(); // USB servicing is done inside the interrupt
87     }
88 }
89
90 //=====
91 // FUNCAO PRINCIPAL
92 //=====
93 void main(void){
94
95     init_PIC();
96
97     HID_Enable(&readbuff,&writebuff); // Enable HID communication
98     //configuração dos periféricos
99     config_DA_mcp4822();
100    config_timer2();
101    config_AD();
102    ADC_Init();
103    init_variaveis();
104
105    habilita_STM();
```

```

106
107     en_timer2 = 1;
108     en_AD = 1;
109
110
111     while(1){
112         //=====
113         // CONTROLE DA POSIÇÃO Z
114         //=====
115         if(en_controle_z){
116             //////////////////////////////////////////////////// INICIO DA CONVERSAO AD ////////////////////////////////////
117             start_AD = 1;
118             while(!flag_AD || start_AD){;}
119             flag_AD = 0;
120             //////////////////////////////////////////////////// MÉTODOS DE DISCRETIZAÇÃO BILINEAR ////////////////////////////////////
121             // controlez[k] = controlez[k-1] - (4*2*T)/(R*C)*(2*SetPoint-SCT[k-1]-SCT[k])
122             SCT = ((ADRESH<<8)&0xFF00)+ADRESL;
123             erro = (2*SetPoint-SCT-SCT_anterior)/64;//divisor deve ser multiplo de 2
124             para ser mais rapido
125             pos_z = pos_z + erro;
126             SCT_anterior = SCT;
127             ////////////////////////////////////////////////////
128             if(pos_z>4095){pos_z = 4095;}
129             else if(pos_z<0){pos_z = 0;}
130             //////////////////////////////////////////////////// FIM DA CONVERSAO AD ////////////////////////////////////
131             //////////////////////////////////////////////////// INICIO COMUNICACAO COM MCP PARA O CONTROLE Z ////////////////////////////////////////////////////
132             CS_Z_ = 0;
133             SSPBUF = 0b01010000+((pos_z>>8)&0x000F);
134             while(!BF_bit);
135             SSPBUF = pos_z & 0x00FF;
136             while(!BF_bit);
137             CS_Z_ = 1;
138             LDAC_Z_ = 0; //libera valor na saida (sincronização)
139             LDAC_Z_ = 1;
140             //////////////////////////////////////////////////// FIM COMUNICACAO COM MCP PARA O CONTROLE Z ////////////////////////////////////
141             en_controle_z = 0;
142         }
143         //=====
144         // CONTROLE DA POSIÇÃO X
145         //=====
146         if(pos_x != pos_x_anterior){
147             CS_XY_ = 0;
148             SSPBUF = 0b11010000+((pos_x>>8)&0x000F);
149             while(!BF_bit);
150             SSPBUF = pos_x & 0x00FF;
151             while(!BF_bit);
152             CS_XY_ = 1;
153             LDAC_XY_ = 0; //libera valor na saida (sincronização)
154             LDAC_XY_ = 1;
155             pos_x_anterior = pos_x;
156         }
157         //=====
158         // CONTROLE DA POSIÇÃO Y

```

```

158 //=====
159 if(pos_y != pos_y_anterior){
160     CS_XY_ = 0;
161     SSPBUF = 0b01010000+((pos_y>>8)&0x000F);
162     while(!BF_bit);
163     SSPBUF = pos_y & 0x00FF;
164     while(!BF_bit);
165     CS_XY_ = 1;
166     LDAC_XY_ = 0; //libera valor na saida (sincronização)
167     LDAC_XY_ = 1;
168     pos_y_anterior = pos_y;
169 }
170 switch(en_varredura){
171     //=====
172     // INICIO DA FUNÇÃO DE RECEPÇÃO DAS INFORMAÇÕES DO SOFTWARE
173     //=====
174     case varredura_desabilitada:
175         if(!HID_Read()){
176             if(readbuff[0]==73){
177                 if(readbuff[1]==255){
178                     //variavel parametro_min_x
179                     parametro_min_x = readbuff[2]*256+readbuff[3];
180                     if(parametro_min_x>4046){parametro_min_x = 4046;}
181                     //////////////////////////////////////////////////
182                     //variavel parametro_max_x
183                     parametro_max_x = readbuff[4]*256+readbuff[5];
184                     if(parametro_max_x>4046){parametro_max_x = 4046;}
185                     if(parametro_max_x<=parametro_min_x){
186                         parametro_max_x = parametro_max_x+1;
187                     }
188                     //////////////////////////////////////////////////
189                     //variavel min_y
190                     parametro_min_y = readbuff[6]*256+readbuff[7];
191                     if(parametro_min_y>4046){parametro_min_y = 4046;}
192                     //////////////////////////////////////////////////
193                     //variavel max_y
194                     parametro_max_y = readbuff[8]*256+readbuff[9];
195                     if(parametro_max_y>4046){parametro_max_y = 4046;}
196                     if(parametro_max_y<=parametro_min_y){
197                         parametro_max_y = parametro_max_y+1;
198                     }
199                     //variavel passo
200                     parametro_passo = readbuff[10]*256+readbuff[11];
201                     if(parametro_passo==0){parametro_passo = 1;}
202                     //variavel temp_leitura
203                     parametro_tempo_pos_xy = readbuff[12]*256+readbuff[13];
204                     //variavel qntd_amostra
205                     parametro_qntde_amostra = readbuff[14]*256+readbuff[15];
206                     if(parametro_qntde_amostra==0){parametro_qntde_amostra = 1;}
207
208                     en_varredura = posicionamento_sonda;
209                     cont_interrupcao = 0;
210                 }

```

```

211         //zera o buffer da recepcao usb
212         for(cont_aux=0;cont_aux<64;cont_aux++){readbuff[cont_aux] = 0;}
213     }
214 }
215 break;
216 //=====
217 // POSCIONAMENTO DA SONDA
218 //=====
219 case posicionamento_sonda:
220     if(pos_x!=parametro_min_x || pos_y!=parametro_min_y){
221         if(cont_interrupcao>=5){//tempo de 5*tempo timer 2
222             //////////////// seta posicao x //////////////////
223             if(pos_x>parametro_min_x){pos_x = pos_x--;}
224             else if(pos_x<parametro_min_x){pos_x++;}
225             //////////////// seta posicao y //////////////////
226             if(pos_y>parametro_min_y){pos_y = pos_y--;}
227             else if(pos_y<parametro_min_y){pos_y++;}
228             cont_interrupcao= 0;
229         }
230     }
231     else{
232         en_varredura = inicio_varredura_amostra;
233         cont_interrupcao = 0;
234     }
235     break;
236 //=====
237 // INICIO DA VARREDURA DA AMOSTRA
238 //=====
239 case inicio_varredura_amostra:
240     ////////////////envio do pacote de inicio da varredura//////////
241     if(cont_interrupcao>=40){//tempo de 40* tempo timer 2
242         //led_teste = 1;
243         numero_pacote = 1;
244         dado_usb = 0;
245         inverte_coluna = 1;
246         for(cont=0;cont<64;cont++){writebuff[cont] = 128;}
247         while(!HID_Write(&writebuff,64));
248         en_varredura = meio_varredura_amostra;
249         cont_interrupcao = 0;
250     }
251     break;
252 //=====
253 // MEIO DA VARREDURA DA AMOSTRA
254 //=====
255 case meio_varredura_amostra:
256     if(pos_y<=parametro_max_y){
257         if(cont_interrupcao>=parametro_tempo_pos_xy){//tempo de
258             parametro_tempo_pos_xy * tempo timer 2
259             //----- posicao x -----
260             writebuff[dado_usb] = (pos_x>>8) & 0xFF;
261             dado_usb++;
262             writebuff[dado_usb] = pos_x & 0xFF;
263             dado_usb++;

```

```

263 //----- posicao y -----
264 writebuff[dado_usb] = (pos_y>>8) & 0xFF;
265 dado_usb++;
266 writebuff[dado_usb] = pos_y & 0xFF;
267 dado_usb++;
268 //----- posicao z -----
269 en_controle_z = 0;
270 media_z = 0;
271 for(cont=1; cont<=parametro_qntde_amostra; cont++){
272     while(en_controle_z==0){;}
273     ////////////////////////////////////////////////// INICIO DA CONVERSAO AD
274     //////////////////////////////////////////////////
275     start_AD = 1;
276     while(!flag_AD || start_AD){;}
277     flag_AD = 0;
278     ////////////////////////////////////////////////// MÉTODOS DE DISCRETIZAÇÃO BILINEAR
279     //////////////////////////////////////////////////
280     // controlez[k] = controlez[k-1] -
281     // (4*2*T)/(R*C)*(2*SetPoint-SCT[k-1]-SCT[k])
282     SCT = ((ADRESH<<8)&0xFF00)+ADRESL;
283     erro = (2*SetPoint-SCT-SCT_anterior)/64; //divisor deve ser
284     // multiplo de 2 para ser mais rapido
285     pos_z = pos_z + erro;
286     SCT_anterior = SCT;
287
288     //////////////////////////////////////////////////
289     //////////////////////////////////////////////////
290     if(pos_z>4095){pos_z = 4095;}
291     else if(pos_z<0){pos_z = 0;}
292     ////////////////////////////////////////////////// FIM DA CONVERSAO AD
293     //////////////////////////////////////////////////
294     ////////////////////////////////////////////////// INICIO COMUNICACAO COM MCP PARA O
295     ////////////////////////////////////////////////// CONTROLE Z //////////////////////////////////////////////////
296     CS_Z_ = 0;
297     SSPBUF = 0b01010000+((pos_z>>8)&0x000F);
298     while(!BF_bit);
299     SSPBUF = pos_z & 0x00FF;
300     while(!BF_bit);
301     CS_Z_ = 1;
302     LDAC_Z_ = 0; //libera valor na saida (sincronização)
303     LDAC_Z_ = 1;
304     ////////////////////////////////////////////////// FIM COMUNICACAO COM MCP PARA O CONTROLE Z
305     //////////////////////////////////////////////////
306     media_z = media_z+pos_z;
307     en_controle_z = 0;
308 }
309 media_z = media_z/parametro_qntde_amostra;
310 writebuff[dado_usb] = ((1024-pos_z/4)>>8) & 0xFF;
311 dado_usb++;
312 writebuff[dado_usb] = (1024-pos_z/4) & 0xFF;
313 dado_usb++;
314 //envia dado pela usb se fez a leitura de 10 pontos
315 if(dado_usb>=60){

```

```

307         writebuff[60] = (numero_pacote>>14) & 0xFF;
308         writebuff[61] = (numero_pacote>>6) & 0xFF;
309         writebuff[62] = ((numero_pacote<<2) & 0b11111100);
310         writebuff[62] = writebuff[62] + inverte_coluna;
311         writebuff[63] = 0;
312         while(!HID_Write(&writebuff,64));
313         numero_pacote++;
314         dado_usb = 0;
315         if(numero_pacote>=4004){led_teste = 1;}
316         else{led_teste = 0;}
317     }
318     if(inverte_coluna==1){pos_x = pos_x+parametro_passo;}
319     else{pos_x = pos_x-parametro_passo;}
320     if(pos_x<parametro_min_x || pos_x>parametro_max_x){
321         //faz a leitura da coluna invertida
322         if(inverte_coluna==1){
323             pos_x = pos_x-parametro_passo;
324             inverte_coluna = 2;
325         }
326         else{
327             pos_x = pos_x+parametro_passo;
328             inverte_coluna = 1;
329         }
330         pos_y = pos_y+parametro_passo;
331     }
332     cont_interrupcao = 0;
333 }
334 }
335 else{
336     if(cont_interrupcao>=40){ //tempo de 40 * tempo timer 2
337         en_varredura = fim_varredura_amostra;
338         cont_interrupcao = 0;
339     }
340 }
341 break;
342 //=====
343 // FIM DA VARREDURA DA AMOSTRA
344 //=====
345 case fim_varredura_amostra:
346     for(cont=0;cont<64;cont++){writebuff[cont] = 150;}
347     while(!HID_Write(&writebuff,64));
348     en_varredura = varredura_desabilitada;
349     for(cont_aux=0;cont_aux<64;cont_aux++){readbuff[cont_aux] = 0;}
350     //led_teste = 0;
351     break;
352 }
353 }
354 }
355 //=====
356 //=====
357 // OUTRAS FUNCOES
358 //=====
359 //=====

```

```
360 void init_PIC(){
361     CMCON   |= 7;                               // Disable comparators
362     TRIS_botao = 1;
363     TRIS_led = 0;
364     PORTA = 0; PORTB = 0; PORTC = 0; PORTD = 0;
365     INTCON = 0b11000000;
366 }
367 //=====
368 void config_AD(){
369     TRISA = 0x0F; //AN0 a AN3 como entrada
370     flag_AD = 0;
371     PIE1.ADIE = 0;//desabilita interrupcao
372     IPR1.ADIP = 0;//baixa prioridade
373     ADCON0 = 0b00000000; //selecao do canal 0 / habilita AD
374     ADCON1 = 0b00001011; //AN0 e AN1 como analogico
375     ADCON2 = 0b10001000; //justificacao para direita tempo aquisicao = 2*Tad clock
    de conversao = Fosc/2
376 }
377 //=====
378 void config_DA_mcp4822(){
379     TRIS_CS_XY_ = 0;
380     TRIS_CS_Z_ = 0;
381     TRIS_LDAC_XY_ = 0;
382     TRIS_LDAC_Z_ = 0;
383     CS_XY_ = 1;
384     LDAC_XY_ = 1;
385     LDAC_Z_ = 1;
386     SPI1_Init();
387 }
388 //=====
389 void config_timer2(){
390     PIR1 = 0;
391     PIE1.TMR2IE = 1; //habilita interrupcao do timer 2
392     IPR1.TMR2IP = 0; //baixa prioridade do timer 2
393     TMR2 = 0;
394     //T2CON = 0b00000001; //prescaler 1:4 postscaler 1:1
395     T2CON = 0b00001011; //prescaler de 1:16 potscaler 1:2
396     //PR2 = 112;           //tempo de 300us
397     PR2 = 188;           //tempo de 500us
398 }
399 //=====
400 void init_variaveis(){
401     unsigned char cont;
402     parametro_min_x = 47;
403     parametro_max_x = 4046;
404     parametro_min_y = 47;
405     parametro_max_y = 4046;
406     parametro_passo = 20;
407     parametro_tempo_pos_xy = 20;
408     parametro_qntde_amostra = 5;
409     pos_x = 0;
410     pos_y = 0;
411     pos_z = 2000;
```

```
412     en_varredura = 0;
413     erro = 0;
414     en_controle_z = 0;
415     for(cont=0;cont<64;cont++){
416         readbuff[cont] = 0;
417         writebuff[cont] = 0;
418     }
419 }
420
421 //=====
422 void habilita_STM(){
423     unsigned char cont;
424     //comunicação USB começa sempre recebendo informação do software
425     for(cont=0;cont<64;cont++){readbuff[cont] = 0;}
426     while(readbuff[0]!=73){
427         while(!HID_Read()){;}
428         if(readbuff[0]!=73){
429             led_teste = 1; delay_ms(500); led_teste = 0; delay_ms(500);
430             led_teste = 1; delay_ms(500); led_teste = 0; delay_ms(500);
431             led_teste = 1; delay_ms(500); led_teste = 0; delay_ms(500);
432             for(cont=0;cont<64;cont++){readbuff[cont] = 0;}
433         }
434     }
435     for(cont=0;cont<64;cont++){
436         writebuff[cont]=readbuff[cont];
437         readbuff[cont] = 0;
438     }
439     //envia informacao ao software
440     while(!HID_Write(&writebuff,64));
441     writebuff[0]=73; writebuff[1]='B'; writebuff[2]='I';
442     writebuff[3]='O'; writebuff[4]='T'; writebuff[5]='A'; writebuff[6]=73;
443     for(cont=7;cont<64;cont++){writebuff[cont]=0;}
444     while(!HID_Write(&writebuff,64));
445     led_teste = 1;
446     delay_ms(2000);
447     led_teste = 0;
448 }
```

Anexo F

Código do descritor USB para o firmware.

```

1  const unsigned int USB_VENDOR_ID = 0x1234;
2  const unsigned int USB_PRODUCT_ID = 0x0001;
3  const char USB_SELF_POWER = 0x80;           // Self powered 0xC0, 0x80 bus powered
4  const char USB_MAX_POWER = 50;             // Bus power required in units of 2 mA
5  const char HID_INPUT_REPORT_BYTES = 64;
6  const char HID_OUTPUT_REPORT_BYTES = 64;
7  const char USB_TRANSFER_TYPE = 0x03;       //0x03 Interrupt
8  const char EP_IN_INTERVAL = 1;
9  const char EP_OUT_INTERVAL = 1;
10
11 const char USB_INTERRUPT = 1;
12 const char USB_HID_EP = 1;
13 const char USB_HID_RPT_SIZE = 33;
14
15 /* Device Descriptor */
16 const struct {
17     char bLength;           // bLength - Descriptor size in bytes (12h)
18     char bDescriptorType;  // bDescriptorType - The constant DEVICE (01h)
19     unsigned int bcdUSB;    // bcdUSB - USB specification release number
20     // (BCD)
21     char bDeviceClass;     // bDeviceClass - Class Code
22     char bDeviceSubClass;  // bDeviceSubClass - Subclass code
23     char bDeviceProtocol;  // bDeviceProtocol - Protocol code
24     char bMaxPacketSize0;  // bMaxPacketSize0 - Maximum packet size for endpoint 0
25     unsigned int idVendor; // idVendor - Vendor ID
26     unsigned int idProduct; // idProduct - Product ID
27     unsigned int bcdDevice; // bcdDevice - Device release number (BCD)
28     char iManufacturer;    // iManufacturer - Index of string descriptor for the
29     // manufacturer
30     char iProduct;         // iProduct - Index of string descriptor for the
31     // product.
32     char iSerialNumber;    // iSerialNumber - Index of string descriptor for the
33     // serial number.
34     char bNumConfigurations; // bNumConfigurations - Number of possible configurations
35 } device_dsc = {
36     0x12,           // bLength
37     0x01,           // bDescriptorType
38     0x0200,        // bcdUSB
39     0x00,           // bDeviceClass
40     0x00,           // bDeviceSubClass
41     0x00,           // bDeviceProtocol
42     8,             // bMaxPacketSize0
43     USB_VENDOR_ID, // idVendor
44     USB_PRODUCT_ID, // idProduct
45     0x0001,        // bcdDevice
46     0x01,          // iManufacturer
47     0x02,          // iProduct
48     0x00,          // iSerialNumber
49     0x01           // bNumConfigurations
50 };
51
52 /* Configuration 1 Descriptor */
53 const char configDescriptor1[] = {

```

```

50 // Configuration Descriptor
51 0x09, // bLength - Descriptor size in bytes
52 0x02, // bDescriptorType - The constant CONFIGURATION (02h)
53 0x29,0x00, // wTotalLength - The number of bytes in the
configuration descriptor and all of its subordinate descriptors
54 1, // bNumInterfaces - Number of interfaces in the
configuration
55 1, // bConfigurationValue - Identifier for Set Configuration
and Get Configuration requests
56 0, // iConfiguration - Index of string descriptor for the
configuration
57 USB_SELF_POWER, // bmAttributes - Self/bus power and remote wakeup
settings
58 USB_MAX_POWER, // bMaxPower - Bus power required in units of 2 mA
59
60 // Interface Descriptor
61 0x09, // bLength - Descriptor size in bytes (09h)
62 0x04, // bDescriptorType - The constant Interface (04h)
63 0, // bInterfaceNumber - Number identifying this interface
64 0, // bAlternateSetting - A number that identifies a
descriptor with alternate settings for this bInterfaceNumber.
65 2, // bNumEndpoint - Number of endpoints supported not
counting endpoint zero
66 0x03, // bInterfaceClass - Class code
67 0, // bInterfaceSubclass - Subclass code
68 0, // bInterfaceProtocol - Protocol code
69 0, // iInterface - Interface string index
70
71 // HID Class-Specific Descriptor
72 0x09, // bLength - Descriptor size in bytes.
73 0x21, // bDescriptorType - This descriptor's type: 21h to
indicate the HID class.
74 0x01,0x01, // bcdHID - HID specification release number (BCD).
75 0x00, // bCountryCode - Numeric expression identifying the
country for localized hardware (BCD) or 00h.
76 1, // bNumDescriptors - Number of subordinate report and
physical descriptors.
77 0x22, // bDescriptorType - The type of a class-specific
descriptor that follows
78 USB_HID_RPT_SIZE,0x00, // wDescriptorLength - Total length of the descriptor
identified above.
79
80 // Endpoint Descriptor
81 0x07, // bLength - Descriptor size in bytes (07h)
82 0x05, // bDescriptorType - The constant Endpoint (05h)
83 USB_HID_EP | 0x80, // bEndpointAddress - Endpoint number and direction
84 USB_TRANSFER_TYPE, // bmAttributes - Transfer type and supplementary
information
85 0x40,0x00, // wMaxPacketSize - Maximum packet size supported
86 EP_IN_INTERVAL, // bInterval - Service interval or NAK rate
87
88 // Endpoint Descriptor
89 0x07, // bLength - Descriptor size in bytes (07h)

```

```

90     0x05, // bDescriptorType - The constant Endpoint (05h)
91     USB_HID_EP, // bEndpointAddress - Endpoint number and direction
92     USB_TRANSFER_TYPE, // bmAttributes - Transfer type and supplementary information
93     0x40, 0x00, // wMaxPacketSize - Maximum packet size supported
94     EP_OUT_INTERVAL // bInterval - Service interval or NAK rate
95 };
96
97 const struct {
98     char report[USB_HID_RPT_SIZE];
99 }hid_rpt_desc =
100 {
101     {0x06, 0x00, 0xFF, // Usage Page = 0xFF00 (Vendor Defined Page 1)
102     0x09, 0x01, // Usage (Vendor Usage 1)
103     0xA1, 0x01, // Collection (Application)
104     // Input report
105     0x19, 0x01, // Usage Minimum
106     0x29, 0x40, // Usage Maximum
107     0x15, 0x00, // Logical Minimum (data bytes in the report may have
108     minimum value = 0x00)
109     0x26, 0xFF, 0x00, // Logical Maximum (data bytes in the report may have
110     maximum value = 0x00FF = unsigned 255)
111     0x75, 0x08, // Report Size: 8-bit field size
112     0x95, HID_INPUT_REPORT_BYTES, // Report Count
113     0x81, 0x02, // Input (Data, Array, Abs)
114     // Output report
115     0x19, 0x01, // Usage Minimum
116     0x29, 0x40, // Usage Maximum
117     0x75, 0x08, // Report Size: 8-bit field size
118     0x95, HID_OUTPUT_REPORT_BYTES, // Report Count
119     0x91, 0x02, // Output (Data, Array, Abs)
120     0xC0} // End Collection
121 };
122
123 //Language code string descriptor
124 const struct {
125     char bLength;
126     char bDscType;
127     unsigned int string[1];
128 } strd1 = {
129     4,
130     0x03,
131     {0x0409}
132 };
133
134 //Manufacturer string descriptor
135 const struct {
136     char bLength;
137     char bDscType;
138     unsigned int string[5];
139 }strd2={
140     12, //sizeof this descriptor string
141     0x03,

```

```
141     {'B','I','O','T','A'}
142     };
143
144     //Product string descriptor
145     const struct{
146         char bLength;
147         char bDscType;
148         unsigned int string[32];
149     }strd3={
150         66,           //sizeof this descriptor string
151         0x03,
152         {'m','i','c','r','o','s','c','o','p','i','o',' ','d','e',' ','t','u','n','e','l','a',
153         , 'm','e','n','t','o',' ','-',' ','S','T','M'}
154     };
155
156     //Array of configuration descriptors
157     const char* USB_config_dsc_ptr[1];
158
159     //Array of string descriptors
160     const char* USB_string_dsc_ptr[3];
161
162     void USB_Init_Desc(){
163         USB_config_dsc_ptr[0] = &configDescriptor1;
164         USB_string_dsc_ptr[0] = (const char*)&strd1;
165         USB_string_dsc_ptr[1] = (const char*)&strd2;
166         USB_string_dsc_ptr[2] = (const char*)&strd3;
167     }
```

Anexo G

Software do gerenciador USB

```
1  using System;
2  using System.Collections.Generic;
3  using System.ComponentModel;
4  using System.Data;
5  using System.Drawing;
6  using System.Text;
7  using System.Windows.Forms;
8  using UsbLibrary;
9  using System.IO;
10
11 namespace UsbApp
12 {
13     public partial class gerenciador_usb : Form
14     {
15
16         //Duas filas do tipo FIFO para armazenamento da leitura
17         Queue<int> qe_leituraZ = new Queue<int>();
18         Queue<int> qe_leituraZ_temp = new Queue<int>();
19
20         //sentido da coluna ( sistema cartesiano - eixo X)
21         //padrão: reversa->1
22         //      direta->0
23         //      nao usada->2
24         int coluna_reversa = 2;
25
26
27         //usado para verificação de pacotes recebidos e contagem na tela
28         int conta_pacotes_validos = 0;
29
30         // Fator para comunicação 2 bytes por variável (firmware no pic)
31         const int fator_de_multiplicacao = 256;
32
33         // 02/09/2011 /*
34         int indice_arquivo = 0;
35         string dadoscoletados = string.Empty;
36         string dadosformatados = string.Empty;
37
38         string dadosX = string.Empty;
39         string dadosY = string.Empty;
40         string dadosZ = string.Empty;
41         // 02/09/2011 */
42
43         public gerenciador_usb()
44         {
45             InitializeComponent();
46         }
47
48         public void acrescenta_linha(string nova_linha)
49         {
50             dadosformatados = dadosformatados + nova_linha + "\n";
51         }
52
53         public string escreve_texto_numerico_em_dois_bytes(string valor)
```

```
54     {
55         int maior = 0, menor = 0;
56         menor = Convert.ToInt32(valor) % fator_de_multiplicacao;
57         maior = Convert.ToInt32(valor) / fator_de_multiplicacao;
58
59         return String.Format("{0:000}", maior) + " " + String.Format("{0:000}",
60             menor) + " ";
61     }
62
63     private void usb_OnDeviceArrived(object sender, EventArgs e)
64     {
65         this.lb_message.Items.Add("Found a Device");
66     }
67
68     private void usb_OnDeviceRemoved(object sender, EventArgs e)
69     {
70         if (InvokeRequired)
71         {
72             Invoke(new EventHandler(usb_OnDeviceRemoved), new object[] { sender, e
73                 });
74         }
75         else
76         {
77             this.lb_message.Items.Add("Dispositivo removido");
78         }
79     }
80
81     //Executada depois do reconhecimento do dispositivo
82     private void usb_OnSpecifiedDeviceArrived(object sender, EventArgs e)
83     {
84         this.lb_message.Items.Add("Dispositivo ok!");
85         this.lb_message.Items.Add("Mensagem padrão pronta para envio.");
86
87         //configuração da string responsável por enviar dados
88         string text = "";
89         for (int i = 0; i < this.usb.SpecifiedDevice.OutputReportLength - 1; i++)
90         {
91             //define o primeiro pacote a ser enviado
92             // 73 -> mensagem padrão para inicialização do firmware
93             if (i == 1)
94             {
95                 text += "073 ";
96             }
97             else
98             {
99                 text += "000 ";
100             }
101         }
102         this.tb_send.Text = text;
103     }
104
105     protected override void OnHandleCreated(EventArgs e)
```

```
105     {
106         base.OnHandleCreated(e);
107         usb.RegisterHandle(Handle);
108     }
109
110     protected override void WndProc(ref Message m)
111     {
112         usb.ParseMessages(ref m);
113         base.WndProc(ref m); // envia mensagem no base form
114     }
115
116     private void btn_ok_Click(object sender, EventArgs e)
117     {
118         try
119         {
120             // 12/09/2011
121             // alterações na tela
122             lb_message.Items.Clear();
123             lb_read.Items.Clear();
124             btn_ok.Text = "Reconectar";
125
126             this.usb.ProductId = Int32.Parse(this.tb_product.Text, System.
127             Globalization.NumberStyles.HexNumber);
128             this.usb.VendorId = Int32.Parse(this.tb_vendor.Text, System.
129             Globalization.NumberStyles.HexNumber);
130             this.usb.CheckDevicePresent();
131         }
132         catch (Exception ex)
133         {
134             MessageBox.Show(ex.ToString());
135         }
136     }
137
138     private void btn_send_Click(object sender, EventArgs e)
139     {
140         try
141         {
142             string text = this.tb_send.Text + " ";
143             text.Trim();
144             string[] arrText = text.Split(' ');
145             byte[] data = new byte[arrText.Length];
146             for (int i = 0; i < arrText.Length; i++)
147             {
148                 if (arrText[i] != "")
149                 {
150                     int value = Int32.Parse(arrText[i], System.Globalization.
151                     NumberStyles.Number);
152                     data[i] = (byte)Convert.ToByte(value);
153                 }
154             }
155
156             if (this.usb.SpecifiedDevice != null)
```

```
155         {
156             this.usb.SpecifiedDevice.SendData(data);
157         }
158         else
159         {
160             MessageBox.Show("Dispositivo não plugado!! ");
161         }
162     }
163 }
164 catch (Exception ex)
165 {
166     MessageBox.Show(ex.ToString());
167 }
168 }
169
170 private void usb_OnSpecifiedDeviceRemoved(object sender, EventArgs e)
171 {
172     if (InvokeRequired)
173     {
174         Invoke(new EventHandler(usb_OnSpecifiedDeviceRemoved), new object[] {
175             sender, e });
176     }
177     else
178     {
179         this.lb_message.Items.Add("Dispositivo removido");
180     }
181 }
182
183 private void usb_OnDataRecieved(object sender, DataRecievedEventArgs args)
184 {
185     if (InvokeRequired)
186     {
187         try
188         {
189             Invoke(new DataRecievedEventHandler(usb_OnDataRecieved), new object
190                 [] { sender, args });
191         }
192         catch (Exception ex)
193         {
194             Console.WriteLine(ex.ToString());
195         }
196     }
197     else
198     {
199         string rec_data = "Data: ";
200         //09/10/11
201         int count = 0;
202         int count_coluna = 0;
203         byte leitura_anterior = 0;
204
205         const int posicao_z_mais_significativo = 5;
```

```
206     const int posicao_z_menos_significativo = 6;
207
208     const byte mascara_para_recuperar_tipo_coluna = 3; // 0000 0011 em
209     binário
210
211     //08/10/11
212     //recuperação dos bytes transmitidos
213     foreach (byte myData in args.data)
214     {
215         //responsável pela recepção dos pacotes de bytes
216         rec_data += "[byte: " + count + " ] ";
217
218         //primeiros pacotes
219         if (myData.ToString().Length == 1)
220         {
221             rec_data += "00";
222         }
223
224         //primeiros pacotes
225         if (myData.ToString().Length == 2)
226         {
227             rec_data += "0";
228         }
229
230         rec_data += myData.ToString() + " ";
231
232         //detecção do começo de uma leitura
233         if (count_coluna == 2 && Convert.ToInt32(leitura_anterior.ToString
234         ()) == 128 && Convert.ToInt32(myData.ToString()) == 128)
235         {
236             leitura_anterior = myData;
237             count_coluna++;
238             count++;
239             rec_data += "[Sequência START detectada]";
240             break;
241         }
242
243         //detecção do final de uma leitura
244         if (count_coluna == 2 && Convert.ToInt32(leitura_anterior.ToString
245         ()) == 150 && Convert.ToInt32(myData.ToString()) == 150)
246         {
247             leitura_anterior = myData;
248             count_coluna++;
249             count++;
250             rec_data += "[Sequência END detectada]";
251
252             //copia ultima coluna válida para queue final
253             if (coluna_reversa == 0)
254             {
255                 // Cria novo vetor com Length igual a contagem de elementos
256                 da fifa temporária
```

```
255         int[] array = new int[qe_leituraZ_temp.Count];
256
257         // Copia fila para vetor criado
258         qe_leituraZ_temp.CopyTo(array, 0);
259         for (int i = 0; i < array.Length; i++)
260         {
261             qe_leituraZ.Enqueue(array[i]);
262         }
263
264         //limpa marcacao de coluna
265         coluna_reversa = 2;
266
267         //limpa queue
268         qe_leituraZ_temp.Clear();
269
270     }
271
272     if (coluna_reversa == 1)
273     {
274
275         // Cria novo vetor com Length igual a contagem de elementos
276         // da fifa temporária
277         int[] array = new int[qe_leituraZ_temp.Count];
278
279         // Copia fila para vetor criado
280         qe_leituraZ_temp.CopyTo(array, 0);
281
282         // Salva valores na fila do último ao primeiro (FILO)
283         for (int i = array.Length - 1; i >= 0; i--)
284         {
285             qe_leituraZ.Enqueue(array[i]);
286         }
287
288         //limpa marcacao de coluna
289         coluna_reversa = 2;
290
291         //limpa queue
292         qe_leituraZ_temp.Clear();
293     }
294
295
296     break;
297 }
298
299
300
301
302
303
304
305
306 //11/10/11
```

```
307 //Varredura de bytes do pacote recebido
308 if (count < 61)
309 {
310     switch (count_coluna)
311     {
312         case 0:
313             //ignora
314             break;
315         case 1:
316             //ignora
317             break;
318         case 2:
319             //ignora
320             break;
321         case 3:
322             //ignora
323             break;
324         case 4:
325             //ignora
326             break;
327         case posicao_z_mais_significativo:
328             //recupera aqui a parte mais significativa da posição Z
329
330             //processamento realizado apenas após a recepção do
331             byte menos significativo
332             //que está no 'case' abaixo
333
334             break;
335         //recupera aqui a parte menos significativa da posição Z e
336         //salva com a mais significativa
337         case posicao_z_menos_significativo:
338             qe_leituraZ_temp.Enqueue(Convert.ToInt32(
339                 leitura_anterior.ToString()) * fator_de_multiplicacao +
340                 Convert.ToInt32(myData.ToString()));
341             break;
342         case 7:
343             //whaat?? se chegar aqui, algo sai muito errado. =0
344             break;
345     }
346 }
347 else
348 {
349     //ATENÇÃO: bytes de controle
350     //byte 1,2 e 3 -> contagem de pacotes
351     //byte 3 - bits 1 e 2 -> marcação de coluna:
352     // 00,11 -> estado não permitido
353     // 01 -> coluna_par (coluna invertida)
354     // 10 -> coluna_impar
355
356     if (count == 63)
357     {
```

```
356     int recuperar_tipo_coluna =
357     mascara_para_recuperar_tipo_coluna & myData;
358
359     if (recuperar_tipo_coluna == 1) // 1 -> 0000 0001
360     (indicação de coluna par)
361     {
362         // Vetor em ordem reversa
363         rec_data += " [ordem reversa detectada] ";
364
365         if (coluna_reversa == 0) //mudança de padrão na coluna
366         DETECTADA
367         {
368             //copia de modo direto da fila temporaria -> fila
369             real
370             // Salva valores na fila do primeiro ao último (FIFO)
371
372             // Cria novo array com o mesmo tamanho de elementos
373             que fila temporária
374             int[] array = new int[qe_leituraZ_temp.Count];
375
376             // Copia fila para o vetor
377             qe_leituraZ_temp.CopyTo(array, 0);
378             for (int i = 0; i < array.Length-10; i++)
379             {
380                 qe_leituraZ.Enqueue(array[i]);
381             }
382
383             //inicia a coluna invertida
384             coluna_reversa = 1;
385
386             //limpa queue
387             qe_leituraZ_temp.Clear();
388             for (int i = array.Length - 10; i < array.Length; i
389             ++))
390             {
391                 qe_leituraZ_temp.Enqueue(array[i]);
392             }
393
394             }
395
396             if (coluna_reversa == 2) //não inicializada
397             {
398                 coluna_reversa = 1;
399             }
400
401         }
402     }
403
404     else
405     {
406         // Vetor em ordem direta
407         rec_data += " [ordem direta] ";
408
409         if (coluna_reversa == 1) //mudança de padrão na coluna
```

```
DETECTADA
403     {
404
405         // Cria novo array com o mesmo tamanho de elementos
406         // que fila temporária
407         int[] array = new int[qe_leituraZ_temp.Count];
408
409         // Copia fila para o vetor
410         qe_leituraZ_temp.CopyTo(array, 0);
411
412         // Salva valores na fila do último ao primeiro (FILO)
413         for (int i = array.Length - 1; i >= 0; i--)
414         {
415             qe_leituraZ.Enqueue(array[i]);
416         }
417
418         //inicia a coluna direta
419         coluna_reversa = 0;
420
421         //limpa queue
422         qe_leituraZ_temp.Clear();
423         for (int i = array.Length - 1; i < array.Length; i
424             ++
425         )
426         {
427             qe_leituraZ_temp.Enqueue(array[i]);
428         }
429
430         if (coluna_reversa == 2) //não inicializada
431         {
432             coluna_reversa = 0;
433         }
434
435     }
436
437 }
438
439 }
440 if (count_coluna == 6)
441 {
442     count_coluna = 1;
443 }
444 else
445 {
446     count_coluna++;
447 }
448
449
450 count++;
451 leitura_anterior = myData;
452
```

```
453     }
454
455     //08/10/11
456     //populando dados dentro da caixa de texto
457     dadoscoletados = dadoscoletados + rec_data + "\n";
458
459     conta_pacotes_validos = conta_pacotes_validos + 1;
460     tb_conta_pacotes.Text = conta_pacotes_validos.ToString();
461
462     }
463 }
464
465 private void usb_OnDataSend(object sender, EventArgs e)
466 {
467     this.lb_message.Items.Add("Dados enviados");
468 }
469
470 private void btn_desconectar_Cli(object sender, EventArgs e)
471 {
472     this.usb.ProductId = 0;
473     this.usb.VendorId = 0;
474 }
475
476 private void tb_send_KeyPress(object sender, KeyPressEventArgs e)
477 {
478     if (e.KeyChar == (char)13)
479     {
480         // Enter pressionado
481         btn_send_Click(sender, e);
482     }
483 }
484
485 private void btn_save_csv_Click(object sender, EventArgs e)
486 {
487     StreamWriter TargetFile, LogDados;
488
489     //23/10/2011
490     //pasta p recepção das amostras
491     string pasta_de_amostras = "C:\\output\\";
492
493     //cria diretório se não existir
494     if (!Directory.Exists(pasta_de_amostras))
495     {
496         Directory.CreateDirectory(pasta_de_amostras);
497     }
498
499
500     TargetFile = new StreamWriter(pasta_de_amostras + "Dados" + indice_arquivo.
501     ToString() + ".csv", true);
502
503     //sem dados recebidos
504     if (qe_leituraZ.Count == 0)
505     {
```

```
505     MessageBox.Show("Arquivo salvo sem informações válidas. Nenhum dado no
506     TargetFile.WriteLine();
507     TargetFile.Write("ERRO: sem dados para salvar.");
508     TargetFile.WriteLine();
509     TargetFile.WriteLine();
510
511     TargetFile.Close();
512 }
513 else
514 {
515     while (qe_leituraZ.Count > 0)
516     {
517         acrescenta_linha(qe_leituraZ.Dequeue().ToString() + ";");
518     }
519
520     TargetFile.WriteLine();
521     TargetFile.Write(dadosformatados);
522     TargetFile.WriteLine();
523     TargetFile.WriteLine();
524
525     TargetFile.Close();
526
527     LogDados = new StreamWriter(pasta_de_amostras + "LogDados" +
528     indice_arquivo.ToString() + ".txt", true);
529
530     LogDados.WriteLine();
531     LogDados.Write(dadoscoletados);
532     LogDados.WriteLine();
533     LogDados.WriteLine();
534
535     LogDados.Close();
536
537
538     //eh o mesmo que limpar o buffer
539     dadosformatados = dadosformatados.Remove(0);
540     dadoscoletados = dadoscoletados.Remove(0);
541
542     //incrementa o indice no nome do arquivo,
543     //que permite salvar vários arquivos com nomes diferentes
544     indice_arquivo++;
545 }
546
547
548 if (ckb_limpar.Checked == true)
549 {
550     btn_clear_buffer_Click(sender, e);
551 }
552 }
553
554 //Botão para limpar tudo
555 private void btn_clear_buffer_Click(object sender, EventArgs e)
```

```
556     {
557         lb_message.Items.Clear();
558         lb_read.Items.Clear();
559         conta_pacotes_validos = 0;
560         tb_conta_pacotes.Text = conta_pacotes_validos.ToString();
561     }
562
563     private void ckb_limpar_MouseClick(object sender, MouseEventArgs e)
564     {
565         if (ckb_limpar.Checked == true)
566         {
567             btn_save_csv.Text = "Imprimir CSV e limpar";
568             btn_clear_buffer.Enabled = false;
569         }
570         else
571         {
572             btn_save_csv.Text = "Imprimir CSV";
573             btn_clear_buffer.Enabled = true;
574         }
575     }
576 }
577
578 private void notify_icon_stm_DoubleClick(object sender, EventArgs e)
579 {
580     if ((Control.MouseButtons & MouseButtons.Left) == MouseButtons.Left)
581     {
582         if (this.Visible == false)
583         {
584             this.Visible = true;
585             this.WindowState = 0;
586         }
587         else
588         {
589             this.Visible = false;
590         }
591     }
592 }
593
594 private void gerenciador_usb_SizeChanged(object sender, EventArgs e)
595 {
596     if (this.WindowState == System.Windows.Forms.FormWindowState.Minimized)
597     {
598         {
599             // this.Visible = false;
600         }
601     }
602 }
603 private void bt_enviar_parâmetros_Click(object sender, EventArgs e)
604 {
605     // 23/10/2011
606     //Reconstroi interface original
607
608     //limpa string de envio
```

```
609     string text = "";
610
611     //parametrização de for APENAS para simulação
612     for (int i = 0; i < this.usb.SpecifiedDevice.OutputReportLength - 1; i++)
613     {
614         switch (i)
615         {
616             case 1:
617                 //código estático
618                 text += "073 ";
619                 break;
620             case 2:
621                 //código START de gravação
622                 //informa ao PIC que os novos parâmetros
623                 text += "255 ";
624                 break;
625             case 3:
626                 //min_x
627                 text += escreve_texto_numerico_em_dois_bytes(tb_parametros_min_x
628                     .Text.ToString());
629                 break;
630             case 4:
631                 //não faz nada
632                 //ação já realizada no case anterior
633                 break;
634             case 5:
635                 //max_x
636                 text += escreve_texto_numerico_em_dois_bytes(tb_parametros_max_x
637                     .Text.ToString());
638                 break;
639             case 6:
640                 //não faz nada
641                 //ação já realizada no case anterior
642                 break;
643             case 7:
644                 //min_y
645                 text += escreve_texto_numerico_em_dois_bytes(tb_parametros_min_y
646                     .Text.ToString());
647                 break;
648             case 8:
649                 //não faz nada
650                 //ação já realizada no case anterior
651                 break;
652             case 9:
653                 //max_y
654                 text += escreve_texto_numerico_em_dois_bytes(tb_parametros_max_y
655                     .Text.ToString());
656                 break;
657             case 10:
658                 //não faz nada
659                 //ação já realizada no case anterior
660                 break;
661             case 11:
```

```
658         //passo
659         text += escreve_texto_numerico_em_dois_bytes(tb_parametros_passo
        .Text.ToString());
660         break;
661     case 12:
662         //não faz nada
663         //ação já realizada no case anterior
664         break;
665     case 13:
666         //temporização da leitura
667         text += escreve_texto_numerico_em_dois_bytes(
        tb_parametros_tempo_leitura.Text.ToString());
668         break;
669     case 14:
670         //não faz nada
671         //ação já realizada no case anterior
672         break;
673     case 15:
674         //quantidade de amostras
675         text += escreve_texto_numerico_em_dois_bytes(
        tb_parametros_qtde_amostras.Text.ToString());
676         break;
677     case 16:
678         //não faz nada
679         //ação já realizada no case anterior
680         break;
681
682     default:
683         //padrão para próximas linhas
684         text += "000 ";
685         break;
686
687     }
688 }
689 this.tb_send.Text = text;
690
691 btn_send_Click(sender, e);
692 usb_OnSpecifiedDeviceArrived(sender, e);
693
694
695     }
696 }
697 }
```

Anexo H

Código matlab para calculo da distancia entre amostra-sonda e geração da imagem.

```
1  function filename = gerar_imagem(mapa_de_cores, Azimute, Elevacao,  
   endereco_do_arquivo_csv)  
2  
3  %Caminho para salvar imagem dentro do servidor WEB  
4  filename = 'C:\Inetpub\wwwroot\images\vizoption';  
5  
6  OutHandle = figure('Numbertitle', 'off', 'visible','off',...  
7                   'Menubar', 'none', 'Name', 'Option Pricing Tool', ...  
8                   'Tag', 'OutputFigure');  
9  
10  
11  %Rotina para importar CSV  
12  DELIMITER = ';';  
13  HEADERLINES = 1;  
14  FILE_TO_READ = endereco_do_arquivo_csv;  
15  
16  %Comando para importar imagem  
17  newdata = importdata(FILE_TO_READ,DELIMITER,HEADERLINES);  
18  data_Z = newdata.('data');  
19  
20  
21  
22  
23  
24  
25  if (size (data_Z,1) < 10000),  
26  % Condição para imagem 50 x 50  
27      numero_de_colunas = 5;  
28  else  
29  % Imagem maior que 50 x 50  
30      numero_de_colunas = 20;  
31  end  
32  
33  %atribuição de variável de amostras por colunas  
34  amostras_por_colunas = numero_de_colunas*10;  
35  
36  
37  if (size (data_Z,1) < 40000 && size (data_Z,1) >= 10000),  
38  % Condição para imagem 50 x 200  
39      [X,Y] = meshgrid(0:1:amostras_por_colunas-1,0:1:amostras_por_colunas-151);  
40  else  
41  % Imagem 200 x 200  
42      [X,Y] = meshgrid(0:1:amostras_por_colunas-1);  
43  end  
44  
45  % Imagem 200 x 200  
46      if (size (data_Z,1) >= 40000),  
47          for i=1:amostras_por_colunas,  
48              Z(i,1:amostras_por_colunas) = data_Z(amostras_por_colunas*i-(  
49                  amostras_por_colunas-1):amostras_por_colunas*i);  
50          end  
51  end
```

```

52 % Imagem 50 x 200
53     if (size (data_Z,1) < 40000 && size (data_Z,1) >= 10000),
54         for i=1:amostras_por_colunas-150,
55             Z(i,1:amostras_por_colunas) = data_Z(amostras_por_colunas*i-(
                    amostras_por_colunas-1):amostras_por_colunas*i);
56         end
57     end
58
59 % Imagem 50 x 50
60     if (size (data_Z,1) <10000),
61         for i=1:amostras_por_colunas,
62             Z(i,1:amostras_por_colunas) = data_Z(amostras_por_colunas*i-(
                    amostras_por_colunas-1):amostras_por_colunas*i);
63         end
64     end
65
66
67 % Cálculo da altura - Componente Z
68 Z = -log(Z/(0.07*1024*1.1*10^8))/2.15;
69
70 %Gera gráfico da imagem
71 surf(X,Y,Z);
72
73 % Se você está realmente lendo esse código nosso PARABÉNS!
74 % VOCÊ é uma pessoa ESPECIAL, no bom sentido ;- )
75 % Tem um brinde para VOCÊ com os nossos orientadores, lá no CPGEI. (#NOT!)
76
77 %específico para plotar gráfico com largura 4x maior que comprimento
78 %dimensionamento dos eixos
79 if (size (data_Z,1) < 40000 && size (data_Z,1) >= 10000),
80     pbaspect([4 1 1])
81 end
82
83
84 switch mapa_de_cores
85     case 1
86         colormap bone;
87     case 2
88         colormap flag;
89     case 3
90         colormap hsv;
91     case 4
92         colormap cool;
93     case 5
94         colormap gray;
95     case 6
96         colormap pink;
97     case 7
98         colormap copper;
99     otherwise
100         colormap copper;
101 end
102

```

```
103     xlabel('Componente X');
104     ylabel('Componente Y');
105     zlabel('Componente Z');
106     title('Amostra');
107     view([Azimute Elevacao-90]);
108
109
110     print(OutHandle, '-djpeg', filename);
111
112     close all;
113
114     end
115
116
117
118
```

Anexo I

Software da Galeria Web STM

```
1  using System;
2  using System.Collections;
3  using System.ComponentModel;
4  using System.Web;
5  using System.Web.SessionState;
6
7  namespace Galeria_WEB
8  {
9      /// <summary>
10     /// Summary description for Global.
11     /// </summary>
12     public class Global : System.Web.HttpApplication
13     {
14         /// <summary>
15         /// Required designer variable.
16         /// </summary>
17         private System.ComponentModel.IContainer components = null;
18
19         public Global()
20         {
21             InitializeComponent();
22         }
23
24         protected void Application_Start(Object sender, EventArgs e)
25         {
26
27         }
28
29         protected void Session_Start(Object sender, EventArgs e)
30         {
31
32         }
33
34         protected void Application_BeginRequest(Object sender, EventArgs e)
35         {
36
37         }
38
39         protected void Application_EndRequest(Object sender, EventArgs e)
40         {
41
42         }
43
44         protected void Application_AuthenticateRequest(Object sender, EventArgs e)
45         {
46
47         }
48
49         protected void Application_Error(Object sender, EventArgs e)
50         {
51
52         }
53
```

```
54     protected void Session_End(Object sender, EventArgs e)
55     {
56
57     }
58
59     protected void Application_End(Object sender, EventArgs e)
60     {
61
62     }
63
64     #region Web Form Designer generated code
65     /// <summary>
66     /// Required method for Designer support - do not modify
67     /// the contents of this method with the code editor.
68     /// </summary>
69     private void InitializeComponent()
70     {
71         this.components = new System.ComponentModel.Container();
72     }
73     #endregion
74 }
75 }
76
77
```

```
1  //-----
2  // <auto-generated>
3  //     This code was generated by a tool.
4  //
5  //     Changes to this file may cause incorrect behavior and will be lost if
6  //     the code is regenerated.
7  // </auto-generated>
8  //-----
9
10 namespace Blacksholes {
11
12
13     public partial class index {
14
15         /// <summary>
16         /// form1 control.
17         /// </summary>
18         /// <remarks>
19         /// Auto-generated field.
20         /// To modify move field declaration from designer file to code-behind file.
21         /// </remarks>
22         protected global::System.Web.UI.HtmlControls.HtmlForm form1;
23
24         /// <summary>
25         /// TreeView1 control.
26         /// </summary>
27         /// <remarks>
28         /// Auto-generated field.
29         /// To modify move field declaration from designer file to code-behind file.
30         /// </remarks>
31         protected global::System.Web.UI.WebControls.TreeView TreeView1;
32
33         /// <summary>
34         /// Imagem control.
35         /// </summary>
36         /// <remarks>
37         /// Auto-generated field.
38         /// To modify move field declaration from designer file to code-behind file.
39         /// </remarks>
40         protected global::System.Web.UI.WebControls.Image Imagem;
41
42         /// No Easter eggs here! Sorry, pal! Equipe STM BRASIL. ;)
43
44         /// <summary>
45         /// bt_troca control.
46         /// </summary>
47         /// <remarks>
48         /// Auto-generated field.
49         /// To modify move field declaration from designer file to code-behind file.
50         /// </remarks>
51         protected global::System.Web.UI.WebControls.Button bt_troca;
52     }
53 }
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.UI;
6  using System.Web.UI.WebControls;
7  using System.IO;
8
9  namespace Galeria_WEB
10 {
11     public partial class index : System.Web.UI.Page
12     {
13         protected void Page_Load(object sender, EventArgs e)
14         {
15
16         }
17
18         protected void TreeView1_SelectedNodeChanged(object sender, EventArgs e)
19         {
20
21         }
22
23         protected void bt_troca_Click(object sender, EventArgs e)
24         {
25             //Imagem.LançaFoguete.Sim(#not);)
26
27             Imagem.ImageUrl = "alerta.jpg";
28             Imagem.Width = 321;
29             Imagem.Height = 326;
30         }
31     }
32 }
```



```
48         <asp:TreeNode Text="leitura 2" Value="leitura 2"></asp:TreeNode>
49     </asp:TreeNode>
50 </Nodes>
51 <NodeStyle Font-Names="Tahoma" Font-Size="12pt" Font-Underline="False"
52     ForeColor="#000066" HorizontalPadding="2px" NodeSpacing="0px"
53     VerticalPadding="2px" />
54 <ParentNodeStyle Font-Bold="True" Font-Size="30pt" />
55 <RootNodeStyle Font-Size="14pt" Font-Underline="False" ForeColor="Black" />
56 <SelectedNodeStyle BackColor="#B5B5B5" Font-Underline="False"
57     HorizontalPadding="0px" VerticalPadding="0px" />
58 </asp:TreeView>
59 <br />
60 <br />
61 </div>
62 <div style="border-style: double; height: 345px; margin-left: 878px;
background-color: #FFFFCC; position: fixed; z-index: auto; top: 45px; left: -288px;
width: 413px; margin-top: 55px; vertical-align: middle;"
63     align="center">
64     <asp:Image ID="Imagem" runat="server" Height="123px" ImageAlign="Middle"
65         ImageUrl="utfpr.gif" style="text-align: left" Width="321px" />
66 </div>
67 <p class="style1">
68     &nbsp;</p>
69 <div style="position: absolute; top: 35px; left: 877px; height: 25px; width: 152px;
text-align: center;">
70     <asp:Button ID="bt_troca" runat="server" Font-Size="8pt" Height="24px"
71         onclick="bt_troca_Click" Text="Amostra remota" Width="101px" />
72 </div>
73 </form>
74 </body>
75 </html>
76
```

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Web;
5  using System.Web.UI;
6  using System.Web.UI.WebControls;
7  using System.Collections;
8  using System.ComponentModel;
9  using System.Data;
10 using System.Drawing;
11 using System.Web.SessionState;
12 using System.Web.UI.HtmlControls;
13 using System.IO;
14
15 using MathWorks.MATLAB.NET.Arrays;
16 using Comp_Gerar_Imagem;
17
18
19 namespace Galeria_WEB
20 {
21     public partial class resultado : System.Web.UI.Page
22     {
23         int valor_padrao_azimute = 15;
24         int valor_padrao_elevacao = 135;
25         int passo_padrao = 5;
26         int valor_maximo = 180;
27         double valor_minimo = 0;
28
29
30
31         protected void Page_Load(object sender, EventArgs e)
32         {
33
34             //polula lista de azimute caso esteja vazio
35             if (lb_azimute.Items.Count < 10)
36             {
37                 double i;
38                 lb_azimute.Items.Clear();
39                 for (i = valor_minimo; i < valor_maximo; i = i + passo_padrao)
40                 {
41                     lb_azimute.Items.Add(new ListItem(i.ToString(), Convert.ToString(i +
42                                     1)));
43                 }
44                 if (Session["azimute"] == null)
45                 {
46                     lb_azimute.SelectedValue = Convert.ToString(valor_padrao_azimute + 1
47                                     );
48                 }
49                 else
50                 {
51                     lb_azimute.SelectedValue = Session["azimute"].ToString();
52                 }
53             }
54         }
55     }
56 }
```

```
52     else
53     {
54         Session["azimute"] = lb_azimute.SelectedValue.ToString();
55     }
56
57     //popula lista de elevacao caso esteja vazia
58     if (lb_elevacao.Items.Count < 10)
59     {
60         double i;
61         lb_elevacao.Items.Clear();
62         for (i = valor_minimo; i < valor_maximo; i = i + passo_padrao)
63         {
64             lb_elevacao.Items.Add(new ListItem(i.ToString(), Convert.ToString(i+
65                 1)));
66         }
67
68         if (Session["elevacao"] == null)
69         {
70             lb_elevacao.SelectedValue = Convert.ToString(valor_padrao_elevacao+1
71                 );
72         }
73         else
74         {
75             lb_elevacao.SelectedValue = Session["elevacao"].ToString();
76         }
77     }
78     else
79     {
80         Session["elevacao"] = lb_elevacao.SelectedValue.ToString();
81     }
82
83     double Azimute= Convert.ToInt32(lb_azimute.SelectedValue.ToString());
84     double Elevacao= Convert.ToInt32(lb_elevacao.SelectedValue.ToString());
85
86     //CARACAA! Você realmente tá lendo isso aqui?!
87     //NOSSO MUITO OBRIGADO !! =)
88     //EQUIPE STM BRASIL
89
90     Comp_Gerar_ImagemClass Comp_Gerar_Imagem= new Comp_Gerar_ImagemClass();
91
92     //Recupera arquivo para plotar
93     if (Page.Request.QueryString["fonte"] != null)
94     {
95         Session["fonte"] = Page.Request.QueryString["fonte"];
96     }
97
98     //valor padrão
99     int colormap = 7;
100     //Recupera colormap
101     if (Page.Request.QueryString["cor"] != null)
102     {
```

```
103         Session["colormap"] = Page.Request.QueryString["cor"];
104     }
105
106     if (Session["colormap"] != null)
107     {
108         switch (Session["colormap"].ToString())
109         {
110             case "bone":
111                 colormap = 1;
112                 break;
113             case "flag":
114                 colormap = 2;
115                 break;
116             case "jet":
117                 colormap = 3;
118                 break;
119             case "cool":
120                 colormap = 4;
121                 break;
122             case "gray":
123                 colormap = 5;
124                 break;
125             case "pink":
126                 colormap = 6;
127                 break;
128             case "copper":
129             default:
130                 colormap = 7;
131                 break;
132         }
133     }
134     else
135     {
136         Session["colormap"] = "copper";
137         colormap = 7;
138     }
139
140     // Gerar jpeg
141     string fileName = ((MWCharArray) (Comp_Gerar_Imagem.gerar_imagem(colormap,
142     Azimute, Elevacao, Session["fonte"].ToString()))).ToString();
143
144
145     string[] diretorio = Session["fonte"].ToString().Split('\\');
146     lb_diretorio.Text = diretorio[diretorio.Length - 2];
147
148     DirectoryInfo di = new DirectoryInfo("C:\\\" + diretorio[diretorio.Length - 3
149     ] + "\\\" + diretorio[diretorio.Length - 2]);
150
151     tv_arquivos.Nodes.Clear();
152     FileInfo[] rgFiles = di.GetFiles("*.csv");
153
```

```
154
155     foreach (FileInfo fi in rgFiles)
156     {
157         TreeNode objArquivo = new TreeNode();
158         objArquivo.Text = fi.Name;
159         objArquivo.ToolTip = fi.FullName;
160         if (fi.Name == diretorio[diretorio.Length - 1])
161         {
162             objArquivo.Selected = true;
163             objArquivo.Text = objArquivo.Text + " (atual)";
164         }
165         else
166         {
167             objArquivo.NavigateUrl = "resultado.aspx?fonte=" + fi.FullName;
168         }
169
170         tv_arquivos.Nodes.Add(objArquivo);
171     }
172
173
174
175 }
176
177 protected void Button1_Click(object sender, EventArgs e)
178 {
179     Response.Redirect("index2.aspx");
180 }
181
182 protected void bt_configuracoes_Click(object sender, EventArgs e)
183 {
184     if (panel_configuracao.Visible == true)
185     {
186         panel_configuracao.Visible = false;
187         panel_configuracao.Height = 0;
188     }
189     else
190     {
191         panel_configuracao.Visible = true;
192         panel_configuracao.Height = 710;
193     }
194 }
195 }
196 }
```

```
1  //-----
2  // <auto-generated>
3  //     This code was generated by a tool.
4  //
5  //     Changes to this file may cause incorrect behavior and will be lost if
6  //     the code is regenerated.
7  // </auto-generated>
8  //-----
9
10 namespace Galeria_WEB {
11
12
13     public partial class resultado {
14
15         /// <summary>
16         /// form1 control.
17         /// </summary>
18         /// <remarks>
19         /// Auto-generated field.
20         /// To modify move field declaration from designer file to code-behind file.
21         /// </remarks>
22         protected global::System.Web.UI.HtmlControls.HtmlForm form1;
23
24         /// <summary>
25         /// Button1 control.
26         /// </summary>
27         /// <remarks>
28         /// Auto-generated field.
29         /// To modify move field declaration from designer file to code-behind file.
30         /// </remarks>
31         protected global::System.Web.UI.WebControls.Button Button1;
32
33         /// <summary>
34         /// bt_configuracoes control.
35         /// </summary>
36         /// <remarks>
37         /// Auto-generated field.
38         /// To modify move field declaration from designer file to code-behind file.
39         /// </remarks>
40         protected global::System.Web.UI.WebControls.Button bt_configuracoes;
41
42         /// <summary>
43         /// Image1 control.
44         /// </summary>
45         /// <remarks>
46         /// Auto-generated field.
47         /// To modify move field declaration from designer file to code-behind file.
48         /// </remarks>
49         protected global::System.Web.UI.WebControls.Image Image1;
50
51         /// #just.to.say.we.are.happy'coz.u.are.reading.this.. Háá! Equipe STM ;-))
52
53         /// <summary>
```

```
54     /// lb_diretorio control.
55     /// </summary>
56     /// <remarks>
57     /// Auto-generated field.
58     /// To modify move field declaration from designer file to code-behind file.
59     /// </remarks>
60     protected global::System.Web.UI.WebControls.Label lb_diretorio;
61
62
63     /// <summary>
64     /// tv_arquivos control.
65     /// </summary>
66     /// <remarks>
67     /// Auto-generated field.
68     /// To modify move field declaration from designer file to code-behind file.
69     /// </remarks>
70     protected global::System.Web.UI.WebControls.TreeView tv_arquivos;
71
72     /// <summary>
73     /// panel_configuracao control.
74     /// </summary>
75     /// <remarks>
76     /// Auto-generated field.
77     /// To modify move field declaration from designer file to code-behind file.
78     /// </remarks>
79     protected global::System.Web.UI.WebControls.Panel panel_configuracao;
80
81     /// really reading or just passing by? whatever, thank's! Equipe STM BRASIL =)
82
83     /// <summary>
84     /// cores control.
85     /// </summary>
86     /// <remarks>
87     /// Auto-generated field.
88     /// To modify move field declaration from designer file to code-behind file.
89     /// </remarks>
90     protected global::System.Web.UI.WebControls.ImageMap cores;
91
92     /// <summary>
93     /// lb_azimute control.
94     /// </summary>
95     /// <remarks>
96     /// Auto-generated field.
97     /// To modify move field declaration from designer file to code-behind file.
98     /// </remarks>
99     protected global::System.Web.UI.WebControls.ListBox lb_azimute;
100
101     /// Vc merece um prêmio! Procure outros Easter eggs ;-))
102
103     /// <summary>
104     /// lb_elevacao control.
105     /// </summary>
106     /// <remarks>
```

```
107         /// Auto-generated field.
108         /// To modify move field declaration from designer file to code-behind file.
109         /// </remarks>
110         protected global::System.Web.UI.WebControls.ListBox lb_elevacao;
111     }
112 }
113
```



```
94         <br />
95
96     </asp:Panel>
97
98 </div>
99 <div align="center"
100
101
102
103     style="width: 156px; background-color: #FFFFCC; position: absolute; top: 25px;
104     left: 870px; height: 676px;">
105     <table class="style2" style="border-style: double">
106         <tr>
107             <td class="style5">
108                 Azimute</td>
109             <td class="style5">
110                 Elevação</td>
111         </tr>
112         <tr>
113             <td class="style4">
114                 <asp:ListBox ID="lb_azimute" runat="server" AutoPostBack="True"
115                 Height="630px"
116                 Width="60px">
117                     <asp:ListItem Selected="True">50</asp:ListItem>
118                     <asp:ListItem>100</asp:ListItem>
119                 </asp:ListBox>
120             </td>
121             <td class="style4">
122                 <asp:ListBox ID="lb_elevacao" runat="server" AutoPostBack="True"
123                 BackColor="White" Height="630px" Width="60px">
124                     <asp:ListItem Selected="True">80</asp:ListItem>
125                     <asp:ListItem>0</asp:ListItem>
126                 </asp:ListBox>
127             </td>
128         </tr>
129     </table>
130     <br />
131
132 </div>
133 </div>
134 </body>
135 </html>
136
```