

Universidade Estadual de Campinas UNICAMP

Instituto de Física Gleb Wataghin



F 530A - Instrumentação (2° semestre – 2012)

Relatório Final de Instrumentação

Aluno: Mariana Zavarize Nica RA: 094171 mariana.zavarizex<arroba>xhotmail.com Orientadora: Prof. Mônica A. Cotta monica x(arroba)x ifi.unicamp.br Eng. João Hermes Clerici Clerice x(arroba)x ifi.unicamp.br Coordenador: Prof. José J. Lunazzi

> Campinas 2012

Programação em LabView[®] para caracterização elétrica e calibração de biossensores

Aluna: Mariana Zavarize Nica RA: 094171

Orientadora: Prof. Mônica A. Cotta

Resumo

Neste projeto será desenvolvido um programa em LabView[®] para controle e aquisição de dados em medidas elétricas em dispositivos semicondutores.

Introdução

O trabalho apresentado neste relatório, realizado no Laboratório SPM do DFA/IFGW, tem como objetivo a automatização da aquisição de curvas de corrente-voltagem e resistência-tempo para aplicação no desenvolvimento de biossensores. A instrumentação utilizada é basicamente o *picoamperímetro Keithley 6487* e a estação de pontas *SemiProbe Lab Assistant*.

O programa de controle desenvolvido realiza a comunicação do computador com o picoamperímetro (através de uma placa GPIB), obtendo todas as informações que o usuário deseja para a realização das medidas (como o tipo de medida, voltagem, intervalo entre as medidas, intervalo de voltagem, etc) e realizando a medida de corrente para obtenção das curvas de interesse.

Uma das linhas de pesquisa do Laboratório SPM é sobre biossensores resistivos e fitopatógenos de plantas. O programa é utilizado para a caracterização elétrica desses sensores. Para isso é sempre necessário medir o tempo de resposta e corrente de saturação do dispositivo. No laboratório o *Keithley 6487* é usado para obter a variação da resistência em função do tempo e da concentração de um antígeno específico. A figura abaixo foi retirada de um artigo feito pelo grupo do laboratório e exemplifica a situação:



Figura 1: Apresentação esquemática da configuração do biossensor de InP e um método de funcionalização para a incorporação do receptor biomolecular que deve detectar a interação específico do ligante através de medição da mudança da resistência elétrica.

Desenvolvimento do Projeto

O objetivo do programa é obter curvas mostrando a variação da resistência de amostras semicondutoras com dois terminais (fonte/dreno), submetidas a uma tensão de porta associada à presença da ligação específica antígeno-anticorpo. Assim, trabalhamos com curvas da corrente em função da tensão e curvas da corrente em função do tempo. Para isso, será implementado o programa que realizará a comunicação do picoamperímetro *Keithley 6487* com o computador. A conexão entre o picoamperímetro e o computador é realizada através de uma placa GPIB.

O fabricante do picoamperímetro fonece alguns VI's (instrumentos virtuais) de rotina, como inicialização, leitura, entre outros que serão mostrados mais adiante. O programa utiliza esses VI's, porém foram necessárias modificações nos comandos para cada tipo de medida, como a formatação da leitura dos dados. Todo o programa, incluindo os subVI's contêm um parâmetro *Error In e Error Out*, estes parâmetros evitam que o programa continue a enviar comandos após algum erro ter ocorrido. O picoamperímetro retorna uma *string* como o valor de corrente, então é necessário converter essa *string* em um número, de preferência em notação científica, por isso todos os dados nos arquivos salvos estão em notação científica.

O *layout* do programa está representado na figura 2, e atende aos requisitos básicos de medida conforme discutido com os usuários da instrumentação no laboratório. A interface gráfica está em inglês pois o laboratório recebe estudantes e pesquisadores estrangeiros.



Figura 2: Layout geral do programa.



Figura 3: Diagrama de blocos geral do programa.

A imagem anterior (figura 1), ilustra o *layout* do programa, que possui várias opções de controle para o usuário. Na parte *"Single measurement"*, são realizadas medidas limitadas pelos seguintes parâmetros: *Ramp start* e *Ramp end* são os valores iniciais e final de tensão, *Ramp increment* é a razão com que se aumenta a tensão a cada

medida, *Ramp point delay* é o intervalo de tempo entre cada medida, *Voltage* representa a tensão a ser aplicada ao sistema para medidas no modo *Resistance*, *Repetitions* e *Repetitions delay* representam o número de vezes que o programa realizará o *loop* e o intervalo de tempo entre cada medida. A imagem a seguir representa o diagrama de blocos para a parte de *Single Measurement-Voltage Ramp*.



Figura 4: Diagrama de blocos para a função Voltage ramp no modo single measurement.

Na figura 4, podemos ver que após o o aparelho ser inicializado e fazer o Zero check, o programa pega os dados inseridos pelo usuário e primeiro define o valor de *n*, número de vezes que o *loop* será executado, ou seja, $n = \frac{Ramp \ end - Ramp \ start}{Ramp \ increment}$.

A seguir o loop começa a ser executado; nele definimos a voltagem como V =Ramp start + (Ramp increment x i). Isso está dentro de uma estrutura True/False para garantir que ele gere apenas n valores para a tensão onde i representa o número de iterações do loop. Portanto a cada repetição do loop temos uma nova voltagem e uma nova medida de corrente. Para medir a corrente o subVI Read é utilizado e então calcula-se a resistência, para isso dividimos o valor da voltagem pela corrente. Quando o loop chega a n repetições os dados são salvos através do subVI Save. Para plotar os gráficos são gerados arrays contendo os valores de corrente e tensão e outro de resistência e tempo. Os arrays mandam a informação para o gráfico através de Shift registers; eles transmitem os dados contidos nos arrays a cada iteração do loop. Já na passagem dos dados para a parte de armazenamento cada valor (corrente, tempo, voltagem, etc) é transmitido separadamente através dos Auto-Indexed tunnels; esses dados são transformados em um array que é enviado para o subVI Save. Para salvar o arquivo o programa oferece a opção automática (True) e também a manual (False), através de uma estrutura (Case structure). Caso o usuário selecione o modo manual (False), aparecerá uma mensagem na tela confirmando se ele realmente deseja salvar a medida através da função Two Button Dialog, e em caso positivo aparecerá uma caixa de diálogos para que se insira o nome do arquivo e local a ser salvo. Todas as medidas são salvas no formato .txt.

O processo de armazenamento de dados será mostrado mais adiante assim como as rotinas de inicialização e medida do aparelho.

Agora será mostrado o diagrama de blocos na função *resistance* no modo *Single Measurement*.



Figura 5: Diagrama de blocos para a função Resistance no modo Single Measurement.

Na figura 5, vemos que o número de repetições do *loop* já foi definido pelo usuário (*Cycles*) e a partir disso e uma voltagem fixa (*Voltage*) são feitas medidas de corrente a cada intervalo de tempo definido (*Cycle delay*). O programa cria um *array* com os valores de corrente e tensão e outro com os valores de resistência (divide-se a tensão, que no caso é fixa, pela corrente) e tempo para plotar os gráficos. Após realizar as n repetições o programa salvará os dados; para isso estes serão transmitidos separadamente pelos *Auto-Indexed tunnels* e um *array* será criado para o subVI *Save* ou *AutoSave*, que será determinado pela *Case structure* (modo *true ou false*).

Nas figuras a seguir serão exibidos os diagramas de blocos para o modo de medida contínua.



Figura 6: Bloco de diagramas para a função Voltage Ramp no modo de medida contínua.



Figura 7: Diagrama de blocos para a função *Resistance* no modo de medida contínua.



Figura 8: Diagrama de blocos para a função Voltage and Resistance no modo de medida contínua.

Nas figuras 6, 7 e 8 temos uma lógica muito parecida com o que foi feito no modo Single Measurement. Nessa parte o usuário pode definir os seguintes parâmetros: Voltage, Cycles, Cycle delay, que representam a voltagem aplicada, o número de vezes que o loop rodará e o intervalo de tempo entre cada medida, Time remaining e Actual cycle indicam ao usuário o tempo que falta para completar os ciclos e o ciclo em que o programa se encontra, respectivamente. O usuário seleciona uma voltagem (Voltage) fixa, então lê-se a corrente com o subVI Read e calcula-se a resistência (divide-se a tensão pela corrente). Em seguida cria-se os arrays, nesse caso no modo Voltage Ramp o programa plotará apenas um gráfico (IxV) e no modo Resistance (Rxt). O indicador Time remaining e Actual Cycles utilizam variáveis locais, que permitem que o ícone seja inserido em diversas partes do programa. Porém aparece apenas um indicador no painel de controle, o *Time remaining* é calculado da seguinte forma t = (n * n)Cycle Delay) – (i * Cycle Delay), onde n é o número total de repetições e i é o número de iterações do loop, no indicador Actual Cycle aparece simplesmente o valor de i. O processo de transmissão dos dados para o processo de armazenamento é o mesmo.

Como foi citado anteriormente e mostrado nas imagens o *picoamperímetro Keithley* 6487 possui alguns subVI's de rotina disponíveis pelo fabricante, como inicialização, checagem de zero, leitura etc. Esses procedimentos são muito importantes para o programa, pois são eles que fazem a comunicação do aparelho e computador, realizam as medidas etc. A seguir será mostrado os digramas de blocos das rotinas utilizadas e tambem dos VI's desenvolvidos para salvar os arquivos e plotar os dados já salvos na tela.

Rotina de inicialização:



Figura 9: Bloco de diagramas do subVI de inicialização.

Rotina de Zero check:



Figura 10: Diagrama de blocos do subVI Zero check; essa rotina garante que o picoamperímetro esteja sem nenhuma informação para começar as medidas.

Rotina de leitura:



Figura 11: Diagrama de blocos do subVI *read*; essa rotina faz a leitura da corrente (*string*) e converte para um número.

O subVI *read* apresentou certa dificuldade para a conversão da *string* para um número, pois foi necessário quebrar a *string* e depois trocar o "." por "," para que o programa reconhecesse o número em notação científica e depois realizasse os cálculos.



Rotina Voltage sweep:

Figura 12: Diagrama de blocos do subVI *Voltage Sweep* ele faz com que a fonte de tensão aplique a voltagem a ser utilizada.

SubVI Autosave:



Figura 13: SubVI Autosave.

O subVI Autosave, recebe um array contendo as medidas e com a função Array to Spreadsheet String converte os dados em strings e em seguida usa a função Concatenate strings para inserir os dados abaixo do cabeçalho. Após esses processos a função Write to text File é utilizada para salvar os arquivos; em seguida as funções Get Date e Format Date/Time são utilizadas para escrever o nome do arquivo que consiste basicamente da data e horário em que a medida foi feita e através da função Concatenate strings o nome do arquivo é salvo como o do exemplo: "Voltage Rampqui 13h46m15s", para salvar o arquivo no local desejado a função Build Path é utilizada.

SubVI Save:



Figura 14: SubVI Save.

O subVI *Save* pergunta ao usuário se deseja salvar os dados; em caso positivo, o subVI recebe um *array* contendo as medidas e com a função *Array to Spreadsheet String* converte os dados em *strings* e em seguida usa a função *Concatenate strings* para inserir os dados abaixo do cabeçalho. Após esses processos a função *Write to text File* é utilizada para salvar os arquivos e a função *File Dialog* para o usuário selecioanar o nome e local desejados.

IXV Rut	
Load and Reset IxV curve	Load and Reset Rxt curve

Figura 15: Diagrama de blocos para plotar dados já salvos no gráfico e também para limpar o gráfico.

Abaixo de cada gráfico existe a opção para plotar dados já armazenados no computador e também limpar o gráfico (botão *reset*); o diagrama de blocos dessas funções são mostrados na figura 15. Vemos que o programa lê um arquivo do computador através da função *Read from Measurement File* e então insere os valores no gráfico. Se o botão *Reset* for acionado ele manda um valor nulo e limpa o gráfico.

Agora que todos os diagramas de blocos foram apresentados e explicados, serão exibidas imagens dos arquivos salvos no modo manual e no autosave para os dois tipos de medida.

🖉 Voltage Ramp-qui 13h46 🗖 🗖 🔀	Continuous Voltage Ra
Arquivo Editar Formatar Exibir Ajuda	Arquivo Editar Formatar Exibir Ajuda
Voltage Ramp	Voltage Ramp
Time;Ampere;Voltage	Time;Ampere;Voltage
0,000E+0;3,002E-7;1,000E+0 1,027E+0;3,002E-7;1,500E+0 1,445E+0;6,004E-7;2,000E+0	0,000E+0;3,001E-7;1,000E+0 1,027E+0;3,001E-7;1,000E+0 1,446E+0;3,004E-7;1,000E+0 1,864E+0;3,005E-7;1,000E+0 2,283E+0;3,005E-7;1,000E+0 3,120E+0;3,001E-7;1,000E+0 3,539E+0;2,998E-7;1,000E+0 3,958E+0;2,998E-7;1,000E+0 4,376E+0;2,998E-7;1,000E+0

Figura 16: Arquivo com os dados coletados em formato .txt no modo *autosave* para as funções *single e continuous measurement.*



Figura 17: Mostra a janela que pergunta ao usuário se ele deseja salvar o arquivo.

🗍 1 - Bloco de notas 💷 🖾 🕅	🗍 2 - Bloco de notas 🛛 🗆 🖾
Arquivo Editar Formatar Exibir Ajuda	Arquivo Editar Formatar Exibir Ajuda
Voltage Ramp	Voltage Ramp
Time;Ampere;Voltage	Time; Ampere; Voltage
0,000E+0;3,002E-7;1,000E+0 1,027E+0;3,001E-7;1,500E+0 1,446E+0;6,004E-7;2,000E+0	0,000E+0;3,004E-7;1,000E+0 1,027E+0;3,003E-7;1,000E+0 1,445E+0;3,003E-7;1,000E+0 1,864E+0;3,002E-7;1,000E+0 2,282E+0;3,001E-7;1,000E+0 3,119E+0;3,000E-7;1,000E+0 3,538E+0;3,001E-7;1,000E+0 3,957E+0;3,001E-7;1,000E+0 4,375E+0;3,001E-7;1,000E+0
-	
E. (▲	. ► La

Figura 18: Arquivo com os dados coletados em formato .txt no modo manual para as funções Single e Continuous Measurement.



Figura 19: Foto do painel de controle com 2 gráficos plotados na função Single Measurement no modo Voltage Ramp.



A seguir serão exibidas imagens realizadas durante os testes:

Figura 20: Foto do computador conectado ao *picoamperímetro Keithley* 6487.



Figura 21: Foto do *picoamperímetro Keithley 6487* pronto para realizar medidas.



Figura 22: Foto da estação de pontas com uma amostra para medidas.

Os testes foram realizados com uma resistência fixa, porém no laboratório o picoamperímetro será acoplado a estação de pontas *SemiProbe Lab Assistant* para a realização das medidas. A *SemiProbe Lab Assistant possui* sistema de pontas

multifuncional adequado para aplicações que envolvem operações simples de sondagem ou que requerem a estabilidade do sistema e flexibilidade.

Referências

- http://www.keithley.com/products/dcac/voltagesource/application?mn= 6487
- http://www.keithley.com/products/dcac/voltagesource/application/?path =6487/Documents#6
- http://www.ni.com/labview/pt/
- Alberto L.D. Moreau, Richard Janissen, Clelton A. Santos, Luis A. Peroni, DagmarR.Stach Machado, Alessandra A.deSouza, AneteP.deSouza, MônicaA.Cotta, Highly-sensitive and label free InP biosensor for early phytopathogen diagnosis, Biosensors and Bioelectronics 36 (2012) 62–68 (Segue em anexo)
- http://www.semiprobe.com/our-solutions/lab-assistant

Opinião do orientador

Mariana fez um trabalho excelente. Para realizá-lo, estudou inicialmente o tipo de caracterização elétrica pretendido e, utilizando as necessidades apontadas pelos usuários do laboratório, implantou a interface gráfica e de comunicação entre os equipamentos. Para isso, estudou e aprendeu novas estruturas lógicas de programação ligadas ao LabView. Os testes realizados com resistores mostraram que a interface é funcional e que pode ser imediatamente utilizada com amostras reais.