

ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO  
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E DE SISTEMAS  
MECÂNICOS

Nota final  
9,0 (uma)  
H.M.

OBTENÇÃO DE IMAGENS 3D DE UM MICROSCÓPIO DE TUNELAMENTO  
ELETRÔNICO

Carlos Akio Nakagawa  
Gustavo Vinicius Aguiar dos Santos

São Paulo  
2005

**ESCOLA POLITÉCNICA DA UNIVERSIDADE DE SÃO PAULO  
DEPARTAMENTO DE ENGENHARIA MECATRÔNICA E DE SISTEMAS  
MECÂNICOS**

**OBTENÇÃO DE IMAGENS 3D DE UM MICROSCÓPIO DE TUNELAMENTO  
ELETRÔNICO**

Trabalho de formatura apresentado à Escola  
Politécnica da Universidade de São Paulo para  
obtenção do título de Graduação em Engenharia  
Mecânica – Ênfase: Automação e Sistemas

**Carlos Akio Nakagawa  
Gustavo Vinicius Aguiar dos Santos**

**Orientador: Prof. Dr. Delson Torikai**

**São Paulo  
2005**

DEDALUS - Acervo - EPMN



31600011840

1493915

## FICHA CATALOGRÁFICA

**Nakagawa, Carlos Akio**

**Obtenção de imagem 3D em um microscópio de tunelamento eletrônico / C.A. Nakagawa, G.V.A. dos Santos -- São Paulo, 2005.**

**p.**

**Trabalho de Formatura - Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos.**

**1.Nanotecnologia 2.Terceira dimensão 3.Microscópio eletrônico I.Santos, Gustavo Vinícius Aguiar dos II.Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos III.t.**

## Sumário

1. Introdução.....	6
2. Proposta.....	6
3. Fundamentos Teóricos.....	7
3.1 Conceitos Iniciais.....	7
3.2 Tunelamento Eletrônico.....	8
3.3 Modos de operação.....	13
3.3.1 Modo de corrente constante.....	13
3.3.2 Modo de altura constante.....	13
3.3.3 Imagem da altura da barreira.....	13
3.4 Aspectos técnicos.....	14
3.4.1 Posicionamento.....	15
3.4.1.1 Piezo-eletricidade.....	15
3.4.1.2 Posicionador refinado.....	16
3.4.2 Circuito eletrônico.....	17
3.4.3 Isolamento de vibração.....	18
3.4.4 Ponta.....	19
4. O microscópio de tunelamento eletrônico.....	20
4.1 Componentes mecânicos.....	21
4.2 Componentes eletrônicos.....	22
4.3 Software.....	24
5. Montagem do experimento.....	24
5.1 Montagem das placas.....	24
5.2 Montagem da mesa inercial.....	24
5.3 Verificação do piezoelemento.....	25
5.3.1 Procedimento experimental.....	25
5.3.2 Resultados.....	27
5.3.3 Discussão dos resultados.....	28
6. Software.....	29
6.1 Adaptação do software.....	29
6.2 Software de visualização de imagem 3D.....	32
7. Considerações finais.....	34

Referências.....	35
Anexo 1 .....	36
Anexo 2.....	41
Anexo 3.....	43

## Lista de Figuras

Figura 1 - Tunelamento eletrônico.....	7
Figura 2 - Barreira de potencial.....	8
Figura 3 - Tunelamento do elétron dependente do estado de densidades.....	10
Figura 4 - Imagem dos estados ocupados de .....	11
Figura 5 - Imagem dos estados desocupados de .....	11
Figura 6- Modelo da ponta.....	12
Figura 7 - Densidade de estados altamente ondulada.....	13
Figura 8 - Efeito piezo-elétrico.....	15
Figura 9 - Sistema de posicionamento em tripé.....	16
Figura 10 - Sistema de posicionamento bimorfo.....	16
Figura 11 - Sistema de posicionamento tubo.....	17
Figura 12 - Esquema do circuito eletrônico.....	18
Figura 13 - Ponta de pequena curvatura.....	19
Figura 14 - Ponta de um átomo.....	19
Figura 15 - Desgaste eletroquímico da ponta.....	20
Figura 16 - Diagrama em blocos do carregamento da imagem.....	29
Figura 17 - Diagrama de blocos da exportação dos dados.....	30
Figura 18 - Tela criada para exportação.....	31
Figura 19 - Tela principal com imagem de superfície de grafite carregada.....	32
Figura 20 - Demonstração de zoom da imagem.....	33
Figura 21 - Demonstração das capacidades de rotação.....	33

## **1. Introdução**

Atualmente, a nanotecnologia vem sendo utilizada cada vez mais em diferentes aplicações. Ela vem sendo empregada em pesquisas acadêmicas, desenvolvimento de eletro-eletrônicos, tecidos e no estudo da estrutura e propriedade de materiais, entre outros.

O estudo da nanociência e nanotecnologia só alavancou a partir de 1986, com o desenvolvimento do primeiro Microscópio de Tunelamento Eletrônico (STM), pela IBM. Este microscópio era muito sofisticado para a época e teve muitas complicações no seu desenvolvimento. Com o avanço tecnológico, muitos componentes daquele microscópio puderam ser substituídos e novos projetos de STM foram desenvolvidos. A possibilidade de simplificação do projeto do STM, aliado a um interesse acadêmico, levou a Universidade de Muenster a desenvolver um projeto simplificado deste equipamento e disponibilizá-lo na forma de um kit com os componentes necessários para sua montagem.

Este trabalho vai ser realizado utilizando-se um kit, fornecido pela Universidade de Muenster em conjunto com o Instituto Tecnológico Brasil – Alemanha, como base.

## **2. Proposta**

Neste trabalho foi criado um software que permite a visualização de imagens em três dimensões a partir dos dados obtidos com o microscópio de tunelamento eletrônico acima mencionado. Para isso, realizou-se inicialmente um estudo dos princípios de funcionamento de um STM, as grandezas físicas envolvidas como precisão necessária dos componentes mecânicos móveis, sinais de corrente de tunelamento, forma de alimentação, coleta, amplificação e processamento desses sinais e o sistema de controle utilizado. Em seguida, foi feita a montagem do kit fornecido pela Universidade de Muenster e ITBA, e finalmente realizou-se o processo de desenvolvimento do software para a visualização de imagens 3D.

### 3. Fundamentos Teóricos

#### 3.1 Conceitos Iniciais

O primeiro Microscópio de Tunelamento Eletrônico (STM) foi criado por G. Binnig e W. Rohrer no Laboratório de Pesquisas da IBM, em 1986. Desde então, este instrumento tem sido de total importância no estudo de estruturas atômicas e de superfícies.

A idéia básica do funcionamento do microscópio é aproximar a ponta de uma agulha fina de metal até a alguns Å de uma amostra metálica. Aplicando-se uma diferença de potencial ( $U \leq 4V$ ) entre a ponta e a amostra, uma pequena corrente elétrica (0,01 nA – 50 nA) poderá passar entre ambas, apesar de não haver contato físico entre as duas partes. Este fenômeno é o tunelamento eletrônico. Este princípio de funcionamento pode ser observado na Figura 1 abaixo.

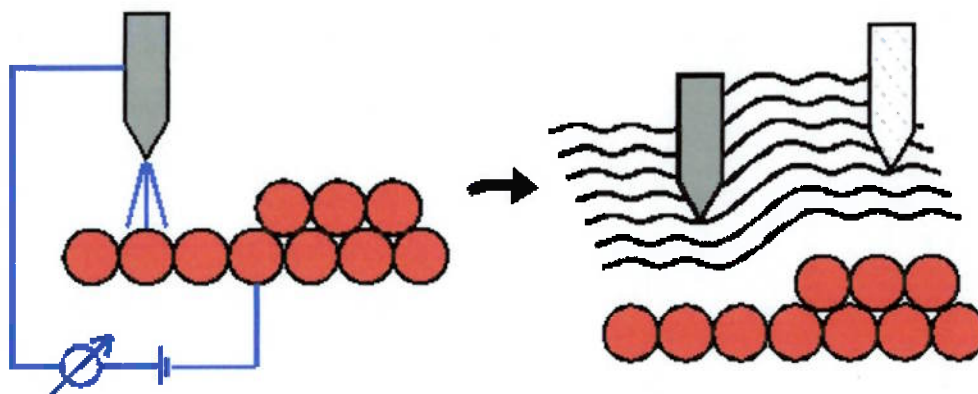


Figura 1 - Tunelamento eletrônico

A variação da corrente, quando se altera a distância entre a ponta e a amostra, se dá exponencialmente. Isto dá uma ótima resolução vertical na medição. Fazendo-se uma varredura da superfície e detectando a corrente (pode-se utilizar a corrente como sinal para identificar a altura da ponta em relação à superfície), um mapa com da superfície pode ser gerado, com uma resolução da ordem de nanômetros.

Temos que evidenciar o fato que o mapa obtido não pode ser utilizado para apenas interpretar diretamente a topologia da superfície. A razão é que a ponta é afetada por variações laterais e verticais do estado de densidade eletrônico da superfície. Para minimizar esses efeitos, o microscópio pode ser utilizado no vácuo ou em meios materiais propícios.



### 3.2 Tunelamento Eletrônico

Na física clássica, um elétron não pode penetrar ou atravessar uma barreira de potencial se sua energia  $E$  for menor que o potencial  $\Phi$  da barreira. A mecânica quântica prevê um decaimento exponencial da função da onda dentro desta barreira. Isto é representado na Figura 2 abaixo. Para uma barreira retangular, temos o seguinte conjunto de equações:

$$\Psi(d) = \Psi(0)e^{-\kappa d} \quad (1),$$

onde  $\kappa = \frac{\sqrt{2m(\Phi - E)}}{\hbar} \quad (2).$

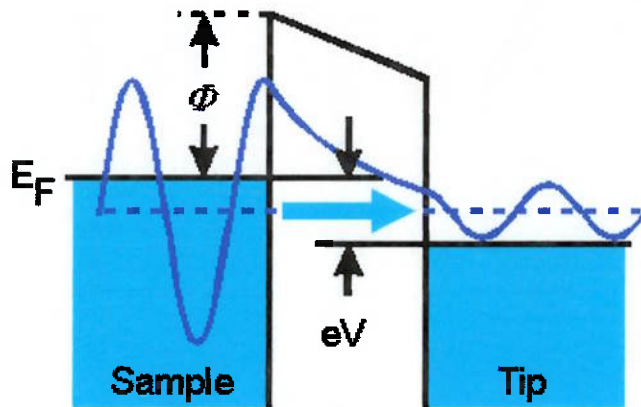


Figura 2 - Barreira de potencial

A probabilidade de se encontrar um elétron dentro de uma barreira de potencial de largura  $d$  é:

$$W(d) = |\Psi(d)|^2 = |\Psi(0)|^2 e^{-2\kappa d} \quad (3)$$

No STM, uma voltagem é aplicada entre a ponta e a amostra, de tal forma que campo elétrico criado faz com que o tunelamento de elétrons isolados se transforme em um tunelamento de uma corrente elétrica  $I$ . A altura da barreira pode ser aproximada pela média da função de trabalho da ponta e da amostra.

$$\Phi = \frac{1}{2}(\Phi_{sample} + \Phi_{tip}) \quad (4)$$

Se a voltagem for muito menor que a função de trabalho  $eV \ll \Phi$ , o inverso do tamanho do decaimento para todos os elétrons do tunelamento pode ser simplificado para:

$$\kappa \approx \frac{\sqrt{2m\Phi}}{\hbar} \quad (5)$$

O valor da corrente elétrica é proporcional à probabilidade de se encontrar elétrons

dentro da barreira:

$$I \propto \sum_{E_n=E_f-eV}^{E_f} |\Psi_n(0)|^2 e^{-2\kappa d} \quad (6).$$

Usando a definição de densidade local dos estados para  $\varepsilon \rightarrow 0$  :

$$\rho(z, E) \equiv \frac{1}{\varepsilon} \sum_{E_n=E-\varepsilon}^E |\Psi_n(z)|^2 \quad (7).$$

A corrente elétrica pode ser expressa por:

$$I \propto V \rho_{sa}(0, E_f) e^{-2\kappa d} \\ \approx V \rho_{sa}(0, E_f) e^{-1,025 \sqrt{\Phi} d} \quad (8),$$

onde  $[d] = \text{\AA}$  e  $[\Phi] = \text{eV}$ .

Com 5 eV como um exemplo típico de valor da função de trabalho, uma mudança na distância entre a ponta e a amostra de 1  $\text{\AA}$  gera uma mudança aproximada de uma ordem de grandeza no valor da corrente elétrica. Isso proporciona ao microscópio uma grande resolução vertical. Ainda temos  $I \propto V \rho_{sa}(d, E_f)$ , o que significa que a corrente é proporcional à densidade local dos estados da amostra, na Energia de Firmini, com a ponta a uma distância  $d$  da amostra.

Um cálculo mais exato do problema da densidade da corrente elétrica numa barreira de potencial quadrada requer que a equação de Schrödinger seja resolvida em três regiões: na fronteira de entrada, dentro e na fronteira de saída da barreira. Os coeficientes têm que ser adaptados de tal forma que a solução obtida seja continuamente diferenciável. Definindo a probabilidade de transição como:

$$T = \frac{j_T}{j_i} \quad (9),$$

temos:

$$T = \frac{16E(V-E)}{V^2} e^{2\kappa d} ; \quad \kappa \approx \sqrt{\frac{2m(V-E)}{\hbar^2}} \quad (10),$$

com a aproximação  $\kappa d \gg 1$ .

A densidade da corrente é definida como:

$$j_z = \frac{-i\hbar}{2m} \left( \Psi^*(z) \frac{d\Psi}{dz} - \Psi(z) \frac{d\Psi^*}{dz} \right) \quad (11).$$

Para uma barreira que não é quadrada, o método WKB deve ser usado. Ele é mais adequado de ser utilizado quando o potencial é alterado pela voltagem aplicada e pela força que o elétron exerce sobre a barreira. Usando o WKB, a probabilidade de transição fica:

$$T(E) \propto \exp\left(\frac{-2}{\hbar} \int_0^1 \sqrt{2mV(z) - E} dz\right) \quad (12)$$

Uma outra forma de descrever o fenômeno do tunelamento eletrônico é a solução proposta por Bardeen, que usa a teoria da perturbação eletrônica dependente do tempo. A probabilidade de um elétron no estado  $\Psi$  em  $E_\Psi$  de realizar o tunelamento para o estado  $\chi$  em  $E_\chi$  é dado pela Regra de Ouro de Fermi:

$$w = \frac{2\pi}{\hbar} |M|^2 \delta(E_\Psi - E_\chi) \quad (13)$$

O elemento de matriz do tunelamento é dado pela integral sobre a superfície da barreira de potencial entre a ponta e a amostra:

$$M = \frac{\hbar}{2m} \int_{z=z_0} \left( \chi^* \frac{\partial \Psi}{\partial z} - \Psi \frac{\partial \chi^*}{\partial z} \right) dS \quad (14)$$

Aplicando uma voltagem  $V$  e aproximando a distribuição de Fermi a uma função discreta, ( $kT \ll \Delta E_{\text{resolução}}$ ), a corrente fica:

$$I = \frac{4\pi e}{\hbar} \int_0^{eV} \rho_{sa}(E_F - eV + \varepsilon) \rho_{tip}(E_F + \varepsilon) |M|^2 d\varepsilon \quad (15)$$

Temos, assim, que a corrente é dada por uma combinação de densidades dos estados da ponta e amostra, sendo que o elemento de matriz do tunelamento  $M$  serve como peso para essas densidades.

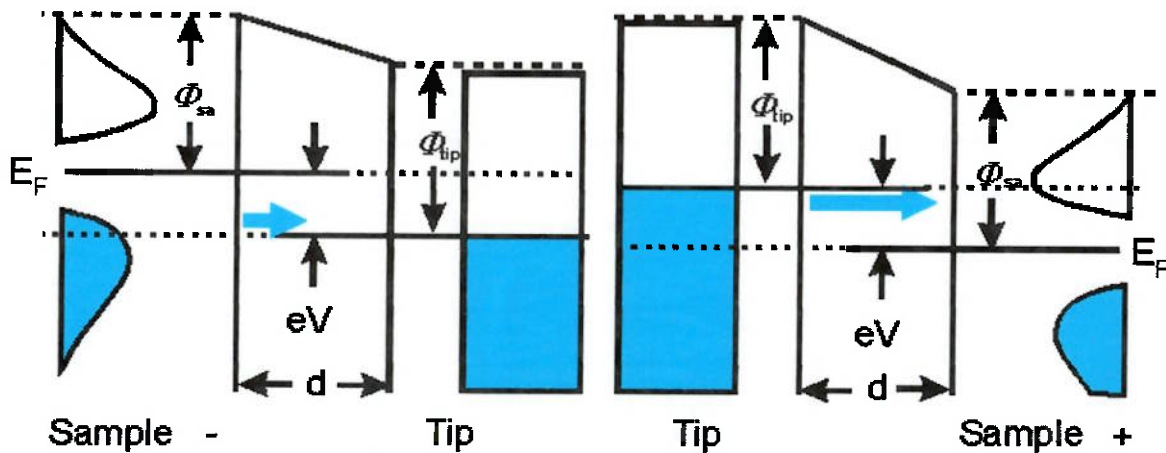
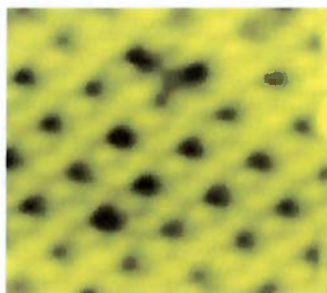


Figura 3 - Tunelamento do elétron dependente do estado de densidades

O termo  $\delta(E_F - E_\gamma)$  significa que um elétron só vai entrar no tunelamento se algum estado não ocupado com a mesma quantidade de energia existir no outro eletrodo (tunelamento inelástico não é tratado aqui). No caso de um potencial negativo na amostra, os estados ocupados geram a corrente e no caso de ser positivo, os espaços não ocupados são de importância. Portanto, como pode ser mostrado pelas imagens abaixo, uma imagem completamente diferente pode ser obtida com a mudança da voltagem. Isto ocorre porque estados diferentes contribuem para a corrente de tunelamento, dependendo do potencial utilizado. Isto é utilizado em tunelamento espectroscópico. Também deve ser mencionado que a probabilidade de tunelamento (expresso por  $M^2$ ) é maior para elétrons que estão mais perto da barreira de Fermi, graças ao fato de a barreira ser menor.



**Figura 4 - Imagem dos estados ocupados de  $\text{SiC}(000\bar{1})3\times 3$**



**Figura 5 - Imagem dos estados desocupados de  $\text{SiC}(000\bar{1})3\times 3$**

A resolução lateral do STM não pode ser entendida em termos de uma resolução de difração de Fraunhofer. O comprimento de onda correspondente do elétron no tunelamento seria  $\lambda > 10 \text{ \AA}$ . Portanto, o STM funciona em um regime de aproximação de campos. A curvatura geométrica média da ponta com um raio de, por exemplo,  $1000 \text{ \AA}$  e  $\kappa = 1 \text{ \AA}^{-1}$  seria necessária para uma resolução de aproximadamente  $50 \text{ \AA}$ .

A resolução atômica real pode ser entendida através da visão da mecânica quântica: o modelo que melhor se adapta ao problema proposto é o da ponta com ondas do tipo s. A ponta

é modelada como um pedaço protuberante de metal de Sommerfeld com um raio  $R$ . É assumido que apenas as soluções de ondas do tipo  $s$  (potencial esférico) deste problema de mecânica quântica são importantes. Este modelo é esquematizado na Figura 6. Neste caso, a baixas voltagens, a corrente de tunelamento é proporcional à densidade local de estados no centro de curvatura da ponta  $r_0$ :

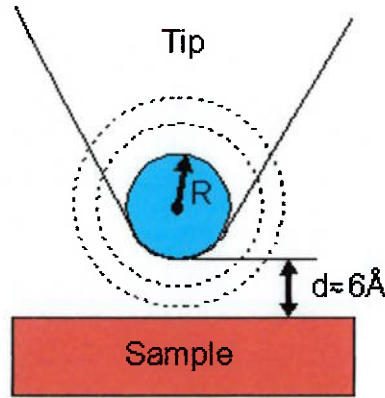


Figura 6- Modelo da ponta

$$I \propto \sum_{E_\nu = E_F - eV}^{E_F} |\Psi_\mu(r_0)|^2 = eV \rho_{sd}(r_0, E_F) \quad (16)$$

Neste modelo, apenas as propriedades da amostra contribuem para a qualidade da imagem obtida pelo STM, tornando-o fácil de ser manipulado. Entretanto, a resolução atômica não pode ser explicada por este modelo.

Cálculos e experimentos mostram que existe normalmente um estado  $d_z^2$  perto da fronteira de Fermi, presente no átomo da ponta, que predominantemente contribui para a corrente de tunelamento. É conhecido o fato que este estado (e também o estado  $p_z$ ) é vantajoso para uma ponta de boa qualidade. Como o tunelamento é a sobreposição do estado da ponta e da amostra, existe uma simetria entre os dois. Trocando-se o estado eletrônico da ponta e da amostra, a imagem obtida deve ser a mesma (princípio da reciprocidade). Isso explica porque, muitas vezes, a amplitude da ondulação obtida em uma imagem produzida por um STM é maior que a densidade de estados local da amostra, pois a ponta do STM detecta, neste caso, uma superfície fictícia com uma distribuição de estados  $d_z^2$ .

O estado do átomo da ponta depende do material e de sua orientação. Como a ponta é difícil de ser manipulada, é um dos problemas mais difíceis do STM.

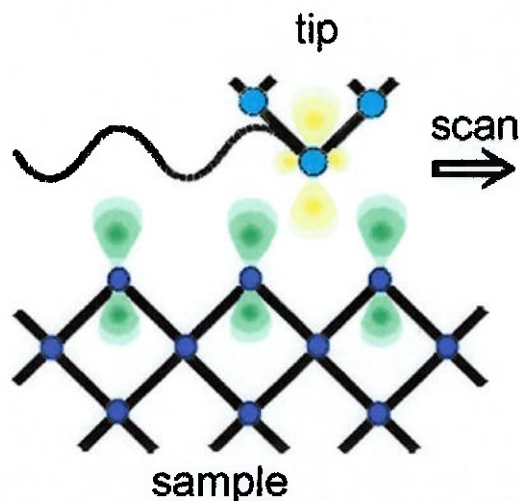


Figura 7 - Densidade de estados altamente ondulada

### 3.3 Modos de operação

#### 3.3.1 Modo de corrente constante

Usando-se um sistema de controle com feedback, a ponta é ajustada verticalmente para se manter a corrente constante. Como a corrente é proporcional à densidade de estados local, a ponta segue o contorno de uma densidade de estados constante durante a varredura. Gravando-se a posição vertical da ponta, obtemos um tipo de imagem topográfica da amostra.

#### 3.3.2 Modo de altura constante

Neste modo, a posição vertical da ponta não é alterada. A corrente, como função do posicionamento lateral representa a imagem da superfície. Este modo somente é propício para superfícies atômicamente planas, caso contrário uma colisão entre a ponta e a amostra seria inevitável. Uma grande vantagem deste modo é que pode ser utilizado em grandes frequências de varredura (até 10 kHz). Comparativamente, no modo da corrente constante, uma única imagem demora entre 1 segundo até alguns minutos para ser obtida.

#### 3.3.3 Imagem da altura da barreira

Até agora, apenas superfícies homogêneas foram consideradas. Se existe algum

composto não homogêneo na amostra, a função de trabalho também não será homogênea. Isto altera a altura da barreira de potencial local. Usando-se algum dos dois métodos acima, obteríamos uma imagem errada. Para resolver este problema, diferenciando-se a equação 8, temos:

$$\frac{d(\ln I)}{ds} \propto \sqrt{\Phi} \quad (17)$$

Portanto a função de trabalho pode ser medida diretamente mudando-se a distância entre a ponta e amostra, o que pode ser feito modulando-se a corrente com o feedback ligado.

### Espectroscópio de tunelamento eletrônico

Se o elemento de matriz e a densidade de estados na ponta forem constantes, a corrente de tunelamento da equação 15 pode ser alterada para:

$$I \propto \int_0^{eV} \rho_{sa}(E_F - eV + \varepsilon) d\varepsilon \quad (18)$$

A diferenciação resulta na densidade de estados:

$$\frac{dI}{dV} \propto \rho_{sa}(E_F - eV) \quad (19)$$

A densidade de estados pode ser deduzida por:

- Modulação da voltagem;
- Espectroscópio de Tunelamento Eletrônico Corrente – Imagem (CITS): A ponta realiza a varredura no modo de corrente constante, para se ter uma distância constante entre a ponta e a amostra. A cada ponto o feedback é desativado e uma curva de corrente - voltagem é gravada.

### 3.4 Aspectos técnicos

O STM tem os seguintes requisitos:

- Controle de movimento da ponta de alguns mm até 0,01Å;
- Posicionamento lateral exato;
- Distância entre amostra e ponta constante;
- Ponta fina;
- Medição da corrente de 0.01nA até 50nA.

### 3.4.1 Posicionamento

Como a ponta tem a necessidade de um ajuste de distância que varia de ordens maiores a muito menores são necessários dois sistemas de posicionamento: um grosseiro e um sistema de posicionamento mais refinado. O sistema de posicionamento refinado é utilizado na varredura. Este sistema é feito com material piezo-cristal ou piezo-cerâmica.

#### 3.4.1.1 Piezo-eletricidade

Primeiramente, vamos introduzir dois conceitos: Efeito Piezoelétrico é o efeito obtido quando se aplica uma tensão mecânica em um material piezo e obtém-se uma polarização elétrica. O Efeito Inverso Piezoelétrico é quando se aplica um campo elétrico ao material piezo e se obtém uma deformação mecânica. São exemplos de piezo-cristais o quartzo e  $\text{BaTiO}_3$ ; e Piezo-cerâmicas o  $\text{Pb}(\text{Ti,Zr})\text{O}_3$  (PZT).

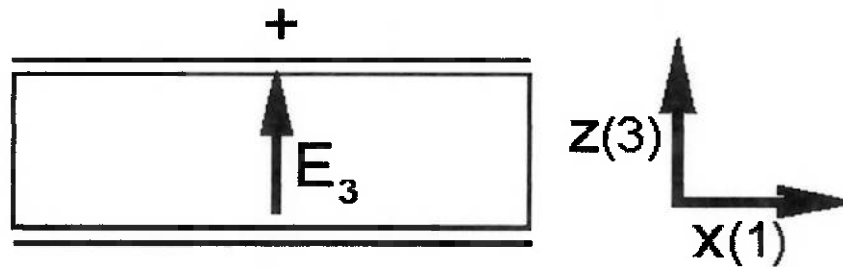


Figura 8 - Efeito piezo-elétrico

No equacionamento a seguir, as direções  $x$ ,  $y$  e  $z$  são referenciadas como 1, 2 e 3 respectivamente. Aplicando-se uma voltagem a um pedaço retangular de material piezo de espessura  $z$ , como mostrado na figura acima, resulta no seguinte campo elétrico:

$$E_3 = \frac{V}{z} \quad (20)$$

Uma deformação é então gerada, a qual pode ser descrita por um tensor com componentes  $-xx$  e  $-zz$ :

$$S_1 = \frac{\delta x}{x} \quad (21)$$

$$S_3 = \frac{\delta z}{z} \quad (22)$$



As constantes piezo-elétricas são definidas como a razão entre a componente de deformação sobre a componente do campo elétrico. Assim, temos:

$$d_{31} = \frac{S_1}{E_3} \quad (23)$$

$$d_{33} = \frac{S_3}{E_3} \quad (24)$$

Valores típicos para componentes piezo-elétricos PZT são  $d_{31} = -1$  a  $-3 \text{ \AA/V}$  e  $d_{33} = 2$  a  $6 \text{ \AA/V}$ .

#### 3.4.1.2 Posicionador refinado

- Tripé: A ponta é controla por três barras ortogonais de material piezo.

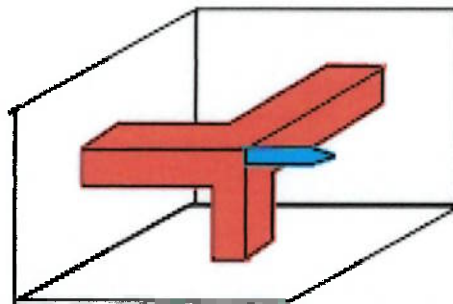


Figura 9 - Sistema de posicionamento em tripé

- Bimorfo: Um bimorfo é feito de duas placas finas de piezo-cristais que são empilhados de forma que a voltagem aplicada nas placas causam tensão mecânica de sinais opostos, o que gera uma deflexão. Combinando-se diversos bimorfos, pode-se criar um sistema tridimensional de varredura.

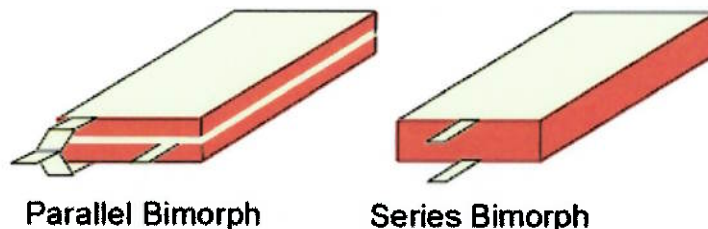


Figura 10 - Sistema de posicionamento bimorfo

- Tubo: Um cilindro, feito de piezo-cerâmicas, é coberto dentro e fora com metal, que

agem como eletrodos. O eletrodo externo é dividido em quatro setores. O movimento na direção z (na direção do eixo) pode ser realizado aplicando-se tensão elétrica entre os eletrodos interno e externo. Uma deflexão na direção xy pode ser obtida aplicando-se voltagem entre dois eletrodos situados em lados opostos.

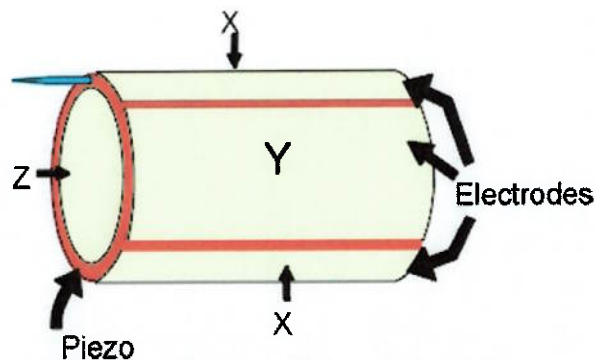


Figura 11 - Sistema de posicionamento tubo

### 3.4.2 Circuito eletrônico

No caso da medição da corrente no modo de corrente constante, um circuito de feedback tem que ser construído para o controle do piezo da direção z.

A corrente de tunelamento (0.01mA-50mA) é convertida em uma voltagem por um amplificador de corrente. Para ter uma resposta linear no que se refere à distância de tunelamento (a corrente é exponencialmente dependente da distância entre a amostra e a ponta) o sinal é processado por um amplificador logarítmico. A saída deste amplificador é comparada com a voltagem pré-determinada, a qual é usada como uma referência para a corrente. O sinal de erro é então enviado ao feedback, que aplica uma voltagem ao piezo z para manter a diferença entre a corrente de referência e a de tunelamento pequena. Cuidados devem ser tomados para se manter o ruído do sinal em baixos níveis. Também, o tempo de resposta do feedback tem que ser minimizado sem se perder acurácia.

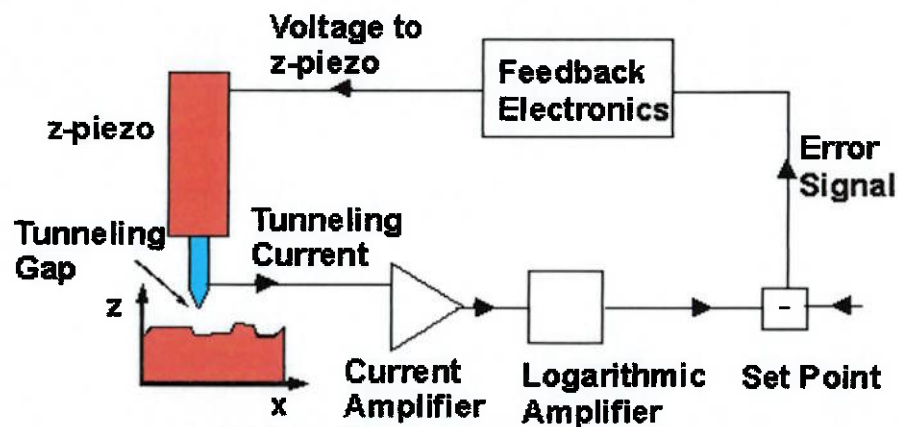


Figura 12 - Esquema do circuito eletrônico

### 3.4.3 Isolamento de vibração

A distância entre a amostra e a ponta deve sempre ser mantida constante com  $0,01\text{\AA}$  para se ter uma boa resolução atômica. Portanto é absolutamente necessário reduzir vibrações internas e isolar o sistema de vibrações externas.

Vibrações do meio são causados por:

- Vibração do edifício 15-20 Hz;
- Pessoas correndo 2-4 Hz;
- Bombas a vácuo;
- Sons.

No que se diz respeito a função de transferência (razão de amplitudes) de um oscilador harmônico amortecido excitado com uma força externa senoidal, é claro que a frequência de ressonância deve ser muito menor que a frequência externa a ser eliminada.

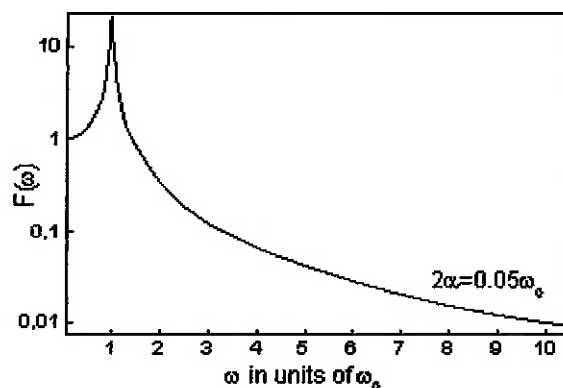


Gráfico 1 - Amplitude de vibração do STM

O amortecimento pode ser feito por:

- Suspensão do STM com o auxílio de molas;
- Sistema de pratos empilhados;
- Sistema pneumático.

#### 3.4.4 Ponta

A ponta é a parte mais complicada de um STM. Ela precisa ter pequena curvatura para poder capturar sinais de áreas pequenas.

Para uma resolução atômica, é necessário que se tenha uma ponta minúscula, de um átomo apenas, como pode ser observado na figura abaixo.

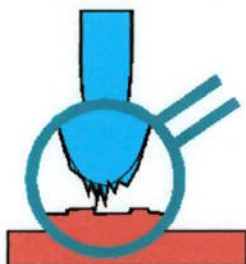


Figura 14 - Ponta de um átomo

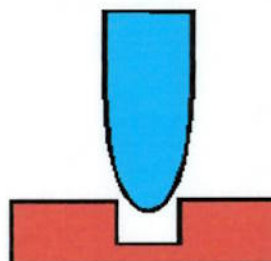


Figura 13 - Ponta de pequena curvatura

Pontas geralmente são feitas de tungstênio, platina ou um fio de Pt-Ir. Uma ponta fina pode ser produzida por:

- Corte e afiação;

Fios de Pt-Ir cortados em um ângulo de 20-30° produzem uma ponta com a curvatura necessária.

- Desgaste eletroquímico.

No desgaste eletroquímico, um fio de tungstênio é mergulhado parcialmente em uma solução de NaOH e ligada ao pólo positivo de uma fonte de tensão, enquanto o pólo negativo é ligado a um fio de cobre mergulhado na solução. O desgaste ocorre predominantemente na região do fio que se encontra na superfície da solução. Quando o pescoço formado pelo desgaste é fino o suficiente, o fio quebra devido ao seu próprio peso. Então, duas pontas são formadas. Elas têm que ser limpas com água deionizada e etanol ou metanol puros.

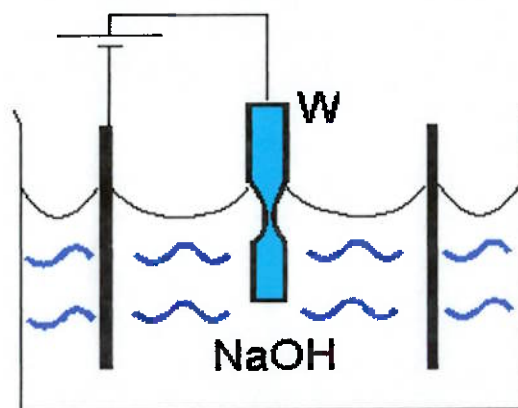


Figura 15 - Desgaste eletroquímico da ponta

É comum que, no final deste processo, a ponta esteja oxidada ou impregnada com impurezas devido à solução. Neste caso, outros tratamentos devem ser dados à ponta para que ela possa ser utilizada.

Para obter pontas afiadas ao final do processo, deve-se interromper o processo químico o mais rápido possível depois do rompimento do fio. Para isso, pode-se fazer um circuito comparador simples, que interrompe o fornecimento de tensão quando um aumento brusco da resistência é detectado.

Também é possível fazer a ponta durante o tunelamento:

- Aumento súbito da voltagem para aproximadamente -7V (na amostra) por 2-4 linhas de varredura:

Por este tratamento, alguns átomos de tungstênio podem sair da ponta, devido ao campo elétrico não-uniforme, e formar uma nano-ponta.

- Colisão controlada em uma superfície de Si.

A ponta pode encontrar um cluster de Si, que forma um apêndice monoatômico com uma ligação do tipo  $p_z$ .

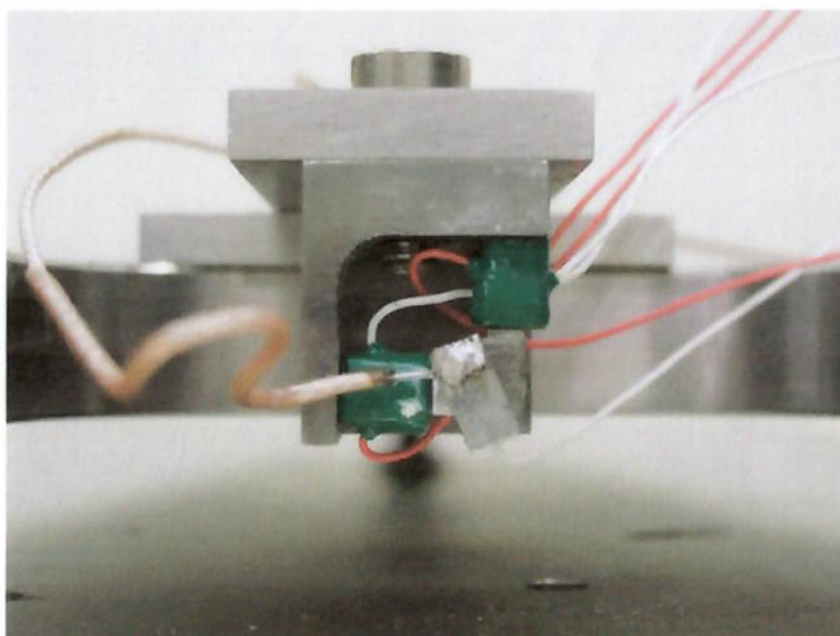
## 4. O microscópio de tunelamento eletrônico

Com base nestes fundamentos, uma equipe da Universidade de Muenster projetou um microscópio de tunelamento eletrônico de baixo custo, controlado através de um PC com uma placa AD/DA e um software específico. Este software também permite a visualização das medições na tela do computador.

Para uma análise mais aprofundada do funcionamento deste STM, é conveniente dividi-la em três partes: componentes mecânicos, componentes eletrônicos e software.

#### 4.1 Componentes mecânicos

Como a ponta do microscópio deve ser capaz de movimentar-se em três dimensões em incrementos da ordem de décimos de nanômetros, é utilizada uma estrutura que comporta 3 piezos, de forma que a variação de tensão em um deles faça com que a ponta se desloque em um dos eixos. Esta estrutura é mostrada abaixo, já com os piezos instalados:



**Figura 16 – Estrutura com os piezos instalados**

O piezoelemento utilizado apresenta um deslocamento de  $1\mu\text{m}$  para uma diferença de potencial de 30V, medida que será verificada mais adiante.

O corpo do microscópio é dividido em duas partes: superior e inferior. Na parte superior situam-se o suporte dos piezos e 3 parafusos micrométricos. Estes parafusos permitem uma regulação grosseira do posicionamento da ponta do microscópio. O parafuso traseiro, por situar-se mais distante dos outros, torna possível uma regulação mais precisa do posicionamento da ponta, devido à configuração de alavanca. Na parte inferior encontram-se reentrâncias para auxiliar no posicionamento da parte superior, de forma a permitir a varredura de uma porção maior da amostra, assim como facilitar a troca de amostras.





**Figura 17 - Corpo do microscópio**

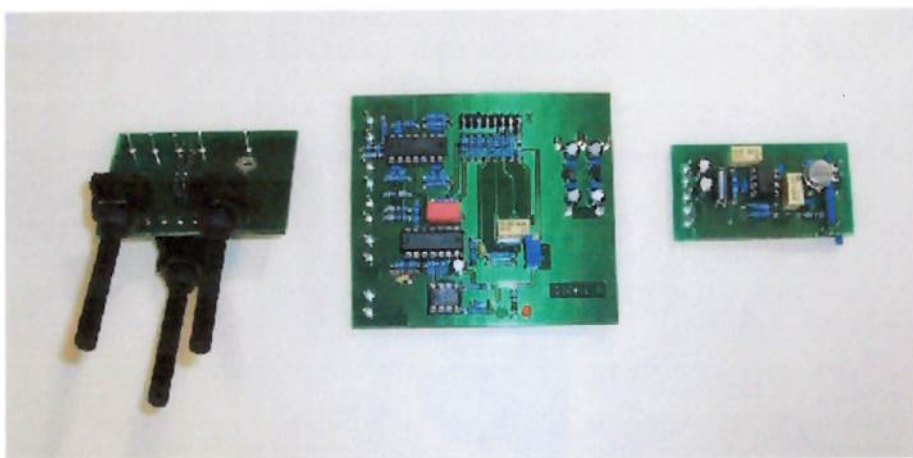
## **4.2 Componentes eletrônicos**

A parte eletrônica do microscópio é responsável por fornecer a tensão adequada aos diversos componentes do microscópio (especialmente aos piezoelementos), amplificar o sinal de corrente proveniente da ponta e responder adequadamente às alterações feitas tanto através software quanto pela caixa de controle. Para atender a estes requisitos, a parte eletrônica do microscópio foi dividida em cinco módulos básicos, cada um com uma função específica. Os esquemas dos circuitos encontram-se no Anexo 1.

- **Fonte:** realiza a suavização da tensão a ser fornecida, necessária quando a tensão é fornecida por um equipamento ligado à corrente alternada.
- **Fornecimento de tensão para os piezos x-y:** este módulo possui dois componentes básicos, sendo que existem duas unidades de cada componente, uma para cada piezo. U1A e U1C (ver Anexo 1) desassociam o sinal proveniente do computador dos piezos, utilizando-se da alta impedância de entrada dos amplificadores operacionais. U1B e U1D fazem com que a voltagem fornecida aos piezos seja simétrica, através da inversão do sinal original. O sinal original é fornecido a um dos eletrodos do piezo, enquanto o outro é ligado ao sinal invertido. Os capacitores de 10nF servem para ajustar os amplificadores operacionais à impedância capacitiva na entrada do mesmo, que ocorre devido aos piezoelementos.
- **“Bias” da amostra:** Fornece a tensão entre a amostra e a ponta do microscópio de acordo com a regulação realizada na caixa de controle.

- **Controle do eixo z:** A corrente de tunelamento é recebida por U3A, que filtra os ruídos de alta frequência. A corrente é então amplificada por meio de um ganho ajustável, e um offset pode ser colocado. Para manter a corrente de tunelamento constante, utiliza-se um controlador integral, que provoca uma mudança na posição da ponta de acordo com as alterações na corrente. O set-point pode ser ajustado na caixa de controle, e a diferença deste em relação ao valor real é integrado por U3C. Este sinal é enviado de forma simétrica para o piezo do eixo z, da mesma maneira que é feito nos piezos dos eixos x e y. A constante do controlador integral, que altera a velocidade de reação do controlador às mudanças na corrente, também pode ser ajustada pela caixa de controle.
- **Caixa de controle:** A caixa de controle permite a regulação da tensão entre a ponta e a amostra, do set-point da corrente de tunelamento e do ganho do controlador integral.
- **Pré-amplificador:** O Pré-amplificador transforma a corrente de tunelamento em uma tensão e então realiza uma amplificação de  $10^8$ . Esta transformação tem por objetivo evitar que o sinal sofra a interferência de ruídos externos. Dessa forma, o pré-amplificador deve situar-se o mais próximo possível da ponta, de forma a minimizar a exposição do sinal a ruídos antes de sua amplificação.

A implementação destes módulos foi feita através de 3 placas de circuito impresso. Em uma das placas encontra-se a caixa de controle, em outra o pré-amplificador e na última, a placa principal, encontram-se componentes que constituem os módulos restantes.



**Figura 18 – Placas do STM: Controle (esquerda), Principal (centro) e Pré-amplificador (direita)**



### **4.3 Software**

O software é responsável pelo controle da parte variável da eletrônica e pela visualização dos resultados da medição na tela. Dessa forma, é necessária uma comunicação bidirecional com a eletrônica do microscópio. Isto é feito com o uso de uma placa de aquisição AD/DA. A posição, tamanho e velocidade da varredura são definidos no software, transformadas em um sinal analógico de tensão e enviadas à placa principal do microscópio; já a corrente de tunelamento e o sinal de controle do piezo do eixo z são enviados à placa AD/DA e transformados em um sinal digital para interpretação pelo software.

O software foi escrito em Visual Basic e seu código-fonte encontra-se disponível para visualização no site do projeto. O manual do software encontra-se no Anexo 3.

## **5. Montagem do experimento**

Após o estudo dos fundamentos teóricos e do funcionamento do STM, realizou-se a montagem do experimento. Isto inclui a montagem das placas do microscópio, montagem de uma mesa inercial para isolar o microscópio de vibrações indesejadas, adaptação do software original para a utilização de uma outra placa AD/DA, verificação do comportamento dos piezos e as conexões finais.

### **5.1 Montagem das placas**

Como as placas de circuito impresso encontravam-se unidas e ainda sem os componentes instalados, foi necessário realizar o corte das placas e a soldagem dos componentes. Para a correta montagem das placas, utilizou-se como referência o layout da placa que se encontra em Anexo 2.

### **5.2 Montagem da mesa inercial**

Para um correto funcionamento do microscópio, é necessário realizar um amortecimento adequado das vibrações, especialmente aquelas com frequência por volta de

10Hz, causadas, por exemplo, pelo andar de pessoas. Assim, foi feita a instalação de uma mesa inercial, constituída de uma base pesada suspensa por três elásticos de 1m de comprimento, fixados no teto.



**Figura 19 – Mesa inercial**

### **5.3 Verificação do piezoelemento**

Para verificar se o comportamento do piezoelemento estava de acordo com o esperado, foi realizado o levantamento de uma curva de deslocamento do piezo em função da tensão aplicada. Utilizou-se para isso um interferômetro de Michelson fabricado pela MEOS. Para maiores detalhes sobre seu funcionamento, o manual do experimento encontra-se no website do fabricante.

#### **5.3.1 Procedimento experimental**

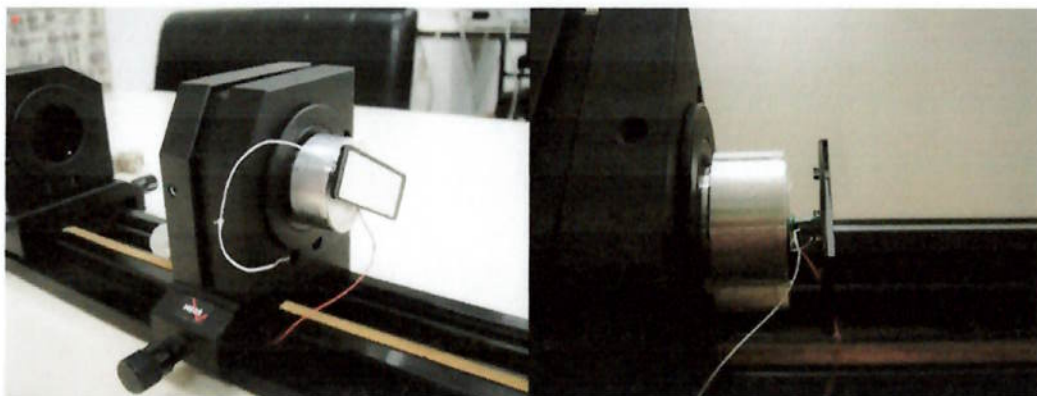
Tendo como base a montagem original do interferômetro de Michelson, foram realizadas pequenas adaptações para permitir a instalação do piezo de forma a obter uma medição mais confiável. A idéia principal foi utilizar o suporte oferecido pelo equipamento, que permite uma regulação precisa da inclinação tanto no plano vertical quanto no horizontal,

facilitando os procedimentos de alinhamento do experimento. Assim, foi projetada e fabricada uma peça que se encaixasse nesse suporte e permitisse a fixação do piezo. A figura abaixo mostra o suporte que foi fabricado:



**Figura 20 - Suporte fabricado**

Nesse suporte foram fixados, utilizando-se cola quente, o piezo e um espelho. A montagem é mostrada nas figuras a seguir:



**Figura 21 – Montagem do piezoelemento no suporte**

O piezo é ligado a uma fonte de tensão variável de 30V. A montagem final da experiência é mostrada abaixo:



**Figura 22 – Interferômetro de Michelson montado para a experiência**

Após a montagem, os procedimentos de alinhamento e a verificação do funcionamento do experimento, iniciou-se o processo de coleta de dados. Variou-se a tensão aplicada de 5 em 5V, indo de 0V a 30V e depois de 30V a 0V. A cada variação de tensão, foi tirada uma foto do padrão de interferência, com a câmera sempre na mesma posição. Posteriormente foi feita a análise das fotos e a confecção da curva de calibração.

### 5.3.2 Resultados

A seguir são apresentadas as fotos que serviram de base para a análise:



**Figura 23 - Seqüência de 0 a 30V (movimentação das franjas “para cima”)**



**Figura 24 - Seqüência de 30 a 0V (movimentação das franjas “para baixo”)**

Com base nas fotografias das franjas de interferência, foi possível medir o

deslocamento das franjas a cada variação de tensão, e conhecendo-se o comprimento de onda do feixe laser utilizado (633nm), construiu-se o seguinte gráfico de tensão x deformação do piezoelemento.

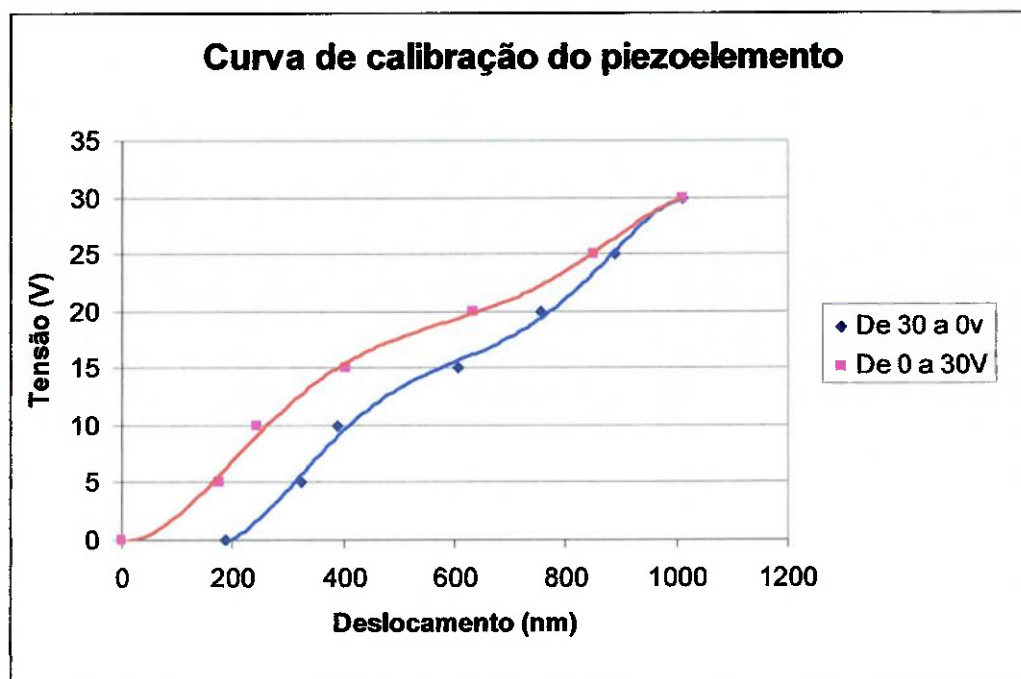


Gráfico 2 – Verificação do piezoelemento

Observando o gráfico, nota-se que existe uma certa diferença entre a curva de 0 a 30V em relação à curva de 30 a 0V. Esperava-se que os pontos iniciais e finais das duas curvas coincidissem, o que não aconteceu. Isso pode ter ocorrido devido à propagação de erros causada pela natureza incremental da medição e também pela vibração da projeção das franjas, que torna as fotos um pouco imprecisas. Os diferentes “caminhos” percorridos por cada curva eram esperados, pois é uma característica dos piezoelementos.

Nota-se pelo gráfico também uma certa não-linearidade no deslocamento, presente de forma similar nas duas curvas.

### 5.3.3 Discussão dos resultados

Apesar da ordem de grandeza do erro do experimento ser muito grande se comparada com a precisão do microscópio, foi possível constatar que o piezo testado apresenta comportamento adequado para uso no STM, com uma deformação de 1µm para uma tensão

de 30V. Poderiam ser utilizadas montagens mais complexas do interferômetro para obter uma maior precisão, entretanto estas apresentam uma dificuldade maior de adaptação para a medição do deslocamento do piezo, e o aumento da precisão não mudaria as conclusões a respeito das medições, tendo em vista o objetivo inicial do experimento.

## 6. Software

### 6.1 Adaptação do software

O microscópio de tunelamento eletrônico sobre o qual foi feito este trabalho possui um software, escrito em Visual Basic 6.0, que permite a visualização de imagens em duas dimensões. Os dados utilizados pelo software para confecção desta imagem (tensão aplicada no piezoelemento de controle da altura) não podem, entretanto, ser utilizados por outros softwares ou serem utilizados para algum tipo de análise.

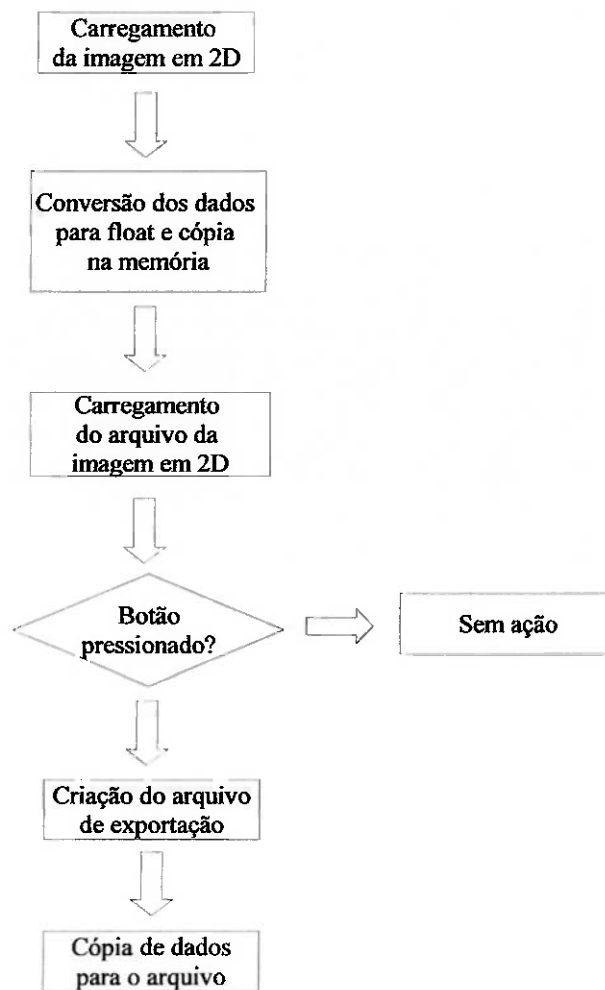
Estes dados não podem ser utilizados porque o programa original utiliza uma estrutura onde os valores da voltagem aplicada no piezoelemento são armazenados em uma matriz. Para permitir a visualização posterior da imagem, estes valores são armazenados em um arquivo na forma binária.



Figura 16 - Diagrama em blocos do carregamento da imagem

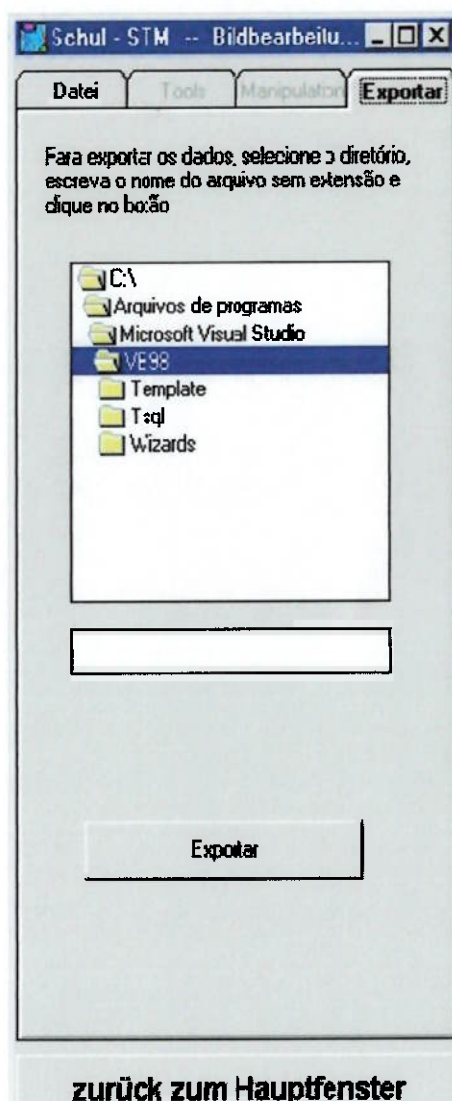
Para que estes valores possam ser armazenados em um arquivo de formato compatível com outros programas, exportamos os valores para um arquivo de formato ASCII. Isto foi implementado convertendo-se os valores de linha (x), coluna (y) e altura (z) em uma string com os valores separados por um espaço, colocando-se um conjunto de valores relacionados

(x, y e z) por linha.



**Figura 17 - Diagrama de blocos da exportação dos dados**

Como não é uma boa prática realizar a exportação em tempo real, isto é feito através de uma ação do usuário. Foi adicionada à interface gráfica do software do microscópio uma tela de exportação. Nela, pode-se escolher o diretório onde o arquivo será armazenado e seu nome. Para executar a exportação, o usuário deverá pressionar o botão da tela. Ao pressionar o botão, o software irá extrair os dados da imagem carregada na memória e irá copiá-los para o arquivo de exportação, utilizando o procedimento já descrito.



**Figura 18 - Tela criada para exportação**

Abaixo segue um trecho de um arquivo de saída exportado:

```

124 1 -6,347656E-02
125 1 7,797852
126 1 9,995117
127 1 9,995117
128 1 9,995117
1 2 9,995117
2 2 9,995117
3 2 9,995117
4 2 9,995117
5 2 5,532227
6 2 9,995117

```

Nota-se que os valores de x e de y variam de 1 em 1. Os valores reais da unidade em x



e y podem ser obtidas através da resolução da captura da imagem, que é configurada pelo usuário antes da captura.

## 6.2 Software de visualização de imagem 3D

Ao realizar a exportação do arquivo para um formato compatível com outros softwares, é possível a visualização de uma imagem 3D da superfície analisada. Junto com o arquivo ASCII descrito na seção anterior, é criado um outro arquivo, também de formato ASCII, mas com os dados dispostos de forma diferente e com o nome fixo (grafico3D.grf).

O arquivo grafico3D.grf tem apenas os valores de z (um por linha), sendo que os valores de x e y já são conhecidos devido à sua natureza sequencial. O software de visualização lê, primeiramente, o número de linhas e, a partir desta informação, já é conhecida a resolução da imagem em número de linhas. Com a resolução da imagem, o programa lê linha por linha do arquivo, fazendo variação em x, primeiramente, de 1 até a resolução máxima e, em seguida, variando y (uma estrutura semelhante a um laço for dentro de outro laço for) até sua resolução máxima. Para cada combinação de x e y, há um valor de z associado.

Ao iniciar o software de visualização de imagem 3D, estes dados são carregados automaticamente e o gráfico resultante é mostrado na janela principal. Este software foi escrito em Visual C++ 6.0 utilizando-se bibliotecas OpenGL.

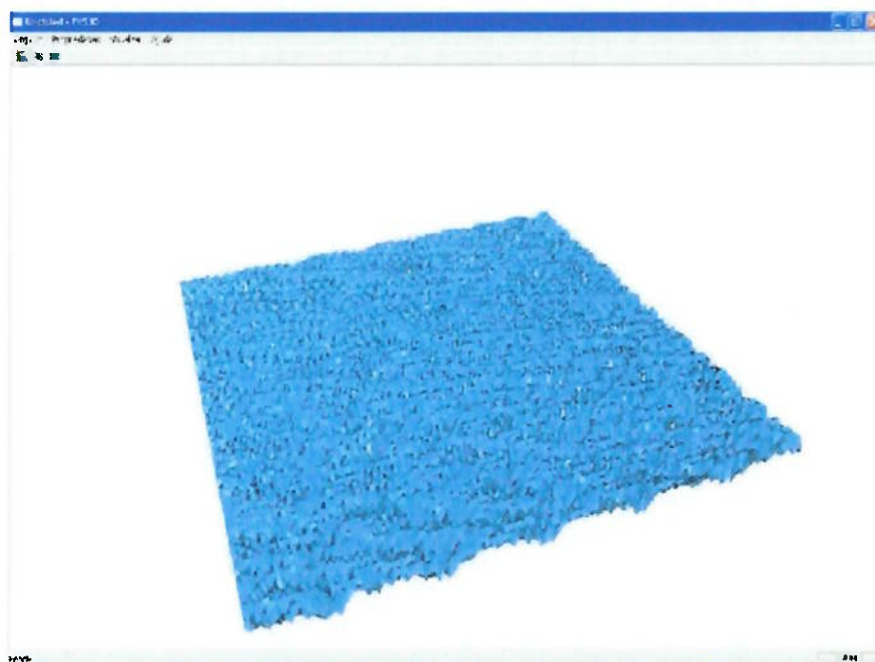
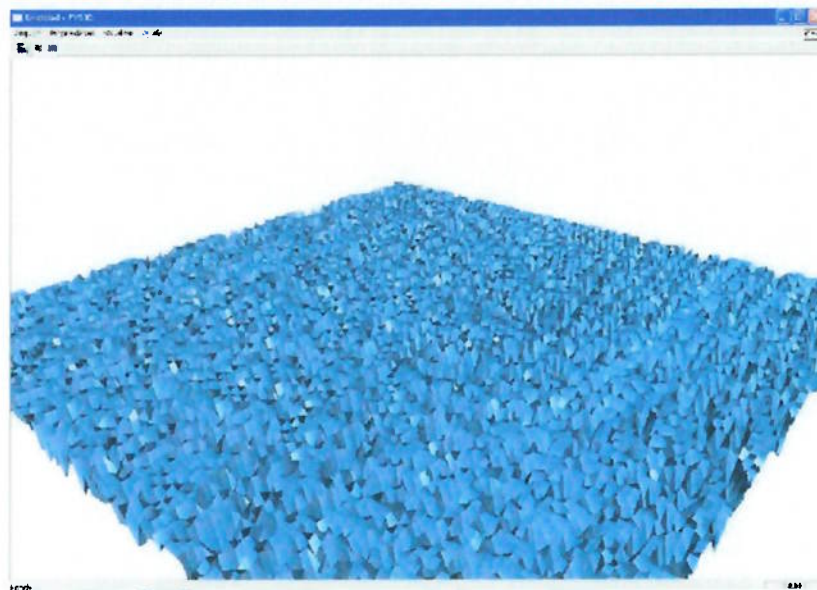


Figura 19 - Tela principal com imagem de superfície de grafite carregada

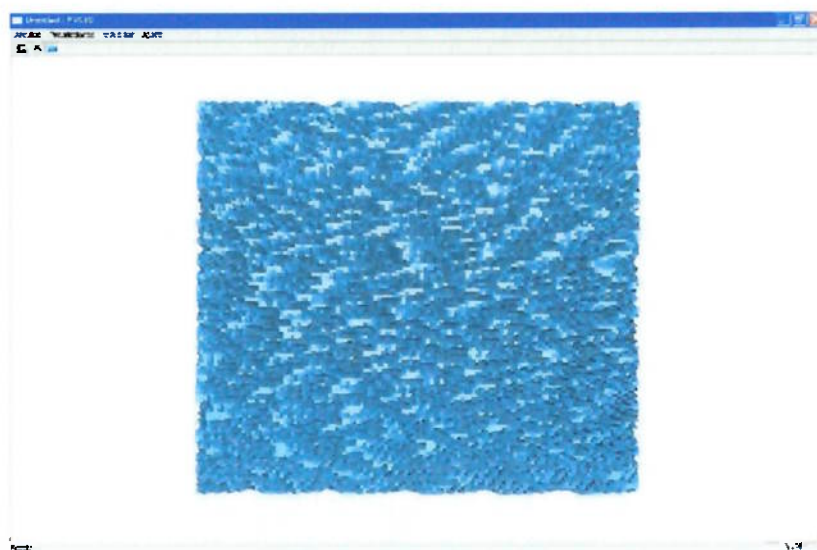
A imagem observada no programa pode ser modificada para melhorar a sua visualização. Entre as possibilidades de modificação, se encontram:

- Rotação do gráfico;
- Zoom;
- Modificação da cor de fundo;
- Dois modos de renderização diferentes (wireframe ou superfície).

Aplicando essas modificações, é possível obter ângulos diferentes de visualização e eliminar duplas interpretações visuais do gráfico.



**Figura 20 - Demonstração de zoom da imagem**



**Figura 21 - Demonstração das capacidades de rotação**

A estrutura atômica da superfície tem uma característica geométrica geralmente conhecida. Entretanto, os valores de altura da superfície não estão ajustados neste software. Para fazê-lo, é necessário modificar um fator de divisão inserido no cálculo da altura  $z$ . Este fator está indicado no código fonte do programa em um comentário. É necessário notar que para cada amostra, esse fator de divisão terá um valor diferente.

## **7. Considerações finais**

Comparando as imagens obtidas através do software de visualização 3D com aquelas obtidas pelo software do microscópio, podemos observar que as duas são originadas de um mesmo conjunto de dados. Entretanto, a qualidade das imagens obtidas pelo microscópio ainda não está adequada para observar com clareza a geometria da superfície do material analisado.

Dessa forma, a melhora da qualidade da imagem obtida no software de visualização 3D depende diretamente de ajustes no microscópio e do fator de divisão da altura.

## Referências

FRIEDRICH ALEXANDER UNIVERSITÄT ERLANGEN NÜRNBERG. Scanning Tunneling Microscopy (STM). Disponível em <<http://www.fkp.uni-erlangen.de/methoden/stmtutor/stmpage.html>>.

TANIGUCHI, N. Nanotechnology: integrated processing systems for ultra-precision and ultra-fine products. Oxford University Press, 1996.

COHEN, S.H., BRAY, M.T., LIGHTBODY, M.L. Atomic force microscopy/scanning tunneling microscopy. New York : Plenum Press, 1994. 453 p.

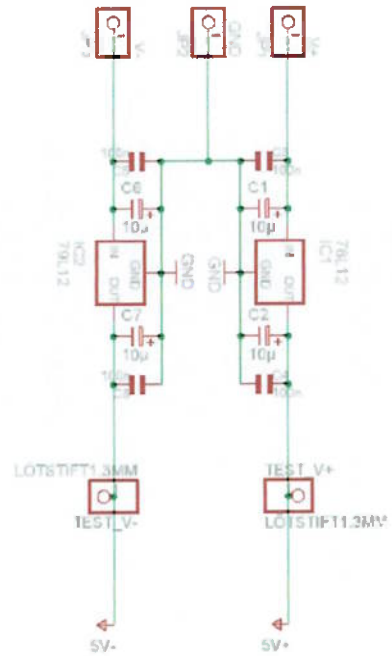
SPHAR, C. Learn Microsoft Visual C++ Now. Redmond, WA : Microsoft Press, 1999. 637 p.

FULLER, J. (Coord.). The Official Guide to Learning OpenGL, Version 1.1. A-W Developers Press, 1997. 616 p.

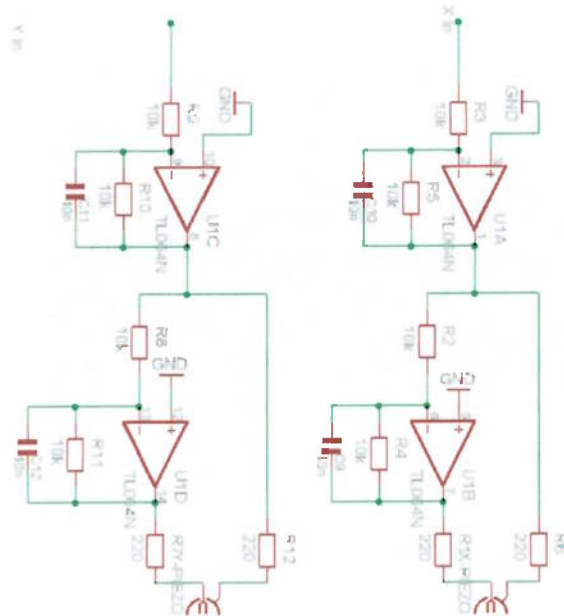
## **Anexo 1**

### **Diagrama dos circuitos**

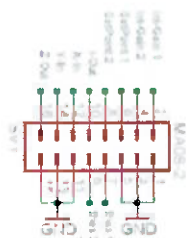
## Power Supply



## Supply for Scan-Piezos



## Connector To Controlbox



## Holes



## Steuerplatine

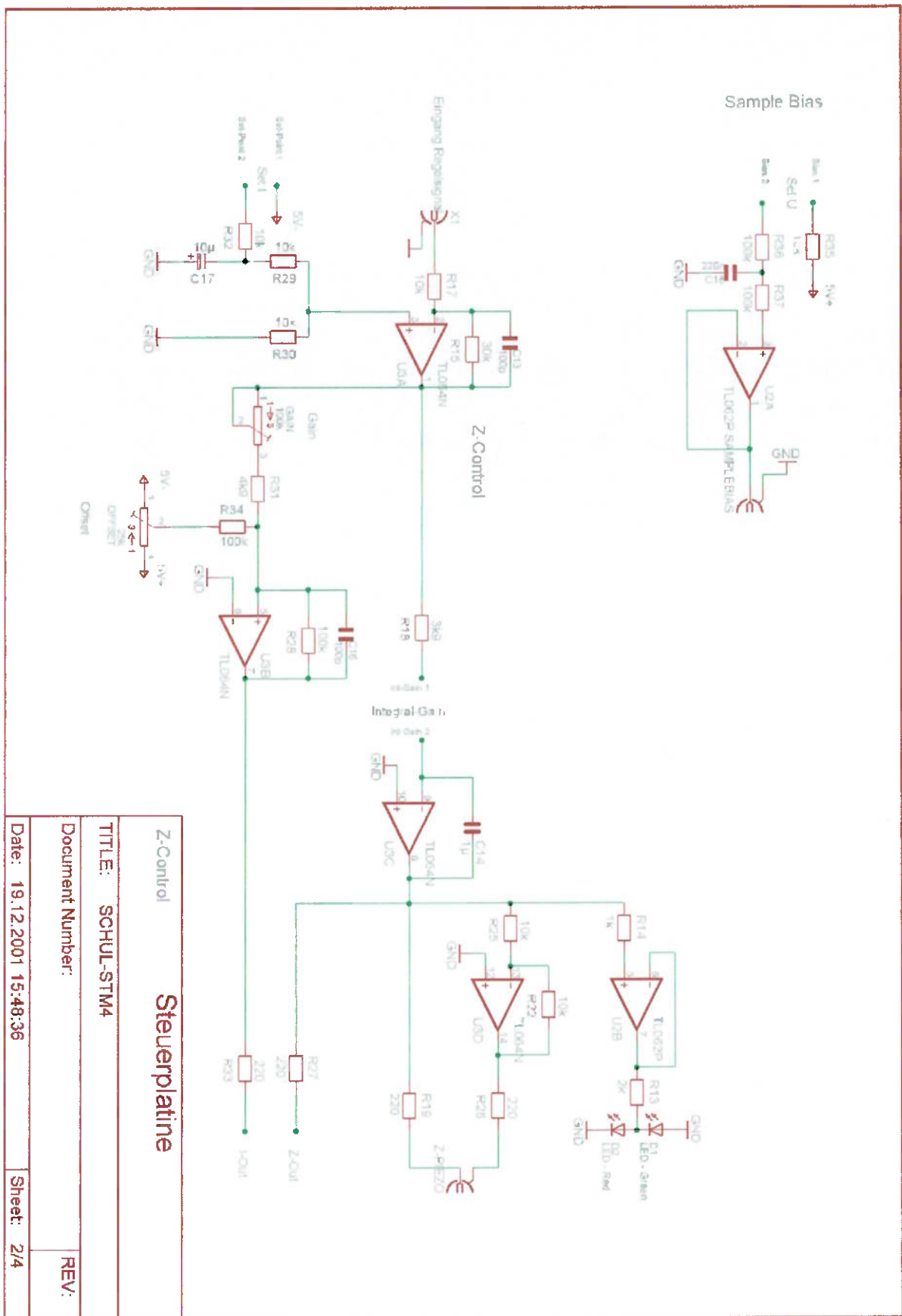
TITLE: SCHUL-STM4

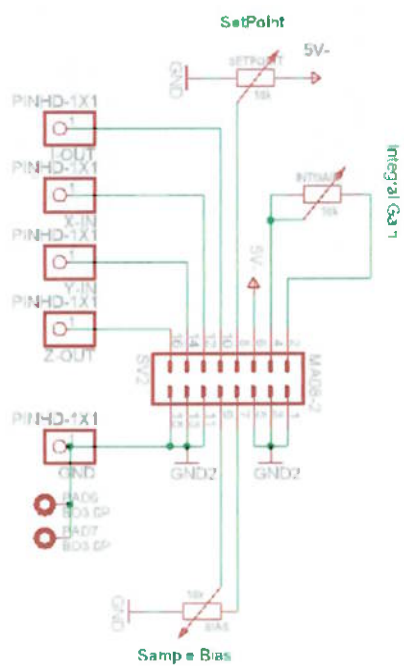
Document Number:

REV:

Date: 19.12.2001 15:48:36

Sheet: 1/4





# Control-Box

TITLE: SCHUL-STM4

Document Number:

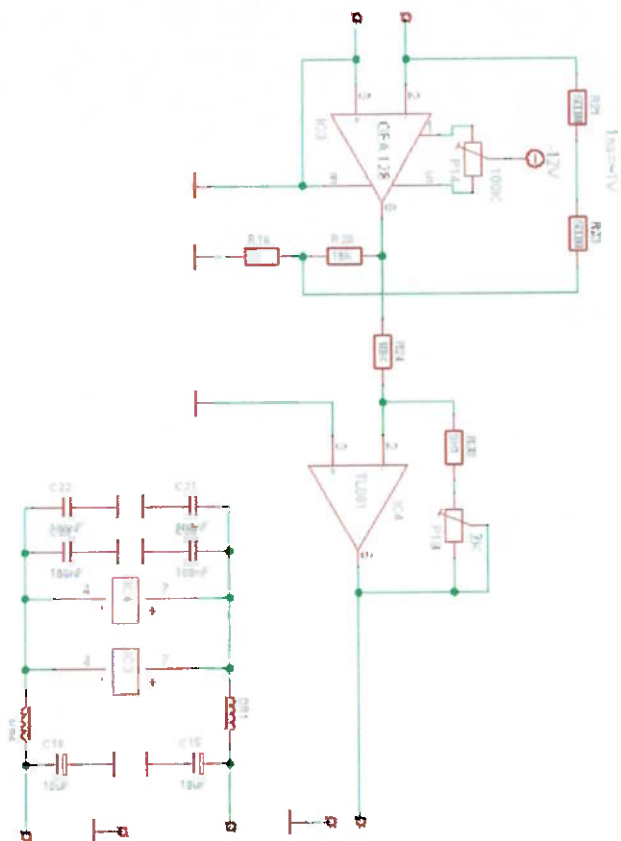
Date: 19.12.2001 15:48:36

REV:

Sheet: 3/4



## Pre-Amplifier



## Tunnel-Verstärker

TITLE: SCHUL-STM6

Document Number:

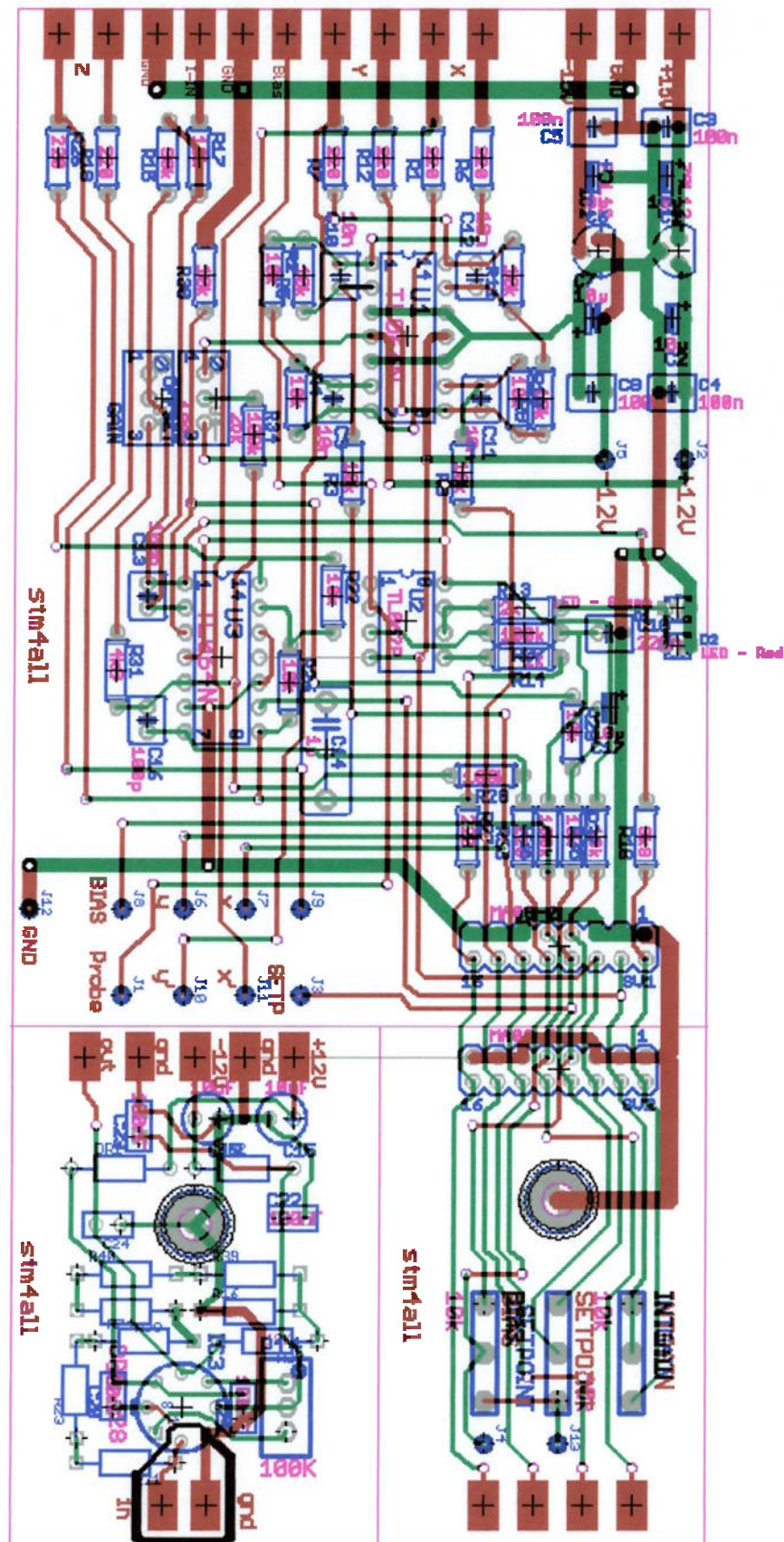
Date: 24.07.2003 17:37:34

REV:

Sheet: 4/4

## **Anexo 2**

### **Layout das placas**



## **Anexo 3**

### **Documentação do Software**

# STM

Documentação do Software

Escrito em Visual Basic® 6.0 pelo Bacharel em  
Física Marcus Schäfer

1.	Informações sobre o Hardware	5
1.1.	Computador	5
1.2.	Placa de aquisição	5
2.	Arquivos do Código Fonte	7
2.1.	Arquivo das classes	7
2.2.	Forms	7
2.3.	Módulos	7
2.4.	Usercontrols	8
2.5.	Arquivos do projeto	8
2.6.	Arquivos de figura	8
3.	Hierarquia de classes	9
3.1.	Duas considerações iniciais	9
3.2.	Representação Gráfica	10
3.3.	CADInfo	10
3.3.1.	Eventos	10
3.3.2.	Propriedades	10
3.3.3.	Public	11
3.3.4.	Private	11
3.4.	CEdit_STM_Image	12
3.4.1.	Eventos	12
3.4.2.	Propriedades	13
3.4.3.	Public	14
3.4.4.	Private	16
3.5.	CME300	17
3.5.1.	Eventos	17
3.5.2.	Propriedades	17
3.5.3.	Public	18
3.5.4.	Private	21
3.6.	CPC20TR	22
3.6.1.	Eventos	22
3.6.2.	Propriedades	22

3.6.3.	Public	23
3.6.4.	Private	26
3.7.	CRandom	26
3.7.1.	Eventos	27
3.7.2.	Propiedades	27
3.7.3.	Public	28
3.7.4.	Private	30
3.8.	CScale	30
3.8.1.	Eventos	30
3.8.2.	Propiedades	30
3.8.3.	Public	31
3.8.4.	Private	32
3.9.	CScan_STM_Image	32
3.9.1.	Eventos	32
3.9.2.	Propiedades	33
3.9.3.	Public	34
3.9.4.	Private	35
3.10.	CSTM_Data	36
3.10.1.	Eventos	36
3.10.2.	Propiedades	36
3.10.3.	Public	37
3.10.4.	Private	38
3.11.	CSTMTip	38
3.11.1.	Eventos	38
3.11.2.	Propiedades	39
3.11.3.	Public	40
3.11.4.	Private	41
3.12.	CstopUhr	41
3.12.1.	Eventos	41
3.12.2.	Propiedades	41
3.12.3.	Public	42

3.12.4.	Private	42
3.13.	CLineScan	42
3.13.1.	Eventos	42
3.13.2.	Propiedades	42
3.13.3.	Public	42
3.13.4.	Private	42
3.14.	CSTM_Image	43
3.14.1.	Eventos	43
3.14.2.	Propiedades	43
3.14.3.	Public	43
3.14.4.	Private	44
4.	Forms	44
4.1.	CSTM_Image.frm	44
4.2.	ClineScan.frm	45
4.3.	frm_coarse_Approach.frm	46
4.4.	frm_EditImages.frm	46
4.5.	frm_ScanFrame.frm	47
4.6.	frm_StartUp.frm	47
4.7.	frmAbout.frm	48
4.8.	Main.frm	49
5.	Módulos	49
5.1.	API_Interface	49
5.2.	Constantes	50
5.3.	Módulo de Matemática	50



## **1. Informações sobre o Hardware**

### **1.1. Computador**

Qual é o sistema que precisaremos para utilizar esse software? A resposta para esta questão é, na verdade, bem simples e barata. Como o software foi escrito em Microsoft Visual Basic, o equipamento não necessita ser muito avançado. O processador mínimo recomendado aqui é, portanto, um Pentium II 300 MHz.

Uma segunda necessidade, é um slot da placa mãe do tipo ISA livre. Na época que o STM foi projetado, não existia no mercado placas de aquisição para slots PCI baratas o suficiente. O slot do tipo ISA será usado para a placa de aquisição de dados. Como os computadores mais atuais não possuem esse slot de placa mãe do tipo ISA, é necessário a utilização de um computador mais antigo para poder executar este software com sucesso.

Portanto, o requisito mínimo de sistema é o seguinte:

- Processador: Pentium II 300MHz;
- 1 slot do tipo ISA livre;
- Sistema operacional: Microsoft Windows 95/98/2000/NT4.0;
- Memória RAM: 64MB;
- Espaço em disco: 10MB livres;
- Para o Microsoft Visual Basic 6.0 cerca de 100MB a mais;
- Monitor de 15"
- Resolução: 1024 × 768 Pixel

### **1.2. Placa de aquisição**

Este software foi especialmente escrito e testado para ser utilizado em dois modelos de placas que podem ser facilmente encontradas no mercado. São elas:

- A ME300 da *Meilhaus Electronic GmbH*
- A PC20TR da *BMC-Messsysteme GmbH*

As duas placas podem ser adquiridas diretamente das fabricantes e a placa PC20TR pode também ser adquirida na empresa Conrad Electronic.

Basicamente, o driver da placa a ser utilizada deve estar atualizado. Isto pode ser feito adquirindo-se a última versão do driver através de download da página na internet do fabricante. Também é possível utilizar o driver que acompanha a placa, se esta foi adquirida recentemente.

A placa mais barata na época que o software foi escrito (agosto de 2001) era a PC20TR. Outras placas de slot tipo ISA estão com um preço muito semelhante ao desta. As placas de slot PCI eram na época três vezes mais caras, pelo menos. Caso se tenha dinheiro disponível para comprá-las, deve-se pensar que, em um futuro próximo, placa-mãe com slots do tipo ISA serão muito difíceis de ser encontradas.

No mercado, também temos disponível sistemas de aquisição de dados que são compatíveis com portas USB. Abaixo, encontramos uma tabela com algumas informações referentes às placas utilizadas na confecção deste software:

	Mínimo	PC20TR	ME-300
Entrada AD	1	16	16
Limite	$\pm 3V$	$\pm 10V$	$\pm 10V$
Resolução	12 Bit	12 Bit	12 Bit
Saída DA	2	2	4
Limite	$\pm 3V$	$\pm 10V$	$\pm 10V$
Resolução	12 Bit	12 Bit	12 Bit
Frequência de amostragem	5 kHz	100 kHz	200 kHz
Slot no PC	-----	ISA – 16 Bit	ISA – 16 Bit

Endereço das empresas fabricantes / vendedoras:

Meilhaus Electronic GmbH	Conrad Electronic GmbH	Dr. Schetter BMC IGmbH
Fischerstraße 2	Klaus-Conrad-Strasse 1	Boschstr. 12
D-82178 Puchheim	D-92240 Hirschau	D-82178 Puchheim

<http://www.meilhaus.de>

<http://www.conrad.de>

<http://www.bmc.de>

## 2. Arquivos do Código Fonte

### 2.1. *Arquivo das classes*

CADInfo.cls	Contém informações da placa AD
CME300.cls	Classe para controle da placa AD ME300
CEdit_STM_Image.cls	Contém as funções de trabalho de imagens
CSTMTip.cls	Simula o comportamento da ponta do STM
CRandom.cls	Classe de simulação da placa AD/DA
CScale.cls	Contém a informação sobre a escala das imagens
CScan_STM_Image.cls	Contém as funções de aquisição de imagem
CSTM_Data.cls	Contém / gerencia os dados adquiridos
CStopUhr.cls	Implementa um cronômetro (timer)
CPC20TR.frm	Classe para controle da placa AD PC20TR

### 2.2. *Forms*

CSTM_Image.frm	Form de apresentação da imagem obtida pelo STM
CLineScan.frm	Form de apresentação da linha de scan da amostra
frm_coarseApproach.frm	Ajuda para aproximação grosseira da ponta
frm_EditImages.frm	Área de trabalho para edição de imagem
frm_ScanFrame.frm	Área de trabalho para aquisição de imagem
frm_StartUp.frm	Splash-screen
frmAbout.frm	Tela de informações
Main.frm	Programa principal

### 2.3. *Módulos*

Konstanten.bas	Definição de constantes globais
MathModule.bas	Definições de funções matemáticas
API_Interface.bas	Definição das API-Functions

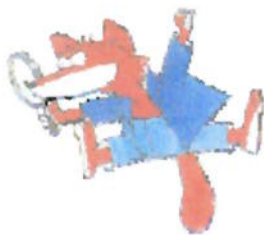
## **2.4. Usercontrols**

Saveimagecontrol.???	Controle de ajuste e gravação de imagem obtida.
ScanFrameOffsetControl.???	Controle de ajuste de Offset
ScanSizeControl.???	Controle do tamanho da área de scan
SngScrollControl.ctl	Controle de Scroll Bar

## **2.5. Arquivos do projeto**

Schueler-STM.vbp  
Schueler-STM.vbw

## **2.6. Arquivos de figura**



fuchslogo.gif  
Logo do Grupo de Trabalho do Prof. Fuchs



Quark+Co-Logo.gif  
Logo do monitor Quarks & Co.



SIGEL.JPG

Selo da Universidade de Münster



SCHLOSS.GIF

O símbolo da Universidade de Münster

### 3. Hierarquia de classes

#### 3.1. *Duas considerações iniciais*

Certamente seria bom, se agora e aqui uma pequena redação sobre Programação com Orientação a Objetos, Classes, Módulos e etc fosse feita. Por uma série de razões, não será feita tal redação, das quais duas são destacadas:

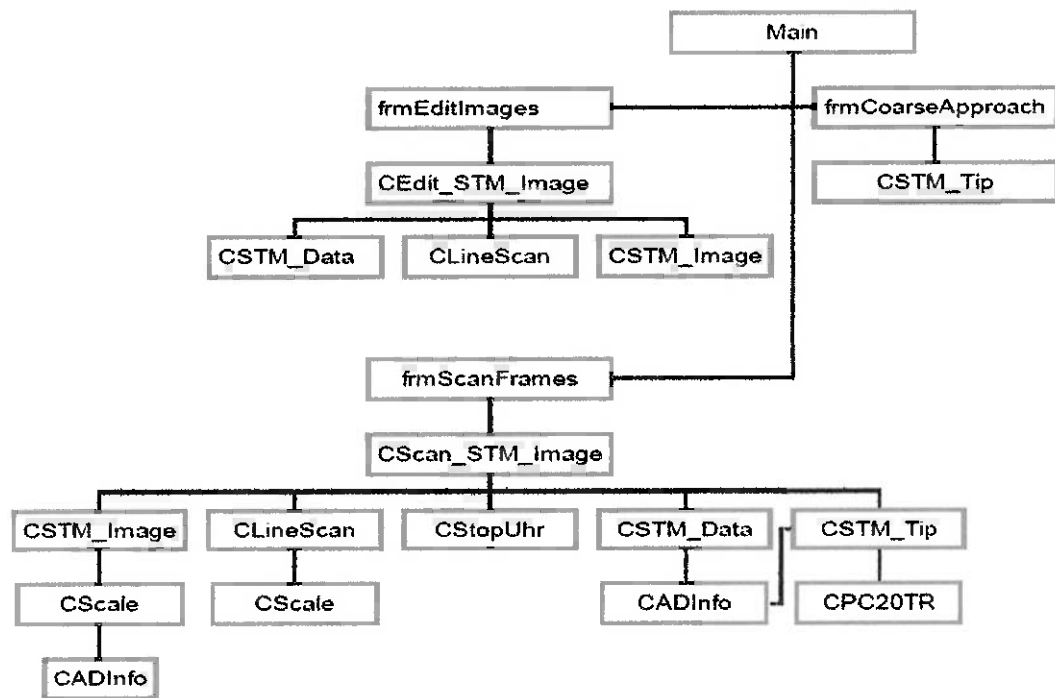
1. Caso fosse feita tal redação, este documento ficaria enorme e fugiria do escopo intencionado;
2. Este tipo de informação não deve ser encontrada em um documento sobre um software, mas sim em um livro ou guia de programação.

Caso o leitor não esteja familiarizado com os termos utilizados no Visual Basic (ou seja, não esteja habituado com Programação Orientada a Objetos), pode-se utilizar da ajuda do Visual Basic para esclarecer eventuais dúvidas.

### 3.2. Representação Gráfica

Como cada Objeto / Classe se relaciona com outro é melhor visualizado em um diagrama.

**Hierarquia de Forms e Classes**



(Representação completa apenas no ramo “frm\_Scan\_Frames)

### 3.3. CADInfo

#### 3.3.1. Eventos

Nenhum

#### 3.3.2. Propriedades

*MaxInputVoltage As Single*

A voltagem máxima que é lida pela placa de aquisição

*MinInputVoltage As Single*

A voltagem mínima que é lida pela placa de aquisição

*MaxSkalenteile As Long*

E em quantas seções é dividida a escala de diferença de voltagem (por regra, uma placa de 12-bit é  $4096 = 0xFFFF$ )

*OP\_Gain As Single*

Fator de amplificação da corrente de tunelamento, passando de nA para V.

### **3.3.3. Public**

*Friend Sub GetInfoFromSTMTip(Optional ByVal STMTip As CSTM\_Tip =  
Nothing)*

Define todas as propriedades das informações da classe CSTM\_Tip

*Friend Sub SetInfo(ByVal MaxInputVolt As Single, ByVal MinInputVolt As Single,  
ByVal MaxSkalenteil As Long, ByVal OPV\_Gain As Single)*

Define todas as propriedades de uma só vez

*Friend Function Skalenteile2NanoAmpere(ByVal ST As Long) As Single*

Calcula a taxa de transferência de escala de nA

*Friend Function Skalenteile2Label(ByVal ST As Long) As String*

Gera, a partir da escala de medida, um rótulo na forma "0.0 nA"

### **3.3.4. Private**

*Private varMaxSkalenteile As Long*

Grava a propriedade pertinente

*Private varMaxInputVoltage As Single*

Grava a propriedade pertinente

*Private varMinInputVoltage As Single*

Grava a propriedade pertinente

*Private varOPGain As Single,*

Grava a propriedade pertinente

*Private Sub Class\_Initialize()*

Inicializa a classe com valores padrão

### **3.4. CEdit STM\_Image**

#### **3.4.1. Eventos**

*DataManipulationStarted(ByVal typ As DataManipulationMethods)*

É iniciada quando uma rotina de manipulação de dados é chamada. typ é a variável que contém o código da rotina pertinente

*DataManipulationProgress(ByVal typ As DataManipulationMethods, ByVal Percent As Single)*

Ocorre em intervalos regulares de tempo, quando a manipulação de dados está operando, possibilitando a utilização de uma barra de progresso. typ é novamente a rotina e por cento para a medição do progresso da rotina.

*DataManipulationFinished(ByVal typ As DataManipulationMethods)*

É executado ao fim da manipulação de dados.

*ScaleHasChanged(ByVal AbsMin As Single, ByVal min As Single, ByVal max As Single, ByVal AbsMax As Single)*

É executado no momento que a escala da figura muda. Min e Max indicam os valores limites.



### 3.4.2. Propriedades

*ADInfo()* *As CADInfo*

Contém as informações da placa AD.

*CommonDialogControl()* *As CommonDialog*

Contém o elemento de controle da caixa de diálogo comum dos formulários.

*Resolution()* *As Long*

O tamanho atual da figura.

*MaxResolution()* *As Long*

O tamanho máximo da figura.

*STMFilesDirectory()* *As String*

A lista de diretórios com os arquivos.

*Visible()* *As Boolean*

Visibilidade da figura

*SampleInfo()* *As String*

Comentários sobre a tomada de dados (utilização do STM)

*Filename()* *As String*

Nome de arquivo atual

*ImageScale()* *As Cscale*

Escala da figura atual

### 3.4.3. Public

*Friend Sub ExecuteDataManipulation(typ As DataManipulationMethods,  
Optional ByVal PercentOfScale As Single = 100)*

Inicializa a manipulação de dados na figura atual

*Typ*: Código para a montagem da manipulação

*PercentOfScale*: Quantos porcentos da escala será utilizado

*Friend Sub ResetScale()*

Reinicializa a escala para valores padrões

*Friend Sub OptimizeScale(Optional ByVal PercentOfScale As Single = 100)*

Otimiza o uso da escala.

*PercentOfScale*: Quantos porcentos da escala será utilizado

*Friend Sub SubtractLine(Optional ByVal PercentOfScale As Single = 100)*

Constrói cada linha e ajusta a escala em seguida

*PercentOfScale*: Quantos porcentos da escala será utilizado

*Friend Sub FlattenData(Optional ByVal PercentOfScale As Single = 100)*

Constrói a figura e ajusta a escala em seguida

*PercentOfScale*: Quantos porcentos da escala será utilizado

*Friend Sub SmoothData()*

Suaviza (smooth) os dados, utilizando-se dos valores médios para cada ponto.

*Public Function LnData(Optional ByVal PercentOfScale As Single = 100)*

Logarítmo dos dados

*PercentOfScale*: Quantos porcentos da escala será utilizado

*Friend Sub Hide()*

Esconde a figura

*Friend Sub Show()*

Mostra a figura

*Friend Sub ShowLineScanZeile(ByVal Line As Long)*

Mostra um perfil de linha da linha *Line*

*Friend Sub ShowLineScanSpalte(ByVal Line As Long)*

Mostra um perfil de linha do range definido por *Line*

*Friend Sub HideLineScan()*

Esconde o perfil da linha

*Friend Sub SaveBMPImage(Optional ByVal Fname As String = "")*

Grava a fuigura como um arquivo BMP

*Fname*: Nome do arquivo. Será definido pelo usuário

*Friend Sub SaveSTMData(Optional ByVal Fname As String = "")*

Grava a figura como um arquivo de dados STM

*Fname*: Nome do arquivo. Será definido pelo usuário

*Friend Sub LoadSTMData(Optional ByVal Fname As String = "")*

Carrega um arquivo de dados STM

*Fname*: Nome do arquivo. Será definido pelo usuário

*Friend Sub ReloadData()*

Recarrega o atual arquivo de dados STM

#### 3.4.4. Private

*Private Sub Class\_Initialize()*

O construtor da classe

*Private Sub Class\_Terminate()*

O destrutor da classe

*Private Function AskForFilename(ByVal Extension As String, ByVal Filter As String) As String*

Pergunta por um nome de arquivo. Será utilizado em conjunto com *SaveBMPImage, SaveSTMData, LoadSTMData*

*Private WithEvents STMData As CSTM\_Data*

Contém os dados da figura

*Private WithEvents STMImage As CSTM\_Image*

Mostra a figura

*Private LineScan As ClineScan*

Mostra o perfil de linha

*Private mvarFilename As String 'lokale Kopie*

Nome do atual arquivo de dados

*Private STMFilesDir As String*

Grava os atributos pertinentes

*Private Cdlg As CommonDialog*

Grava os atributos pertinentes

### **3.5. CME300**

#### **3.5.1. Eventos**

*InitFailed()*

Será executado no caso da inicialização da placa de aquisição falhar

*AnalogInputRangeChanged(ByVal ChannelNr As ME300InChannelNumbers,*

*ByVal fromRange As ME300AnalogInputRanges,*

*ByVal toRange As ME300AnalogInputRanges)*

Será executado assim que um range de canal AD mudar

*ChannelNr*: Número do canal afetado

*fromRange*: Antigo range AD

*toRange*: Novo range AD

*AnalogOutputRangeChanged(ByVal ChannelNr As ME300OutChannelNumbers,*

*ByVal fromRange As ME300AnalogOutputRanges,*

*ByVal toRange As ME300AnalogOutputRanges)*

Será executado assim que um range de canal DA mudar

*ChannelNr*: Número do canal afetado

*fromRange*: Antigo range DA

*toRange*: Novo range DA

#### **3.5.2. Propriedades**

*BoardNumber()* As ME300CardNumbers

Número de utilização da placa ME300

*MaxSkalenteile() As Long*

Resolução da escala da placa AD/DA. A placa ME300 tem uma resolução de 12-bit, tal que o valor sempre será 4095=0xFFF

### **3.5.3. Public**

*Public Sub InitCard(Optional ByVal nr As ME300CardNumbers =  
ME300NoCard)*

Inicializa a placa ME300 com o número *nr*

*Public Sub SetInputChannelRange(ByVal ChannelNr As  
ME300InChannelNumbers, ByVal range As  
ME300AnalogInputRanges)*

Define o range do canal AD de número *ChannelNr*

*Public Function GetInputChannelRange(ByVal ChannelNr As  
ME300InChannelNumbers) As  
ME300AnalogInputRanges*

Recupera o range do canal AD de número *ChannelNr*

*Public Function MaxInputVoltage(ByVal ch As ME300InChannelNumbers) As  
Single*

Recupera do canal AD *ch* a voltagem máxima que pode ser lida

*Public Function MinInputVoltage(ByVal ch As ME300InChannelNumbers) As  
Single*

Recupera do canal AD *ch* a voltagem mínima que pode ser lida

*Public Sub SetOutputChannelRange(ByVal ChannelNr As  
ME300OutChannelNumbers, ByVal range  
As ME300AnalogOutputRanges)*

Define o range do canal DA de número *ChannelNr*

*Public Function GetOutputChannelRange(ByVal ChannelNr As  
ME300OutChannelNumbers) As  
ME300AnalogOutputRanges*

Recupera o range do canal DA de número *ChannelNr*

*Public Function MaxOutputVoltage(ByVal ch As ME300OutChannelNumbers) As  
Single*

Recupera do canal DA *ch* a voltagem máxima que pode ser lida

*Public Function MinOutputVoltage(ByVal ch As ME300OutChannelNumbers) As  
Single*

Recupera do canal DA *ch* a voltagem mínima que pode ser lida

*Public Sub AnalogOutput(ByVal channel As ME300OutChannelNumbers, ByVal  
signal As Single, Optional ByVal OutputRange As  
ME300AnalogOutputRanges = AO\_NORange)*

Recupera uma voltagem de um canal da placa ME300

*channel*: Número do canal de aquisição

*signal*: Voltagem a ser adquirida em V

*OutputRange*: range do canal DA (opcional)

*Public Sub DirectAnalogOutput(ByVal channel As ME300OutChannelNumbers,  
ByVal value As Long, Optional ByVal  
OutputRange As ME300AnalogOutputRanges =  
AO\_NORange)*

Recupera uma voltagem de um canal da placa ME300

*channel*: Número do canal de aquisição

*value*: Sinal a ser adquirido no formato digital, 0x000 – 0xFFFF

*OutputRange*: range do canal DA (opcional)

*Public Function AnalogInput(ByVal channel As ME300InChannelNumbers,  
Optional ByVal InputRange As  
ME300AnalogInputRanges = AI\_NORange) As  
Single*

Lê uma voltagem a partir do canal do ME300

*channel*: Número do canal de aquisição

*OutputRange*: range do canal AD (opcional)

Retorna: Voltagem em V

*Public Function DirectAnalogInput(ByVal channel As ME300InChannelNumbers,  
Optional ByVal InputRange As  
ME300AnalogInputRanges = AI\_NORange)  
As Integer*

Lê uma voltagem a partir do canal do ME300

*channel*: Número do canal de aquisição

*OutputRange*: range do canal AD (opcional)

Retorna: Sinal na forma digital, 0x000 – 0xFFFF

*Public Function ConvertInSignal(ByVal value As Integer, ByVal InputRange As  
ME300AnalogInputRanges) As Single*

Converte um sinal digital em analógico

*value*: Sinal digital

*InputRange*: range AD, no qual o sinal convertido deve estar

Retorno: Voltagem em V



*Public Sub FillComboBox(cb As ComboBox, IODirection As ME300Directions,  
Optional ByVal DynamicAllowed As Boolean = False)*

Completa uma ComboBox com um range de voltagem

*cb*: A ComboBox alvo

*IODirection*: Range para o canal DA (Saída) ou para o canal AD (entrada)

*DynamicAllowed*: Se um ajuste dinâmico do range for necessário, ativa-se esta propriedade

*Public Function Combo2RangeConst(cb As ComboBox, IODirection As  
ME300Directions)*

Converte o range de uma ComboBox para uma constante pertinente

*cb*: A ComboBox alvo

*IODirection*: Range para o canal DA (Saída) ou para o canal AD (entrada)

Retorno: Constante de range

#### **3.5.4. Private**

*Private Sub Class\_Initialize()*

Construtor da classe

*Private Sub Class\_Terminate()*

Destrutor da classe

*Private Function ConvertOutSignal(ByVal signal As Single, ByVal OutRange As  
ME300AnalogOutputRanges) As Integer*

Converte um valor analógico em digital

*signal*: Voltagem analógica em V

*OutRange*: Range DA no qual o valor digital deve estar contido

Retorno: Valor digital

## **3.6. CPC20TR**

### **3.6.1. Eventos**

*InitFailed()*

Será executado no caso da inicialização da placa de aquisição falhar

*AnalogInputRangeChanged(ByVal ChannelNr As PC20TR\_InChannelNumbers,  
ByVal fromRange As PC20TR\_AnalogInputRanges,  
ByVal toRange As PC20TR\_AnalogInputRanges)*

Será executado assim que um range de canal AD mudar

*ChannelNr*: Número do canal afetado

*fromRange*: Antigo range AD

*toRange*: Novo range AD

*AnalogOutputRangeChanged(ByVal ChannelNr As  
PC20TR\_OutChannelNumbers, ByVal fromRange  
As PC20TR\_AnalogOutputRanges, ByVal toRange  
As PC20TR\_AnalogOutputRanges)*

Será executado assim que um range de canal DA mudar

*ChannelNr*: Número do canal afetado

*fromRange*: Antigo range DA

*toRange*: Novo range DA

### **3.6.2. Propriedades**

*BoardNumber() As PC20TR\_CardNumbers*

Número de utilização da placa PC20TR

*MaxSkalenteile() As Long*

Resolução da escala da placa AD/DA. A placa PC20TR tem uma resolução de 12-bit, tal que o valor sempre será 4095=0xFFF

### 3.6.3. Public

*Public Sub InitCard(Optional ByVal nr As PC20TR\_CardNumbers =  
PC20TR\_NoCard)*

Inicializa a placa PC20TR com o número *nr*

*Public Sub SetInputChannelRange(ByVal ChannelNr As  
PC20TR\_InChannelNumbers, ByVal range As  
PC20TR\_AnalogInputRanges)*

Define o range do canal AD de número *ChannelNr*

*Public Function GetInputChannelRange(ByVal ChannelNr As  
PC20TR\_InChannelNumbers) As  
PC20TR\_AnalogInputRanges*

Recupera o range do canal AD de número *ChannelNr*

*Public Function MaxInputVoltage(ByVal ch As PC20TR\_InChannelNumbers) As  
Single*

Recupera do canal AD *ch* a voltagem máxima que pode ser lida

*Public Function MinInputVoltage(ByVal ch As PC20TR\_InChannelNumbers) As  
Single*

Recupera do canal AD *ch* a voltagem mínima que pode ser lida

*Public Sub SetOutputChannelRange(ByVal ChannelNr As  
PC20TR\_OutChannelNumbers, ByVal range  
As PC20TR\_AnalogOutputRanges)*

Define o range do canal DA de número *ChannelNr*

*Public Function GetOutputChannelRange(ByVal ChannelNr As  
PC20TR\_OutChannelNumbers) As  
PC20TR\_AnalogOutputRanges*

Recupera o range do canal DA de número *ChannelNr*

*Public Function MaxOutputVoltage(ByVal ch As PC20TR\_OutChannelNumbers)  
As Single*

Recupera do canal DA *ch* a voltagem máxima que pode ser lida

*Public Function MinOutputVoltage(ByVal ch As PC20TR\_OutChannelNumbers)  
As Single*

Recupera do canal DA *ch* a voltagem mínima que pode ser lida

*Public Sub AnalogOutput(ByVal channel As PC20TR\_OutChannelNumbers,  
ByVal signal As Single, Optional ByVal OutputRange  
As PC20TR\_AnalogOutputRanges = AO\_NORange)*

Recupera uma voltagem de um canal da placa PC20TR

*channel*: Número do canal de aquisição

*signal*: Voltagem a ser adquirida em V

*OutputRange*: range do canal DA (opcional)

*Public Sub DirectAnalogOutput(ByVal channel As PC20TR\_OutChannelNumbers,  
ByVal value As Long, Optional ByVal  
OutputRange As PC20TR\_AnalogOutputRanges  
=AO\_NORange)*

Recupera uma voltagem de um canal da placa PC20TR

*channel*: Número do canal de aquisição

*value*: Sinal a ser adquirido no formato digital, 0x000 – 0xFFFF

*OutputRange*: range do canal DA (opcional)

*Public Function AnalogInput(ByVal channel As PC20TR\_InChannelNumbers,  
Optional ByVal InputRange As  
PC20TR\_AnalogInputRanges = AI\_NORange) As  
Single*

Lê uma voltagem a partir do canal do PC20TR

*channel*: Número do canal de aquisição

*OutputRange*: range do canal AD (opcional)

Retorna: Voltagem em V

*Public Function DirectAnalogInput(ByVal channel As  
PC20TR\_InChannelNumbers, Optional  
ByVal InputRange As  
PC20TR\_AnalogInputRanges =  
AI\_NORange) As Integer*

Lê uma voltagem a partir do canal do PC20TR

*channel*: Número do canal de aquisição

*OutputRange*: range do canal AD (opcional)

Retorna: Sinal na forma digital, 0x000 – 0xFFFF

*Public Function ConvertInSignal(ByVal value As Integer, ByVal InputRange As  
PC20TR\_AnalogInputRanges) As Single*

Converte um sinal digital em analógico

*value*: Sinal digital

*InputRange*: range AD, no qual o sinal convertido deve estar

Retorno: Voltagem em V

*Public Sub FillComboBox(cb As ComboBox, IODirection As PC20TR\_Directions,  
Optional ByVal DynamicAllowed As Boolean = False)*

Completa uma ComboBox com um range de voltagem

*cb*: A ComboBox alvo

*IODirection*: Range para o canal DA (Saída) ou para o canal AD (entrada)

*DynamicAllowed*: Se um ajuste dinâmico do range for necessário, ativa-se esta propriedade

*Public Function Combo2RangeConst(cb As ComboBox, IODirection As  
PC20TR\_Directions)*

Converte o range de uma ComboBox para uma constante pertinente

*cb*: A ComboBox alvo

*IODirection*: Range para o canal DA (Saída) ou para o canal AD (entrada)

Retorno: Constante de range

#### **3.6.4. Private**

*Private Sub Class\_Initialize()*

Construtor da classe

*Private Sub Class\_Terminate()*

Destrutor da classe

*Private Function ConvertOutSignal(ByVal signal As Single, ByVal OutRange As  
PC20TR\_AnalogOutputRanges) As Integer*

Converte um valor analógico em digital

*signal*: Voltagem analógica em V

*OutRange*: Range DA no qual o valor digital deve estar contido

Retorno: Valor digital

### **3.7. CRandom**

Esta classe é análoga à classe CME300 e deve simular uma placa ME300. CRandom foi requisitado no começo do projeto para simular e testar a funcionalidade e programação dos arquivos. Quando o projeto estava em um

estágio mais avançado, a classe CME300 foi criada e a partir de então a classe CME300 não seria mais implementada. Caso esta classe seja utilizada no software, modificações no software têm que ser feitas de forma a fazê-la funcionar novamente.

### 3.7.1. Eventos

*AnalogInputRangeChanged(ByVal ChannelNr As ME300InChannelNumbers,  
ByVal fromRange As ME300AnalogInputRanges,  
ByVal toRange As ME300AnalogInputRanges)*

Será executado assim que um range de canal AD mudar

*ChannelNr*: Número do canal afetado

*fromRange*: Antigo range AD

*toRange*: Novo range AD

*AnalogOutputRangeChanged(ByVal ChannelNr As ME300OutChannelNumbers,  
ByVal fromRange As ME300AnalogOutputRanges,  
ByVal toRange As ME300AnalogOutputRanges)*

Será executado assim que um range de canal DA mudar

*ChannelNr*: Número do canal afetado

*fromRange*: Antigo range DA

*toRange*: Novo range DA

### 3.7.2. Propiedades

*MaxSkalenteile()* As Long

Resolução da escala da placa AD/DA. A placa ME300 tem uma resolução de 12-bit, tal que o valor sempre será 4095=0xFFF

### 3.7.3. Public

*Public Sub InitCard(ByVal c As Long)*

Função dummy para a compatibilização com a classe CME300

*Public Sub SetInputChannelRange(ByVal ch As Long, ByVal range As Long)*

Define o valor da variável local *miv* de acordo com o valor de *range*

*Public Function MaxInputVoltage(ByVal ch As Long) As Single*

Recupera do canal AD *ch* a voltagem máxima que pode ser lida ( $=+miv$ )

*Public Function MinInputVoltage(ByVal ch As Long) As Single*

Recupera do canal AD *ch* a voltagem mínima que pode ser lida ( $=-miv$ )

*Public Sub SetOutputChannelRange(ByVal ch As Long, ByVal range As Long)*

Função dummy para a compatibilização com a classe CME300

*Public Function MaxOutputVoltage(ByVal ch As Long) As Single*

Recupera do canal DA *ch* a voltagem máxima que pode ser lida ( $=+miv$ )

*Public Function MinOutputVoltage(ByVal ch As Long) As Single*

Recupera do canal DA *ch* a voltagem mínima que pode ser lida ( $=-miv$ )

*Public Sub AnalogOutput(ByVal ch As Long, ByVal signal As Single)*

Grava o valor de *signal* na variável local *x*

*Public Sub DirectAnalogOutput(ByVal ch As Long, ByVal value As Long)*

Função dummy para a compatibilização com a classe CME300

*Public Function AnalogInput(ByVal channel As ME300InChannelNumbers,*

*Optional ByVal InputRange As*



*ME300AnalogInputRanges = AI\_NORange) As  
Single*

Lê uma voltagem a partir do canal do ME300

*channel*: Número do canal de aquisição

*OutputRange*: range do canal AD (opcional)

Retorna: Voltagem em V

*Public Function DirectAnalogInput(ByVal channel As ME300InChannelNumbers,  
Optional ByVal InputRange As  
ME300AnalogInputRanges = AI\_NORange)  
As Integer*

Lê uma voltagem a partir do canal do ME300

*channel*: Número do canal de aquisição

*OutputRange*: range do canal AD (opcional)

Retorna: Sinal na forma digital, 0x000 – 0xFFFF

*Public Function ConvertInSignal(ByVal value As Integer, ByVal InputRange As  
ME300AnalogInputRanges) As Single*

Converte um sinal digital em analógico

*value*: Sinal digital

*InputRange*: range AD, no qual o sinal convertido deve estar

Retorno: Voltagem em V

*Public Sub FillComboBox(cb As ComboBox, IODirection As ME300IDirections,  
Optional ByVal DynamicAllowed As Boolean = False)*

Completa uma ComboBox com um range de voltagem

*cb*: A ComboBox alvo

*IODirection*: Range para o canal DA (Saída) ou para o canal AD (entrada)

*DynamicAllowed*: Se um ajuste dinâmico do range for necessário, ativa-se esta propriedade

*Public Function Combo2RangeConst(cb As ComboBox, IODirection As  
ME300Directions)*

Converte o range de uma ComboBox para uma constante pertinente

*cb*: A ComboBox alvo

*IODirection*: Range para o canal DA (Saída) ou para o canal AD (entrada)

Retorno: Constante de range

#### **3.7.4. Private**

*Private miv As Single*

Guarda a maior voltagem absoluta

*Private Sub Class\_Initialize()*

Construtor da classe

### **3.8. CScale**

#### **3.8.1. Eventos**

*ScaleHasChanged(ByVal AbsMin As Single, ByVal min As Single, ByVal max As  
Single, ByVal AbsMax As Single)*

Será executado quando a escala da figura for mudada. A variável guarda o novo valor da escala.

*ColormapChanged()*

Será executado quando o valor do mapa de cores mudar.

#### **3.8.2. Propriedades**

*ADInfo()*

Guarda a informação AD atual

*AbsoluteMaximumValue() As Single*

Guarda o valor absoluto máximo da escala em nA

*AbsoluteMinimumValue() As Single*

Guarda o valor absoluto mínimo da escala em nA

*ScaleMinLabel() As String*

Nomeia o valor mínimo da escala

*ScaleMinValue() As Single*

Guarda o atual menor valor da escala em nA

*ScaleMaxLabel() As String*

Nomeia o valor máximo da escala

*ScaleMaxValue() As Single*

Guarda o atual maior valor da escala em nA

### **3.8.3. Public**

*Friend Sub DrawScale(ByVal left As Long, ByVal top As Long, ByVal width As Long, ByVal window As Form)*

Desenha uma escala no canto superior esquerdo da janela *window*

*Friend Sub ResetScale()*

Define a escala para o valor padrão

*Friend Sub SetAbsScaleValues(ByVal AbsMinValue As Single, ByVal AbsMaxValue As Single)*

Define o valor máximo absolute da escala

*Friend Sub SetAllScaleValues(ByVal AbsMin As Single, ByVal min As Single, ByVal max As Single, ByVal AbsMax As Single)*

Define o valor máximo (absolute e atual) da escala

*Friend Sub SetScaleValues(ByVal MinValue As Single, ByVal MaxValue As Single)*

Define o máximo valor atual da escala

*Friend Sub SetDefaultColormap()*

Define o mapa de cores padrão

*Friend Function GetColor(ByVal vdata As Single) As Long*

Retorna a cor do mapa de cores equivalente ao valor *vdata*

#### **3.8.4. Private**

*Private Sub Class\_Initialize()*

Construtor da classe

*Private Sub Class\_Terminate()*

Destrutor da classe

*Private Sub CalcScaleFactor()*

### **3.9. CScan\_STM\_Image**

#### **3.9.1. Eventos**

*ScanSizeHasChanged()*

É executado quando o tamanho do scan é mudado

*ResolutionHasChanged(ByVal oldRes As Long, ByVal NewRes As Long)*

É executado quando a resolução do scan é mudada

*oldRes*: número de pixels da imagem antigo

*NewRes*: número de pixels da imagem novo

*DataSaved()*

É executado quando a imagem é gravada

*NewScanVelocity(ByVal t As Double)*

É executado quando uma nova velocidade de scan é calculada

*t*: velocidade em segundos por linha.

### **3.9.2. Propriedades**

*ScanAngle() As Single*

O ângulo em graus, o qual a estrutura do STM tem

*LogarithmSignal() As Boolean*

O sinal deve ser medidos aplicando-se uma escala logarítmica?

*LinesPerSecond() As Double*

Atual velocidade de scan em linhas por segundo

*DelayAfterPixel() As Long*

Pausa em ms até o próximo pixel

*Get Tip() As C.STM\_Tip*

Retorna os dados da ponta da classe

*DatafilePath() As String*

Diretório dos arquivos de dados

*FitScaleToFirstLine() As Boolean*

Deve a escala ser dinamicamente proporcional à primeira linha da imagem?

*ScanDirection() As ScanDirectionConstants*

Direção do scan. Esquerda->Direita; Esquerda<-Direita ou  
Esquerda<->Direita

*Resolution() As Long*

Resolução da imagem em pixels

*SampleInfo() As String*

Comentários sobre a amostra

*XOffset() As Long*

X-Offset da estrutura do microscópio em nm

*YOffset() As Long*

Y-Offset da estrutura do microscópio em nm

*ScanSize() As Long*

Tamanho do scan em nm

*ImageVisible() As Boolean*

Mostrar figura?

*LineScanVisible() As Boolean*

Mostrar perfil de linha?

*SaveData() As SaveDataConstants*

Modo de gravação do arquivo.

### **3.9.3. Public**

*Friend Sub ShowImage()*

Mostra a imagem do STM

*Friend Sub HideImage()*

Esconde a imagem do STM

*Friend Sub ShowLineScan()*

Mostra o perfil de linha

*Friend Sub HideLineScan()*

Esconde o perfil de linha

*Friend Sub HideAll()*

Esconde o perfil de linha e a imagem do STM

*Friend Sub SaveSTMData()*

Grava os dados atuais

*Friend Sub StopScan()*

Interrompe o scan

*Friend Sub StartScan(Optional ByVal Direction As ScanDirectionConstants =  
None)*

Começa o scan

#### **3.9.4. Private**

*Private Sub Class\_Initialize()*

Construtor da classe

*Private Sub Class\_Terminate()*

Destruitor da classe

*Private Sub refresh\_ADInfo()*

Atualiza a informação AD de acordo com o valor adquirido na ponta

*Private Sub FitScaleToLine(ByVal y As Long, Optional ByVal PercentOfScale As  
Single = 100)*

Faz a escala ficar de acordo com o tamanho da linha y. *PercentOfScale* informa o quanto a escala foi mudada.

*Private Sub ProcessCurrentSignal(Optional signal As Single = -234)*

Lê o a corrente de tunelamento atual e processa o valor

*Private Sub Scan\_LR()*

Controla um scan com captura de dados saído da esquerda para a direita

*Private Sub Scan\_RL()*

Controla um scan com captura de dados saído da direita para a esquerda

### **3.10. CSTM Data**

#### **3.10.1. Eventos**

*ResolutionHasChanged(ByVal oldRes As Long, ByVal NewRes As Long)*

É executado quando a resolução do scan é mudada

#### **3.10.2. Propriedades**

*STMDataFileExtension() As String*

Extensão dos arquivos de dados STM

*ScanSize() As Long*

Tamanho do scan em nm



*Resolution() As Long*

Resolução da imagem em pixels

*SampleInfo() As String*

Comentários sobre a amostra

*ADInfo() As CADInfo*

Informação AD

### **3.10.3. Public**

*Public Sub CopyInto(ByRef newData As CSTM\_Data)*

Copia os dados da instância da classe na instância *newData* (assim como um construtor de cópia)

*Friend Function Save(ByVal path As String, Optional ByVal*

*PathContainsFilename As Boolean = False)*

*As STMDataFileResult*

Salva os dados atuais como um arquivo STM

*Path*: Caminho do arquivo

*PathContainsFilename*: usado se o caminho contém um nome de arquivo

*Friend Function Load(ByVal Filename As String) As STMDataFileResult*

Carrega um arquivo de dados

*Friend Sub SetDataPoint(ByVal x As Long, ByVal y As Long, ByVal signal As  
Single)*

Define o valor de um ponto no scan

*(x,y)*: Coordenadas do ponto

*signal*: corrente de tunelamento em nA

*Friend Function getDataPoint(ByVal x As Long, ByVal y As Long) As Single*

Retorna o valor da corrente de tunelamento em nA do ponto (x,y)

#### **3.10.4. Private**

*Private Sub Class\_Initialize()*

Construtor da classe

*Private Sub Class\_Terminate()*

Destrutor da classe

As 5 funções a seguir são para a aquisição de dados e não necessitam de esclarecimento:

*Private Sub GotoEndOfFileHeader()*

*Private Function PutText(s As String)*

*Private Sub GetString(s As String)*

*Private Function ExtractInt(s As String) As Long*

*Private Function ExtractSng(s As String) As Single*

### **3.11. CSTMTip**

#### **3.11.1. Eventos**

*FrameBorderReached(position As CSTFBR)*

É executado quando as margens do scan é atingida. *Position* indica a posição das bordas

*TipPropertiesChanged()*

É executado quando alguma propriedade da classe CSTMTip muda

### **3.11.2. Propriedades**

*ScanAngle() As Single*

O ângulo em graus, o qual a estrutura do STM tem

*Logarithm() As Boolean*

O sinal deve ser medidos aplicando-se uma escala logaritimica?

*MaxInputVoltage() As Single*

Voltagem máxima de entrada (depende da placa IO)

*MinInputVoltage() As Single*

Voltagem mínima de entrada (depende da placa IO)

*MinOutputVoltage() As Single*

Voltagem mínima de saída (depende da placa IO)

*MaxOutputVoltage() As Single*

Voltagem máxima de saída (depende da placa IO)

*MaxSkalenteile() As Long*

Número máximo de divisões de escala (depende da placa IO)

*XOffset() As Long*

X-Offset da estrutura do microscópio em nm

*YOffset() As Long*

Y-Offset da estrutura do microscópio em nm

*ScanSize() As Long*

Tamanho do scan

*Resolution() As Long*

Resolução do scan

*y() As Long*

Posição atual y da ponta em pixel

*x() As Long*

Posição atual x da ponta em pixel

*wasFirstLineInFrame() As Boolean*

A última linha gerada foi a primeira da imagem?

*IOKarte() As \_\_\_\_\_*

Retorna uma instância da classe da placa IO

### **3.11.3. Public**

*SetInputVoltageRange(ByVal v As Single)*

Define o range de voltagens, onde v é o valor ótimo de digitalização em V

*Friend Sub SetXYOffset(ByVal deltaX As Long, ByVal deltaY As Long)*

Define o offset do scan

*Friend Sub setXY(ByVal Xvalue As Long, ByVal Yvalue As Long)*

Define a atual posição da ponta

*Friend Sub NextLine()*

Movimenta a ponta para a próxima linha

*Friend Sub MoveHome()*

Movimenta a ponta para a posição (1,1)

*Friend Sub MoveToFirstCol()*

Movimenta a ponta para a primeira coluna

*Friend Sub MoveToLastCol()*

Movimenta a ponta para a última coluna

*Friend Function getSignal() As Single*

Lê a corrente de tunelamento em escala logarítmica

#### **3.11.4. Private**

*Private Sub SetTip()*

Define todos os parâmetros da ponta (x, y, offset x, offset y, ângulo)

*Private Function getI() As Single*

Lê a corrente de tunelamento

Retorna: valor da voltagem amplificada do tunelamento

### **3.12. CstopUhr**

#### **3.12.1. Eventos**

*Nenhum*

#### **3.12.2. Propriedades**

*TimeElapsed() As Double*

Retorna o valor do tempo parado em segundos

### **3.12.3. Public**

*Friend Sub Start()*

Aciona o timer

*Friend Sub Stopp()*

Pára o timer

### **3.12.4. Private**

*Private Function SystemTime2Seconds(ST As SYSTEMTIME) As Double*

Calcula o horário do sistema em segundos

## **3.13. CLineScan**

### **3.13.1. Eventos**

### **3.13.2. Propriedades**

*ADInfo() As CADInfo*

*Resolution() As Long*

*ScanSize() As Long*

*ImageScale() As CScale*

### **3.13.3. Public**

*Friend Sub DrawPoint(x As Long, y As Long, signal As Single)*

### **3.13.4. Private**

*Private Sub Form\_Paint()*

*Private Sub Resize\_Form()*

### **3.14. CSTM Image**

#### **3.14.1. Eventos**

*Public Event ColormapHasChanged()*

*Public Event ScaleHasChanged(ByVal AbsMin As Single, ByVal min As Single,  
ByVal max As Single, ByVal AbsMax As Single)*

#### **3.14.2. Propriedades**

*ImageScale() As Cscale*

Escala

*ADInfo() As CADInfo*

Informações AD da ponta

*UserCanUnload() As Boolean*

O usuário pode fechar o form?

*Resolution() As Long*

Resolução da imagem em pixels

*ScanSize() As Long*

Tamanho do scan em nm

*SampleInfo() As String*

Comentários sobre a amostra

#### **3.14.3. Public**

*Friend Sub RedrawScale()*

Desenha a nova escala

*Friend Sub DrawPoint(ByVal x As Long, ByVal y As Long, ByVal signal As Single)*

Desenha um ponto na posição (x,y)

#### **3.14.4. Private**

*Private Sub Form\_Paint()*

Desenha a escala e nomeia a imagem

*Private Sub DrawBox(x1 As Long, y1 As Long, x2 As Long, y2 As Long, color As Long, Optional ByVal cleaned As Boolean = False)*

Desenha uma caixa

*Private Sub WriteText(ByVal x As Long, ByVal y As Long, ByVal t As String, Optional ByVal clean As Boolean = False)*

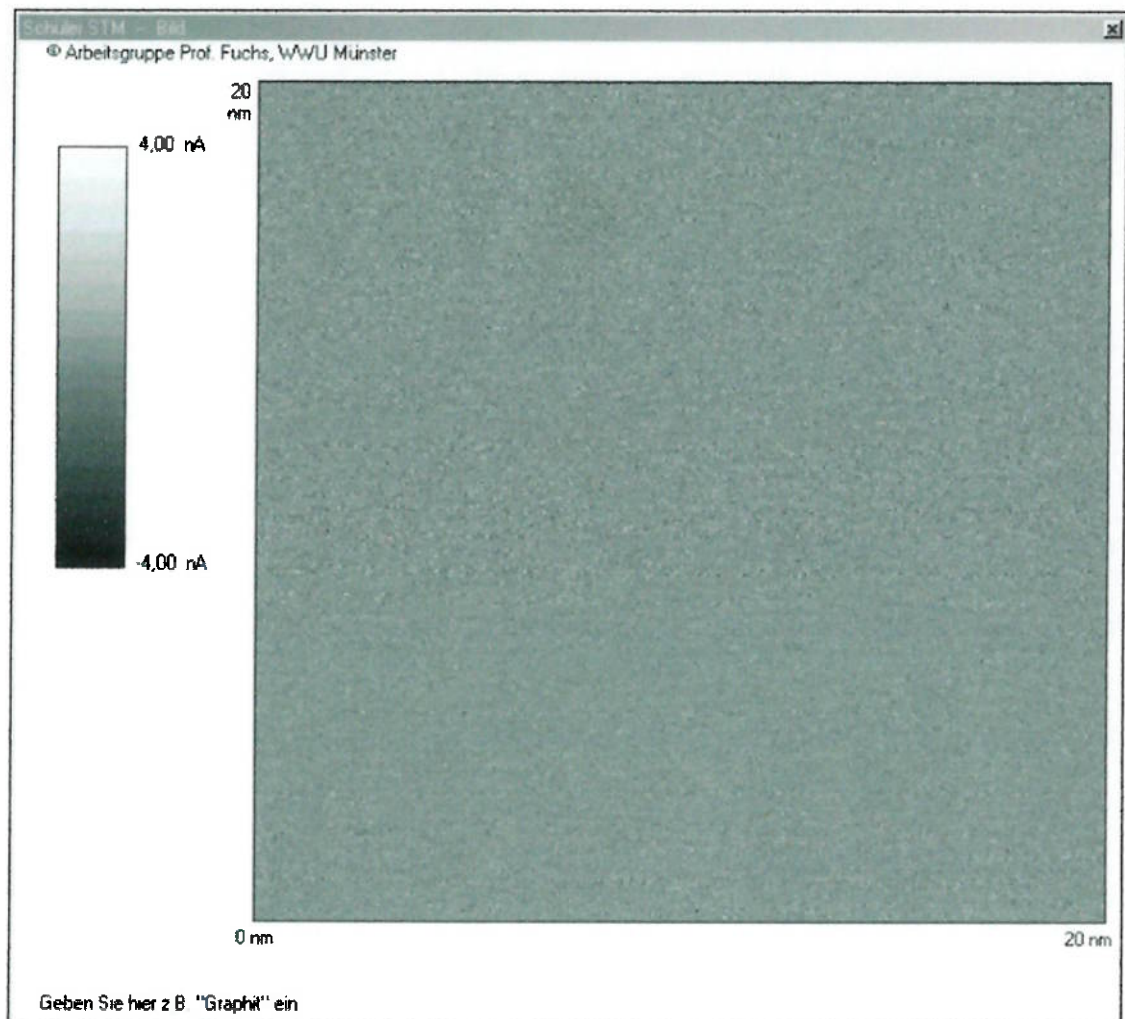
Escreve um texto e apaga um anterior, se necessário

## **4. Forms**

### **4.1. CSTM\_Image.frm**

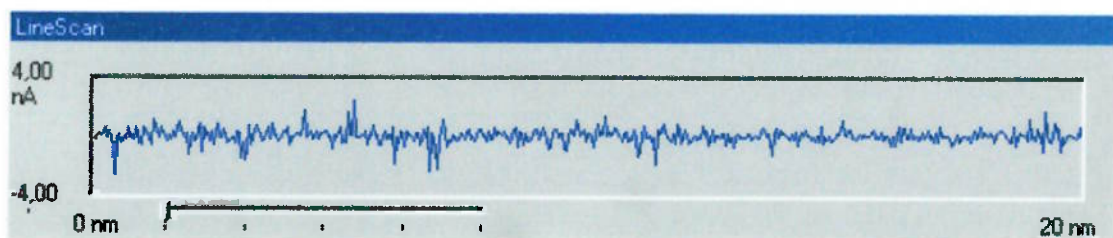
É neste form que a imagem obtida é exposta. Este form usará uma classe.





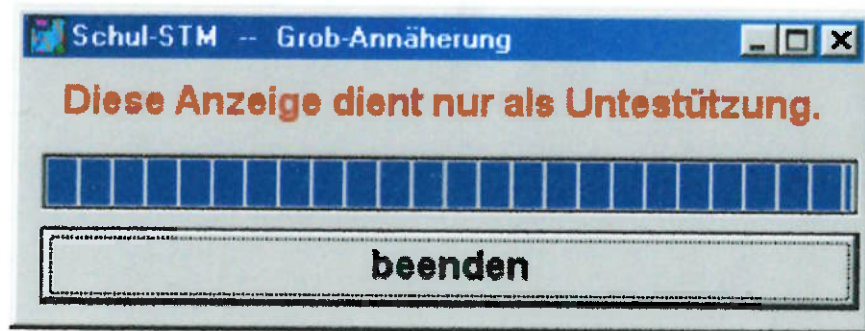
#### 4.2. *ClineScan.frm*

Neste form é exibido um perfil de linha obtido. Ele usa uma classe.



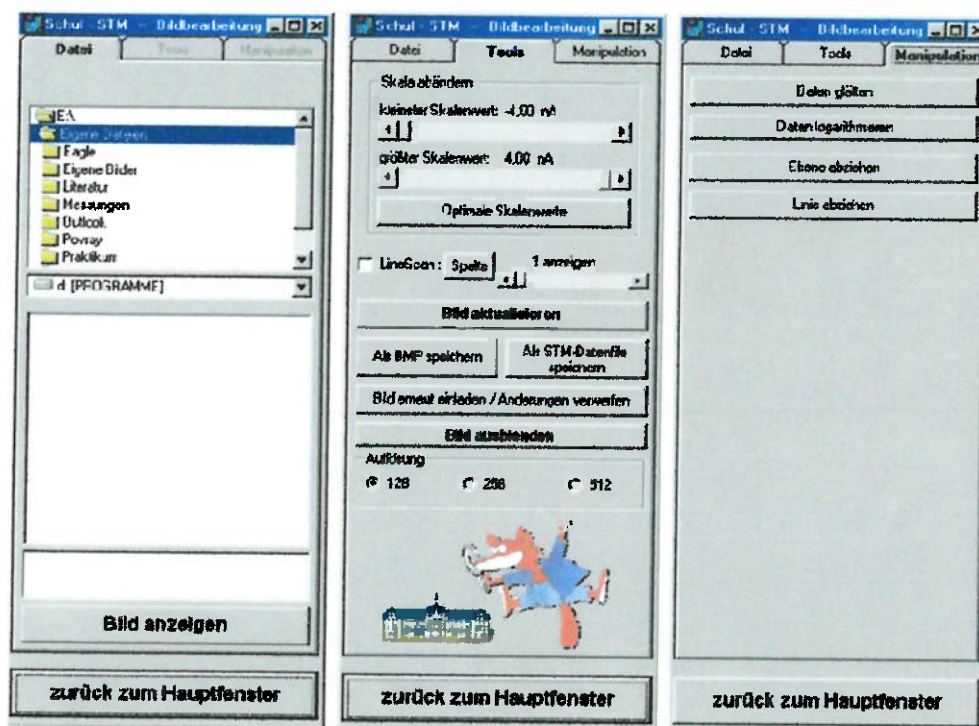
### 4.3. *frm\_coarse\_Approach.frm*

Este form tem como função auxiliar a aproximação entre a ponta e a amostra. O gráfico de barras mostra a corrente de tunelamento naquele momento.



### 4.4. *frm\_EditImages.frm*

Este form é chamado quando o usuário entra no modo de trabalho de imagem. Ele contém todas as funções mostradas na imagem abaixo.



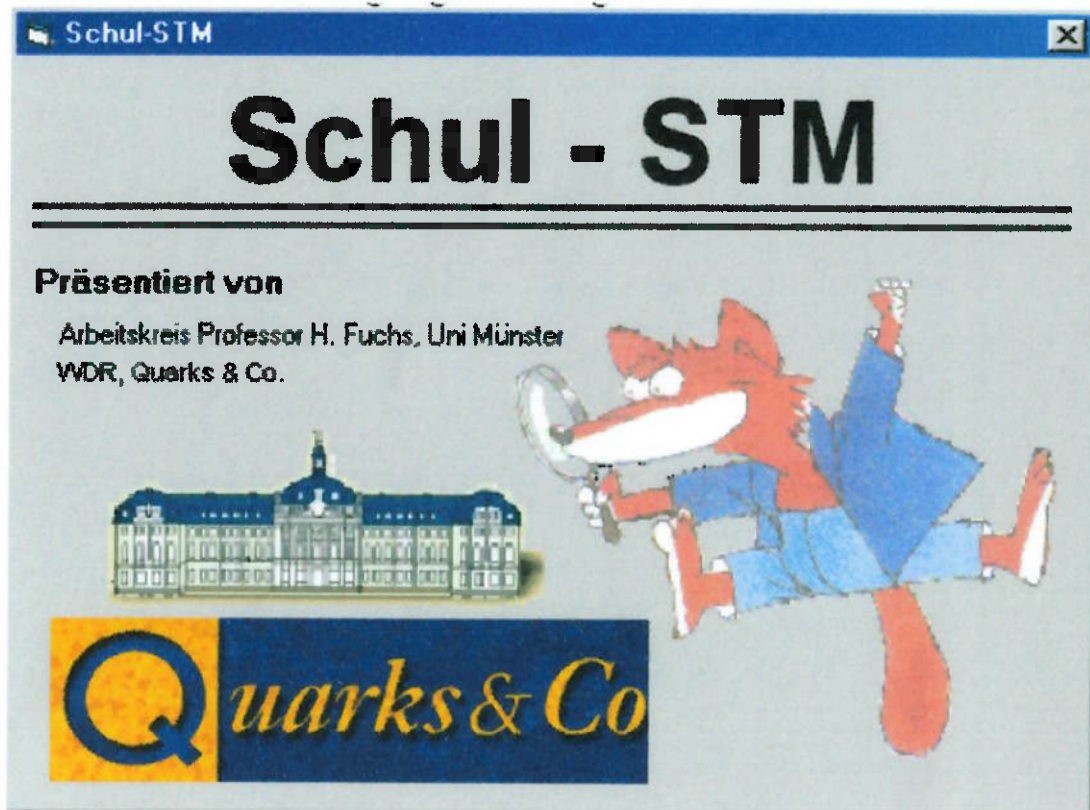
#### 4.5. *frm\_ScanFrame.frm*

Este form aparece quando o usuário entra no modo de edição de imagem. Ele contém todas as funções mostradas na imagem abaixo.



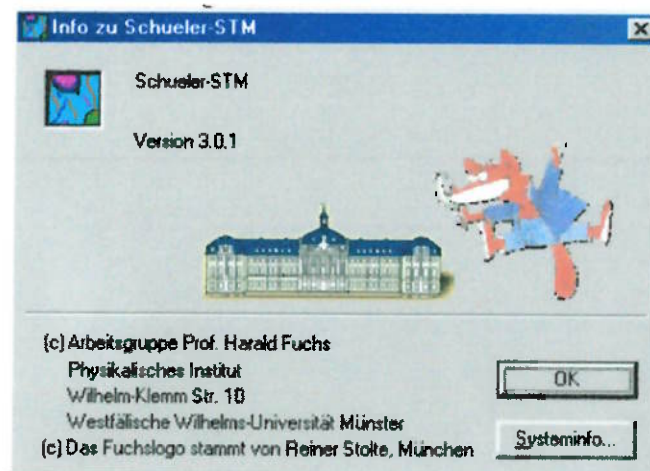
#### 4.6. *frm\_StartUp.frm*

Este form mostra o logo do programa, na inicialização do software



#### 4.7. *frmAbout.frm*

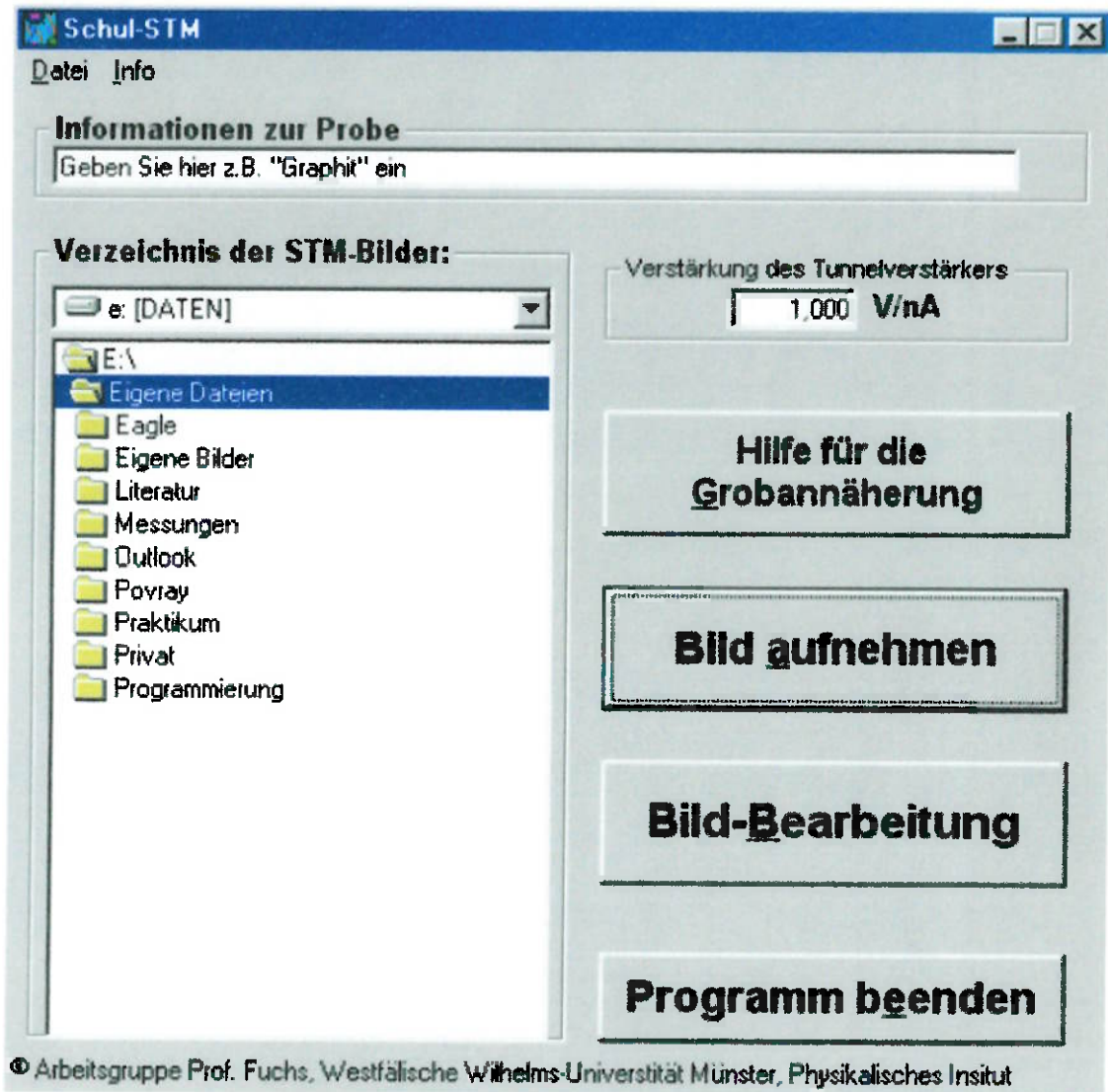
Caixa de informações





#### 4.8. *Main.frm*

É o programa principal



### 5. Módulos

#### 5.1. *API\_Interface*

*Public Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Long)*

Espera o tempo inserido, antes que o programa continue a executar suas tarefas.  
Durante o tempo de espera, nenhuma mensagem é processada

*Public Function GetSpecialFolder(RootFolder As ShellSpecialFolderConstants) As String*

Verifica no registro a ordem do sistema do caminho dado

*Public Sub Beep()*

Chama a função API Beep

## **5.2. Constantes**

*Public Const FileExt = ".STM"*

Extensão padrão do arquivo de dados

*Public Const MaxScanSize = 400*

Depende dos elementos de piezos. É a largura máxima de scan possível em nm

## **5.3. Módulo de Matemática**

As funções a seguir foram retirados de

William H. Press et al.: *Numerical Recipes, The Art of Science Computing (Fortran Version)*, Cambridge University Press

Para uma documentação detalhada, por favor consultar a fonte acima citada.

*Public Sub MedFit(x() As Single, y() As Single, a As Single, b As Single)*

[Página 554f]

*Public Sub Fit(x() As Single, y() As Single, a As Single, b As Single)*

[Página 508f]