

Introdução ao LabVIEW

Programação Gráfica



 **LabVIEW™**

Seminário Prático em 6 horas

ni.com

 **NATIONAL
INSTRUMENTS™**

Objetivos do Curso

- Se familiarizar com o ambiente LabVIEW e com execuções de fluxo de dados
- Habilidade para usar o LabVIEW para resolver problemas
- Conceitos do LabVIEW
 - Adquirir, salvar e carregar dados
 - Localizar e usar as funções math e complex analysis
 - Trabalhar com diferentes tipos de dados, como arrays e clusters
 - Demonstrar e imprimir resultados

ni.com



Esta é a lista de objetivos deste curso.

O curso prepara o aluno para fazer as seguintes coisas:

- Usar o LabVIEW para criar aplicações.
- Entender o painel frontal, diagrama de blocos, ícones e o painel de conexões.
- Usar as funções embutidas do LabVIEW.
- Criar e salvar programas no LabVIEW para usá-los em sub-rotinas.
- Criar aplicações que utilizam dispositivos DAQ.

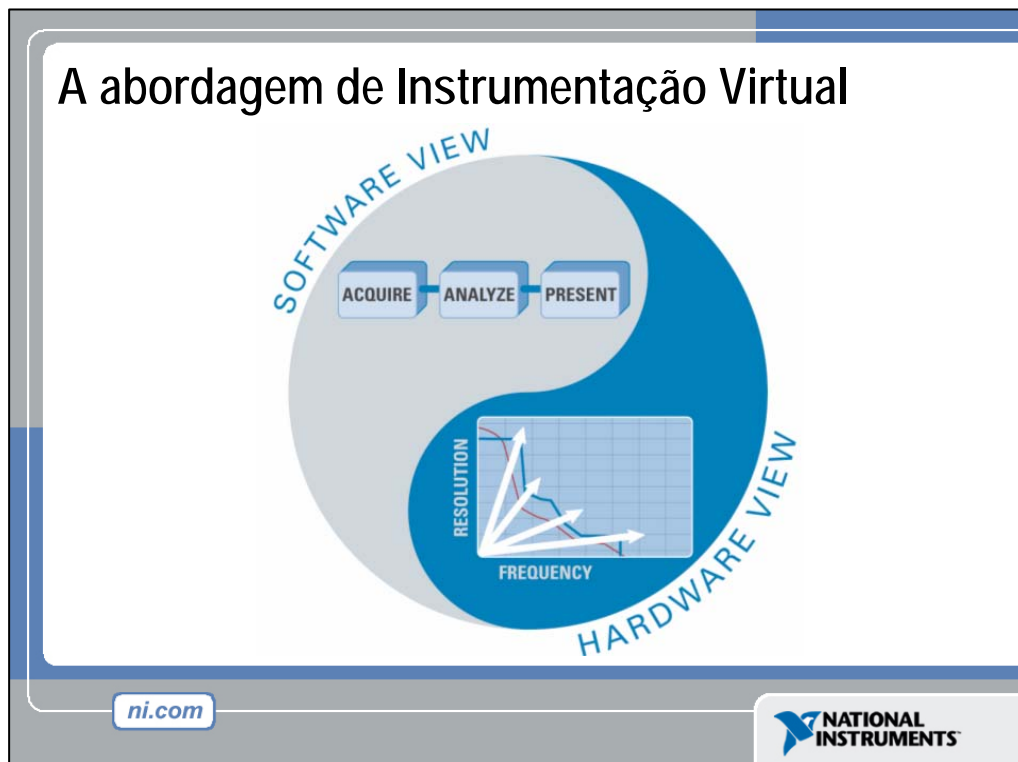
Este curso não descreve nenhum dos itens a seguir:

- Teoria de programação
- Cada objeto ou função embutida do LabVIEW
- Teoria de Analógico-para-Digital (A/D)

A NI providencia material de referência gratuito sobre os assuntos acima no site ni.com.

O Help do LabVIEW também é bastante útil:

LabVIEW»Help»Search the LabVIEW Help...



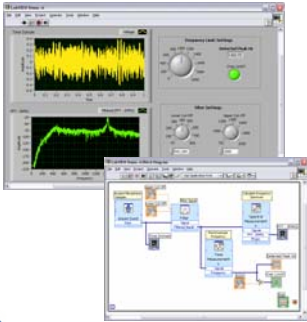
Instrumentação Virtual

Há mais de 25 anos, a National Instruments tem revolucionado o modo como engenheiros e cientistas na indústria, e em órgãos acadêmicos e governamentais encaram processos de medidas e automação. Alavancando PCs e tecnologia comercial, a instrumentação virtual aumenta a produtividade e reduz custos de teste, controle, e desenvolvimento de aplicações através de um software fácil de se integrar como o NI LabVIEW, e hardware de controle e medição modular para PXI, PCI, USB e Ethernet.

Com a instrumentação virtual, engenheiros usam um software de programação gráfica para criar soluções personalizadas para cobrir suas necessidades específicas, que é uma ótima alternativa para a instrumentação tradicional. Por exemplo, durante testes, medidas e controle, engenheiros têm usado a instrumentação virtual para reduzir o número de equipamentos de teste automatizados e aumentaram a produção em até 10 vezes com uma fração do custo de uma solução de instrumentação tradicional. No último ano 25.000 companhias em 90 países investiram em mais de 6 milhões de canais de instrumentação virtual da National Instruments.

Sistema de Desenvolvimento Gráfico LabVIEW

- Ambiente de Programação Gráfica
- Compile códigos para vários OS e dispositivos
- Útil em diversas aplicações



The image shows the LabVIEW graphical development environment. On the left, there are three windows: a waveform graph showing a yellow signal, a control panel with knobs and buttons, and a block diagram showing a complex flow of data and control logic. On the right, there is a table titled "LabVIEW Graphical Development Platform for Design, Control, and Test" which categorizes the software's capabilities.

LabVIEW Graphical Development Platform for Design, Control, and Test				
Embedded Design and Prototyping		Industrial Monitoring and Control		Automated Test and Measurement
Filter Design/DSP	Advanced Control	HMI/SCADA	Data Logging and NVH	Communications Test
System Prototyping	Industrial Control (PID)	Machine Vision and Motion	ATE	
Computing Targets				
Desktop	Industrial	Mobile	Embedded	

ni.com

NATIONAL INSTRUMENTS

O LabVIEW da National Instruments é um software de liderança-industrial para o desenvolvimento de testes, medições e controle de sistemas. Desde sua introdução em 1986, engenheiros e cientistas ao redor do mundo que confiaram no NI LabVIEW para desenvolver projetos em todas as etapas do ciclo de produção têm aumentado sua qualidade de produto, diminuído o tempo até o mercado, e aumentado a eficiência de engenharia e manufatura. Usando o ambiente integrado do LabVIEW para interfacear com sinais reais, analisar dados e extrair informações importantes e compartilhar resultados, você pode aumentar produtividade através de toda a sua organização. Como o LabVIEW tem uma linguagem de programação flexível combinado com ferramentas embutidas desenvolvidas especialmente para testes, medidas, e controle, você pode criar aplicações que variam de um simples monitoramento de temperatura até uma simulação sofisticada e sistemas de controle. Independente do seu projeto, o LabVIEW possui a ferramenta necessária para que você tenha sucesso rapidamente.

Aplicações de Instrumentação Virtual

- Design
 - Processamento de Sinais e Imagem
 - Programação de Sistema Embutido
 - (PC, DSP, FPGA, Microcontroladores)
 - Simulação e Prototipagem
 - E mais...
- Controle
 - Controles Automáticos e Sistemas Dinâmicos
 - Mecatronica e Robotica
 - E mais...
- Medidas
 - Circuitos e Electronica
 - Medidas e Instrumentação
 - E mais...

A single graphical development platform



ni.com

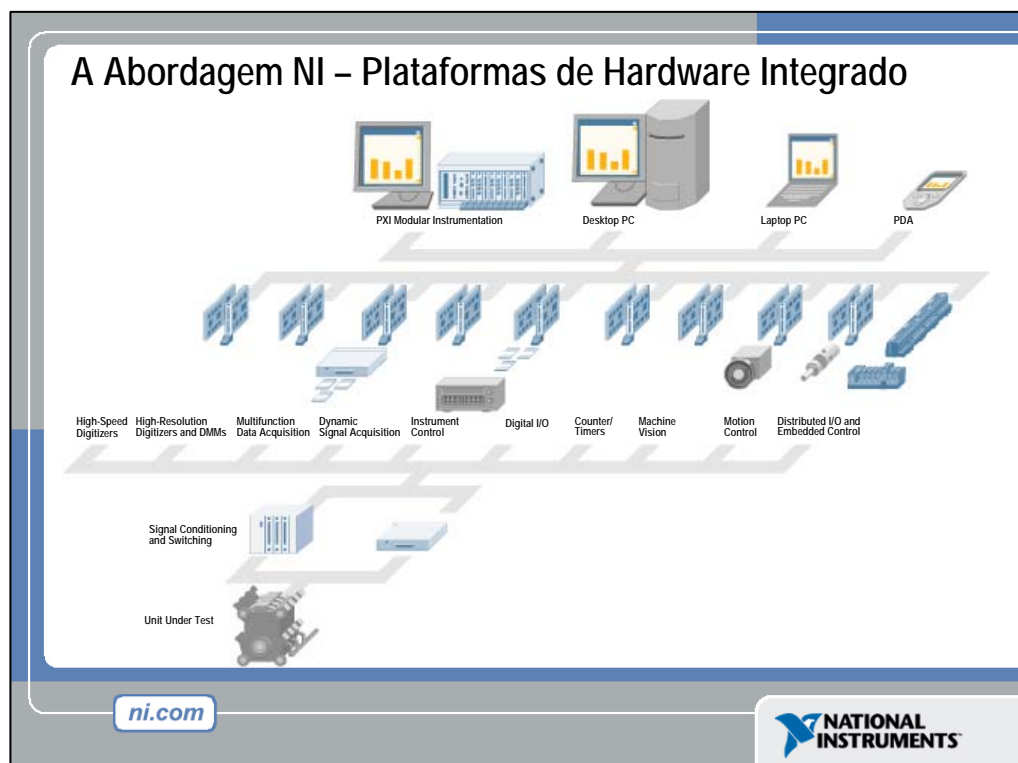
NATIONAL INSTRUMENTS

Aplicações de Instrumentos Virtuais

Instrumentação Virtual é aplicável em diversos ambientes, desde design até prototipagem e implementação. A plataforma LabVIEW traz ferramentas específicas para resolver aplicações variando de desenvolvimento algoritmos de processamento de sinais até fazer medidas de voltagem e pode extrair informação de diversas plataformas do seu desktop ou até dispositivos Embarcados - com um paradigma gráfico que é potente e intuitivo

Com a versão 8, o LabVIEW escala de design e implementação no PC para diversos alvos Embarcados, de protótipos robustos para sistemas Embarcados em chips. **O LabVIEW agiliza o desenvolvimento de sistemas com uma única plataforma de desenvolvimento gráfico.**

Assim, o LabVIEW garante um gerenciamento melhor de sistemas distribuídos em rede, pois na medida que os alvos para o LabVIEW se tornam mais variados e Embarcados, você precisará distribuir e comunica, entre várias pedaços de código LabVIEW em seu sistema.



Plataformas de Hardware integrado

Um instrumento virtual consiste em um computador de padrão industrial ou estação de trabalho equipado com um software de aplicação potente, hardware de “custo-efetivo” como placas plug-in, e drivers de software, que em conjunto fazem a função de instrumentos tradicionais

Instrumentos Virtuais representam uma mudança fundamental de instrumentação tradicional centrada em hardware para sistemas centrados em software que exploram a potencia computacional, display, e capacidade de conectividade de computadores populares e estações de trabalho.

Mesmo o PC e tecnologia de circuitos integrados terem passado por avanços significativos nas últimas duas décadas, o software realmente oferece a flexibilidade para crescer nesta fundação de hardware e criar instrumentos virtuais, trazendo maneiras melhores para inovar e reduzir custos significativamente. Com instrumentos virtuais, engenheiros e cientistas constroem sistemas de medidas e automação que satisfazem exatamente suas necessidades (definido pelo usuário) em vez de ser limitado por instrumentos tradicionais de função-fixa (definido pelo vendedor).

Seção I – Ambiente LabVIEW

A. Colocando dados no seu Computador

- Dispositivos de Aquisição de Dados
 - NI-DAQ
 - Aquisição de Dados Simulados
 - Placa de Som

B. Ambiente LabVIEW

- Painel Frontal / Diagrama de Blocos
- Barra de Ferramentas / Paleta de Ferramentas

C. Componentes de uma Aplicação LabVIEW

- Creating a VI
- Data Flow Execution

D. Ajuda Adicional

- Encontrando Funções
- Dicas para trabalhar com o LabVIEW

ni.com



A. Configurando seu Hardware

- Track A**
 - Dispositivos de Aquisição de Dados (DAQ)
 - Dispositivos USB, PCI ou PXI verdadeiros
 - Configurados no MAX
- Track B**
 - Dispositivo de Aquisição de Dados Simulado (DAQ)
 - Software simulado no nível do driver
 - Configurado no MAX
- Track C**
 - Placa de Som
 - Localizada dentro da maioria dos computadores



ni.com

NATIONAL INSTRUMENTS

Este curso de LabVIEW foi desenvolvido para ser utilizado por pessoas com ou sem acesso à hardware da National Instruments.

Cada exercício é dividido em tres partes (tracks, A, B, e C):

Track A foi desenvolvido para ser usado com hardware suportado pelo DAQmx da National Instruments. Isto inclui a maioria de dispositivos de aquisição de dados do tipo USB, PCI, PXI com entradas analógicas. Alguns condicionamentos de sinais e excitações requerem o uso de um microfone com o dispositivo DAQ (alguns sensores, como microfones, requerem potência externa para funcionar (excitação)).

Track B foi desenvolvido para ser usado sem hardware. O hardware pode ser simulado com o DAQmx Driver versão 7.5 ou mais nova. Um dispositivo DAQmx simulado é uma réplica do dispositivo criado usando a opção NI-DAQmx Simulated Device no menu Create New do MAX para o proposito de operar uma função ou programa sem o hardware. Um dispositivo simulado age de forma parecida a dispositivo verdadeiro. Seus drivers são carregados, e o programa utilizando-o é verificado completamentei

Track C foi desenvolvido para ser usado com uma placa de som e um microfone padrão. O LabVIEW inclui VIs simples para trabalhar com entradas e saídas analógicas usando a placa de som que vem embutida na maioria dos computadores. (Isto é muito conveniente uma vez que na maioria dos laptops a placa de som e o microfone já vem Embarcados).

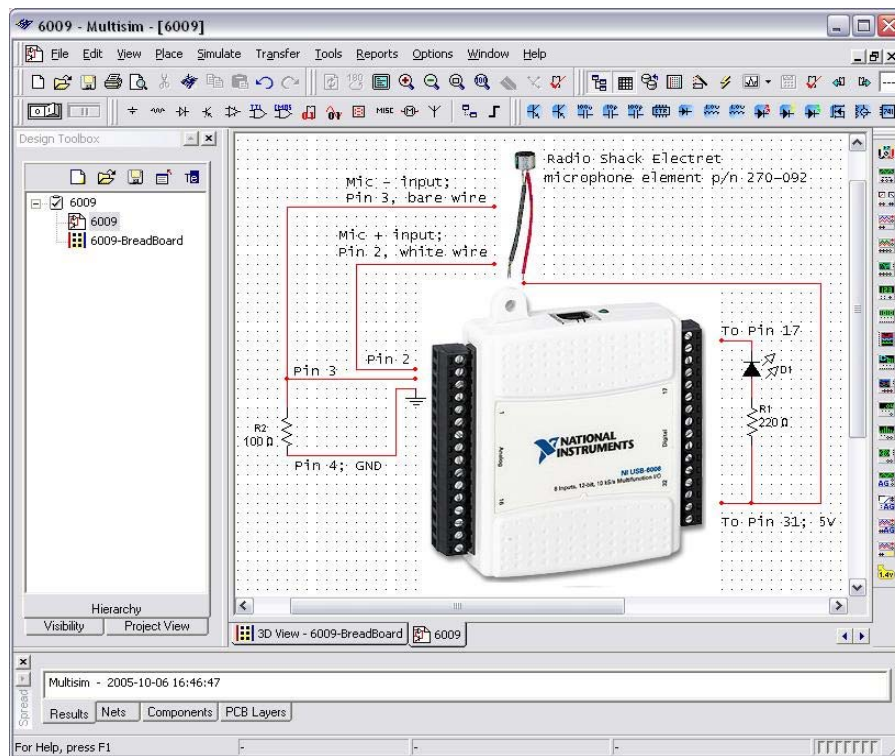
Configurando seu Hardware para o Track selecionado

Track A – NI Data Acquisition com Microfone: USB-6009 com Microphone & LED

Hardware Sugerido:

Qty	Part Number	Description	Supplier
1	779321-22	Low-cost USB DAQ	National Instruments
1	270-092	Electret Microphone	RadioShack
1		100 Ohm Resistor	RadioShack
1		220 Ohm Resistor	RadioShack
1	276-307	Light Emitting Diode (LED)	RadioShack

O seguinte é uma esquemática desenhada com **Multisim**, uma esquemática SPICE altamente usada e ferramenta de simulação. Visite <http://www.electronicworkbench.com> para mais informações.



Track A – Aquisição Simulada de dados NI: Software NI-DAQ versão 8.0 ou mais novo

Track C – Placa de Som: Placa de Som eMicrofone

Hardware Sugerido:

Qty	Part Number	Description	Supplier
1		Standard Plug-in PC Microphone*	RadioShack

* Laptops normalmente têm um microfone embutido (um microfone externo não é necessário)

Que tipo de dispositivo devo usar?



	Sound Card*	NI USB DAQ	NI PCI DAQ	Instruments*
AI Bandwidth	8–44 KS/s	10–200 KS/s	250 K–1.2 Ms/s	20kS/s–2 GS/s
Accuracy	12–16 bit	12–16 bit	14–18 bit	12–24 bit
Portable	x	x	—	some
AI Channels	2	8–16	16–80	2
AO Channels	2	1–2	2–4	0
AC or DC	AC	AC/DC	AC/DC	AC/DC
Triggering	—	x	x	x
Calibrated	—	x	x	x

ni.com



* The above table may not be representative of all device variations that exist in each category

Que dispositivo devo usar?

Há diversos tipos de dispositivos de aquisição de dados e controle disponíveis no mercado. Alguns estão destacados acima. Escolha normalmente cai entre taxa amostragem (samples/segundo), resolução (bits), número de canais, e taxa de transferência de dados (normalmente limitado por tipo de barramento: USB, PCI, PXI, etc.). Dispositivos multifuncionais podem ser usados em uma variedade de aplicações.

USB-6008 & USB-6009 DAQ USB de baixo-custo

O USB-6009 da National Instruments traz funções básicas de aquisição de dados para aplicações como armazenamento de dados, medidas portáteis, experimentos em laboratórios acadêmicos. O NI USB-6008 e NI USB-6009 são ideais para estudantes. Crie suas próprias aplicações de medidas programando o NI USB-6009 com o LabVIEW e o NI-DAQmx driver software para Windows. Usuários de Mac OS X e Linux podem baixar e usar o NI-DAQmx Base driver.

NI USB-6009 Specifications:

- Oito entradas analógicas de 14-bit
- 12 linhas de E/S digital
 - 2 saídas analógicas
 - 1 contador



<http://www.ni.com/daq/>

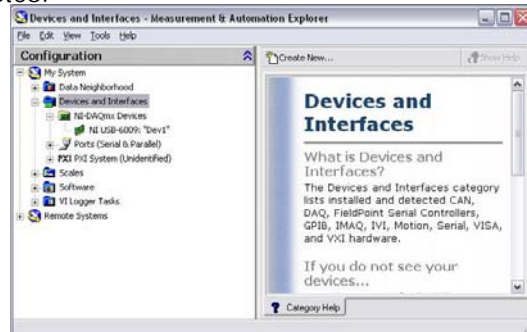
O que é MAX?

- MAX significa Measurement & Automation Explorer.
- MAX configura e organiza todos seus dispositivos DAQ, PCI/PXI instruments, GPIB, IMAQ, IVI, Motion, VISA, e VXI da National Instruments.
- Usado para configuração e testes.

Ícone Found on
Windows Desktop



Measurement
& Automation



ni.com



O próximo nível de software que trataremos chama-se Measurement & Automation Explorer (MAX). MAX é uma interface de software que lhe proporciona acesso a todos os seus dispositivos DAQ, GPIB, IMAQ, IVI, Motion, VISA, e VXI da National Instruments. O atalho para o MAX será colocado na sua área de trabalho após a instalação. Uma foto do ícone está acima. MAX é usado principalmente para configurar e testar seu hardware da National Instruments, mas ele oferece outras funções como checar se você tem a última versão do NI-DAQ instalado. Quando você executa uma aplicação usando o NI-DAQmx, o software lê as configurações do MAX e determina os dispositivos que você configurou. Assim, você deve configurar o DAQ antes com o MAX.

A funcionalidade do MAX é dividido em sete partes:

- Data Neighborhood
- Devices and Interfaces
- IVI Instruments
- Scales
- Historical Data
- Software
- VI Logger Tasks

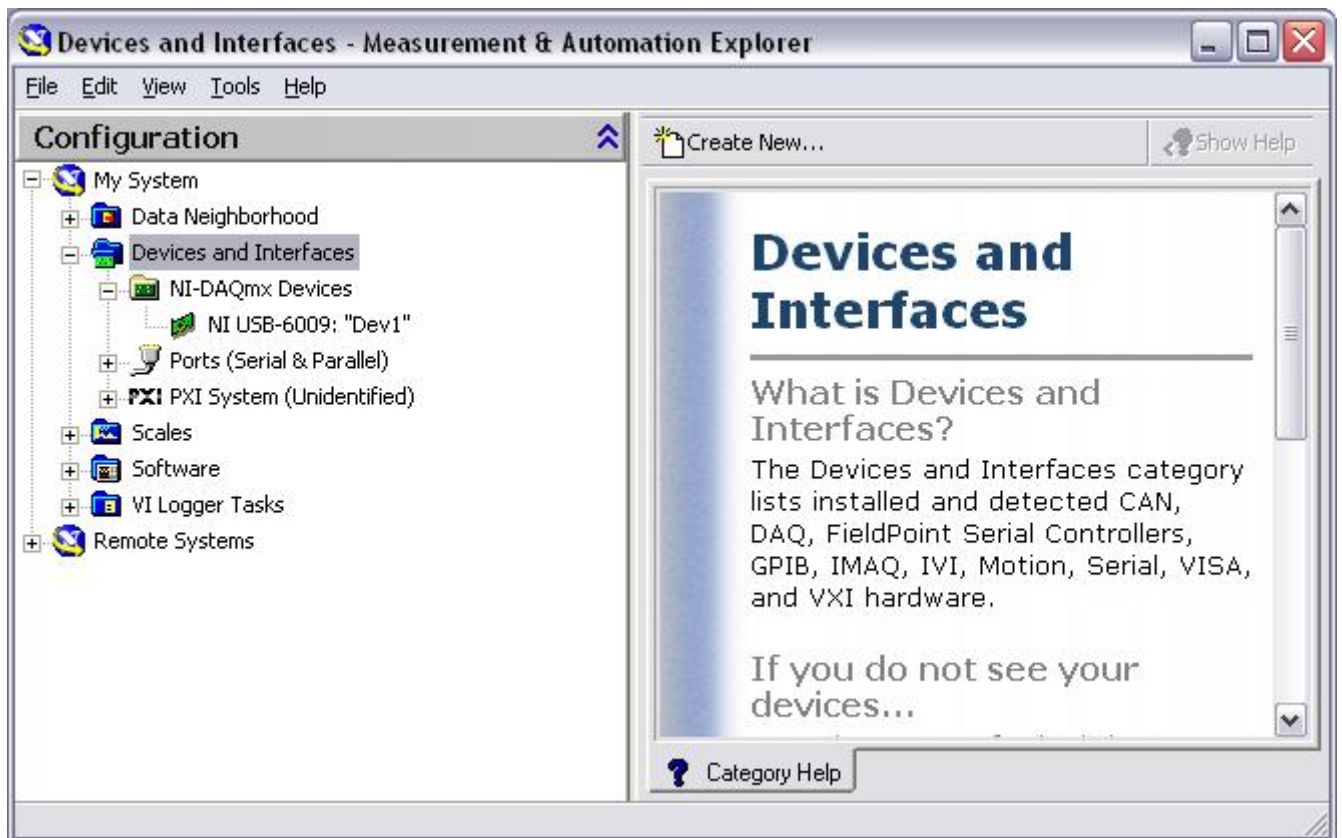
Para este curso, focaremos no Data Neighborhood, Devices and Interfaces, Scales, e Software. Passaremos por cada uma destas categorias e aprenderemos sobre as funções que cada uma oferece.



Exercício 1.1 – Testando seu dispositivo (Track A)

Neste exercício você usará o Measurement and Automation Explorer (MAX) para testar seu dispositivo DAQ NI USB-6009.

1. Abra o MAX clicando duas vezes no ícone no desktop ou selecionando **Start»Programs»National Instruments»Measurement & Automation**.
2. Expanda a seção **Devices and Interfaces** para ver os dispositivos National Instruments que estão instalados. O MAX mostra hardware e software da National Instruments que está instalado em seu computador.
3. Expanda a seção **NI-DAQmx Devices** para ver o hardware compatível com NI-DAQmx. O número do dispositivo aparece entre aspas seguido do nome. Os VIs de aquisição de dados usam este número para determinar que dispositivo DAQ realiza a operação. Você verá seu hardware listado como NI USB-6009: “Dev1”.
4. Realize um self-test no dispositivo clicando com o botão direito do mouse na árvore de configuração e selecionando **Self-Test** ou clicando em “Self-Test” no topo da janela. Isto testa os recursos do sistema atribuídos a este dispositivo. O dispositivo deve passar no teste se ele já estiver configurado.
5. Cheque a pinagem de seu dispositivo. Clique com o botão direito do mouse no dispositivo na árvore de configuração e selecione **Device Pinouts** ou clique em “Device Pinouts” que se encontra no topo central da janela.
6. Abra o painel de testes. Clique com o botão direito do mouse na árvore de configuração do dispositivo e selecione **Test Panels...** ou clique em “Test Panels...” localizado no topo central da janela. O painel de testes permite que você teste as funções de seu dispositivo, entrada/saída analógica, entrada/saída digital, e contadores sem fazer nenhuma programação.
7. Na aba **Analog Input** do painel de testes, mude o **Mode** para “Continuous” e **Rate** para 10,000 Hz. Clique “Start” e faça um zumbido ou assobie no microfone para observar que um sinal é plotado. Clique em “Finish” quando você tiver terminado.
8. Na aba **Digital I/O** veja que inicialmente a porta está configurada para entrada somente. Observe debaixo do **Select State** os LEDs que representam o estado de linhas de entrada. Clique no botão “All Output” em **Select Direction**. Veja que você agora tem booleanos no **Select State** para especificar um estado de saída de diferentes linhas. Altere linha 0 e veja o LED ascender. Clique “Close” para fechar o painel de testes.
9. Feche o MAX.



(Fim do Exercício)

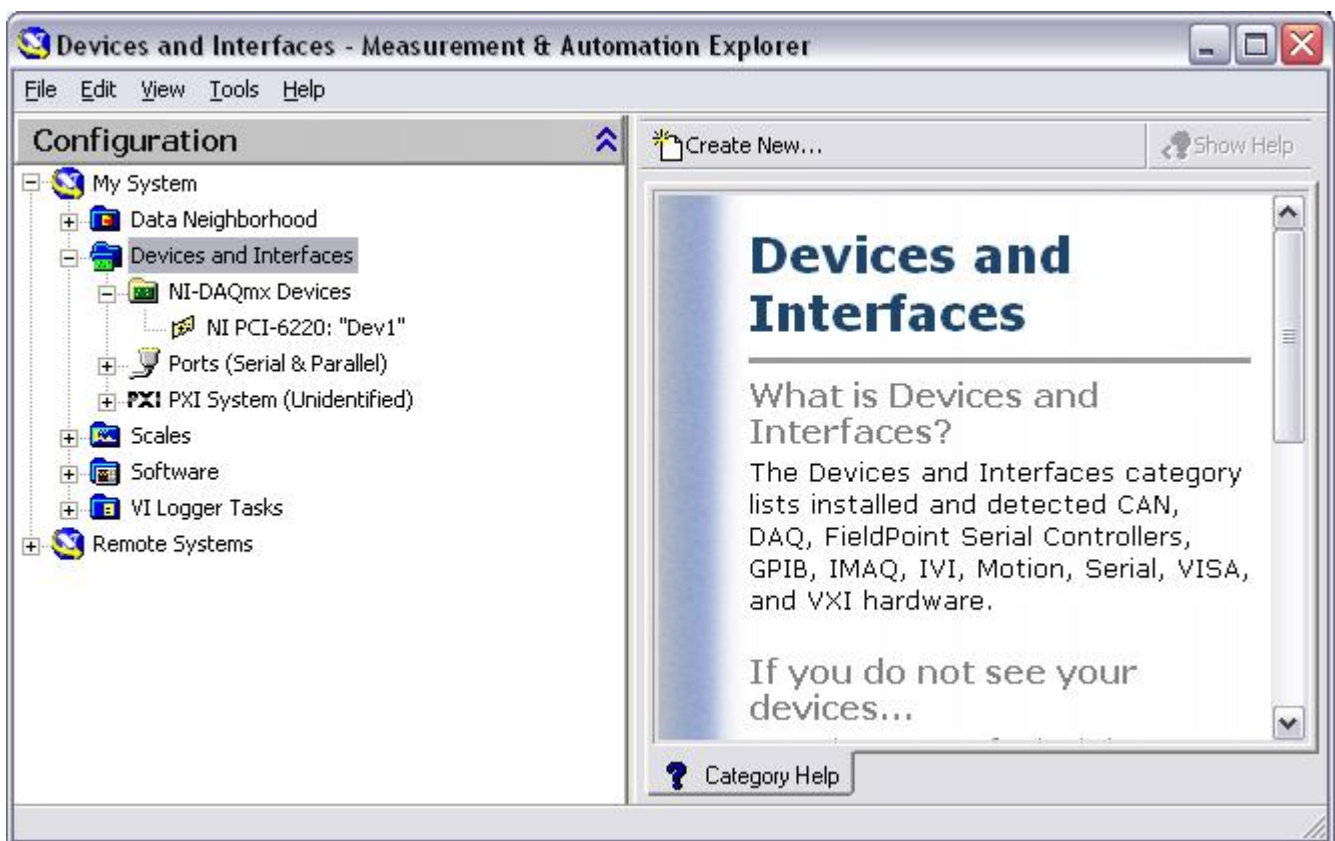


Exercício 1.1 – Configurando seu dispositivo (Track B)

Neste exercício você usará o Measurement and Automation Explorer (MAX) para configurar um dispositivo DAQ simulado.

1. Abra o MAX clicando duas vezes no ícone no desktop ou selecionando **Start»Programs»National Instruments»Measurement & Automation**.
 2. Expanda a seção **Devices and Interfaces** para ver os dispositivos National Instruments que estão instalados. O MAX mostra hardware e software da National Instruments que está instalado em seu computador. O número do dispositivo aparece entre aspas seguido do nome. Os Vis de aquisição de dados usam o número do dispositivo para determinar que dispositivo realiza a operação DAQ.
 3. Crie um dispositivo DAQ simulado para ser usado mais tarde neste curso. Dispositivos simulados são ferramentas potentes para o desenvolvimento sem ter o hardware fisicamente instalado no seu computador. Clique com o botão direito do mouse em **Devices and Interfaces** e selecione **Create New...»NI-DAQmx Simulated Device**. Clique “Finish”.
 4. Expanda a seção M Series DAQ. Selecione o **PCI-6220** ou qualquer outro dispositivo PCI e clique “OK”.
 5. Os dispositivos da pasta NI-DAQmx serão expandidos e você verá uma nova entrada para o PCI-6220: “Dev1”. Você criou um dispositivo simulado!
- Realize um self-test no dispositivo clicando com o botão direito do mouse na árvore de configuração e selecionando **Self-Test** ou clicando em “Self-Test” no topo da janela. Isto testa os recursos do sistema atribuídos a este dispositivo. O dispositivo deve passar no teste se ele já estiver configurado.
 - Cheque a pinagem de seu dispositivo. Clique com o botão direito do mouse no dispositivo na árvore de configuração e selecione **Device Pinouts** ou clique em “Device Pinouts” que se encontra no topo central da janela.
 - Abra o painel de testes. Clique com o botão direito do mouse na árvore de configuração do dispositivo e selecione **Test Panels...** ou clique em “Test Panels...” localizado no topo central da janela. O painel de testes permite que você teste as funções de seu dispositivo, entrada/saída analógica, entrada/saída digital, e contadores sem fazer nenhuma programação.
 - Na aba **Analog Input** do painel de testes, mude o **Mode** para “Continuous”. Clique “Start” e observe o sinal que é plotado. Clique “Stop” quando você tiver terminado.

10. Na aba **Digital I/O** veja que inicialmente todas as portas estão configuradas para entrada. Observe os LEDs abaixo do **Select State** que representam o estado das linhas de entrada. Clique no botão “All Output” em **Select Direction**. Veja que agora você tem booleanos abaixo do **Select State** para especificar o estado das diferentes linhas. Clique “Close” para fechar o painel de testes.
11. Feche o MAX.



(Fim do Exercício)



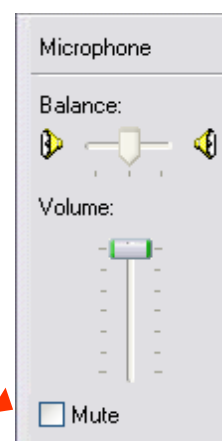
Exercício 1.1 – Setting Up Your Device (Track C)

Neste exercício, você irá usar as utilidades do Windows para verificar sua placa de som e usa-la com um microfone.

1. Prepare seu microfone para uso. Dê um clique duplo no controle de volume e abra a janela de configuração. A configuração de som também pode ser encontrada no painel de controle do Windows: **Start Menu»Control Panel»Sounds and Audio Devices»Advanced**.
2. Se você não ver uma seção para o microfone vá para **Options»Properties»Recording** e coloque um cheque na caixa **Microphone**. Isto mostrará o microfone no volume de controle. Clique “OK”.
3. Des-selecione a caixa **Mute** se ela ainda estiver selecionada. Verifique que o volume esta em uma altura desejável.



Uncheck Mute





4. Feche a janela de configuração do volume.
5. Abra o gravador de som selecionando **Start»Programs»Accessories»Entertainment»Sound Recorder**.
6. Clique no botão de gravação e fale em seu microfone. Veja que o sinal do som é mostrado no Sound Recorder.
7. Clique stop e feche o Sound Recorder sem salvar mudanças quando você tiver terminado.



(Fim do Exercício)

Abra e Execute o LabVIEW

Start»All Programs»National Instruments LabVIEW

 »  National Instruments LabVIEW 8.0

Tela Inicial:

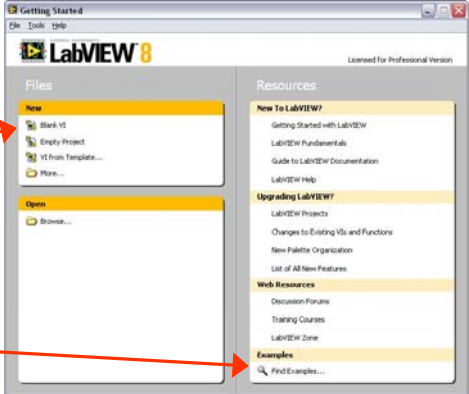
Comece com uma VI em branco:

New»Blank VI

OU

Comece a partir de um Exemplo:

Examples»Find Examples...



ni.com

NATIONAL INSTRUMENTS

LabVIEW

LabVIEW é uma linguagem de programação gráfica que usa ícones em vez de linhas de texto para criar aplicações. Em contraste com linguagens de programação baseadas em texto, onde instruções determinam a execução do programa, o LabVIEW usa programação de dataflow, onde o fluxo de dados determina a ordem de execução. Você pode adquirir diversos toolkits adicionais para desenvolver aplicações especializadas. Todos os toolkits se integram perfeitamente com o LabVIEW. Veja o site da National Instruments para mais informações sobre toolkits.

LabVIEW também traz vários wizards para ajudar você a configurar seu dispositivo DAQ e instrumentos baseados em computadores e criar aplicações.

LabVIEW Example Finder

O LabVIEW traz centenas de exemplos de VIs que você pode usar e incorporar em VIs que você criar. Além de exemplos de VIs que vêm com o LabVIEW, você pode acessar centenas de outros exemplos no NI Developer Zone (zone.ni.com). Você pode modificar um exemplo para se ajustar à sua aplicação, ou copiar e colar um ou mais exemplos na VI que você criar.

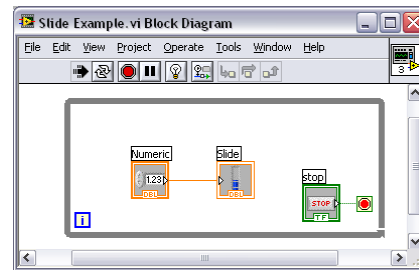
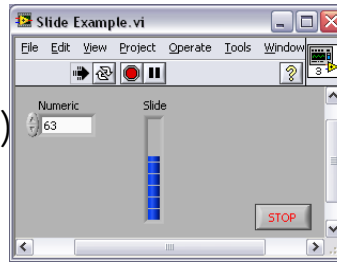
Programas LabVIEW são chamados de Virtual Instruments (VIs)

Each VI tem 2 Janelas
Painel Frontal

- Interface de Usuário (UI)
 - Controles = Inputs
 - Indicadores = Outputs

Diagrama de Blocos

- Código Gráfico
 - Dados passam em fios dos controles, através de funções até os indicadores
 - Blocos executados por Dataflow



ni.com

NATIONAL
INSTRUMENTS

Os programas do LabVIEW são chamados de virtual instruments (VIs).

Controles são entradas e indicadores são saídas.

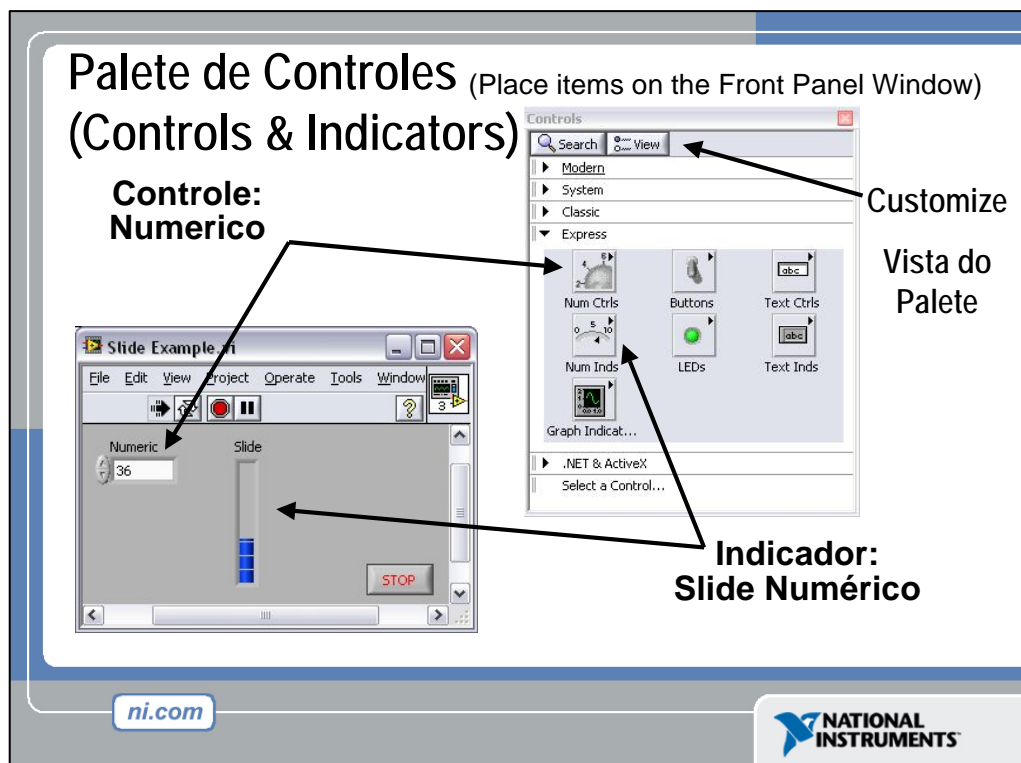
Cada VI contém três partes principais:

- Painel Frontal – Onde o usuário interage com o VI.
- Diagrama de Blocos – O código que controla o programa.
- Ícones/Conectores – Meio de conexão entre VIs.

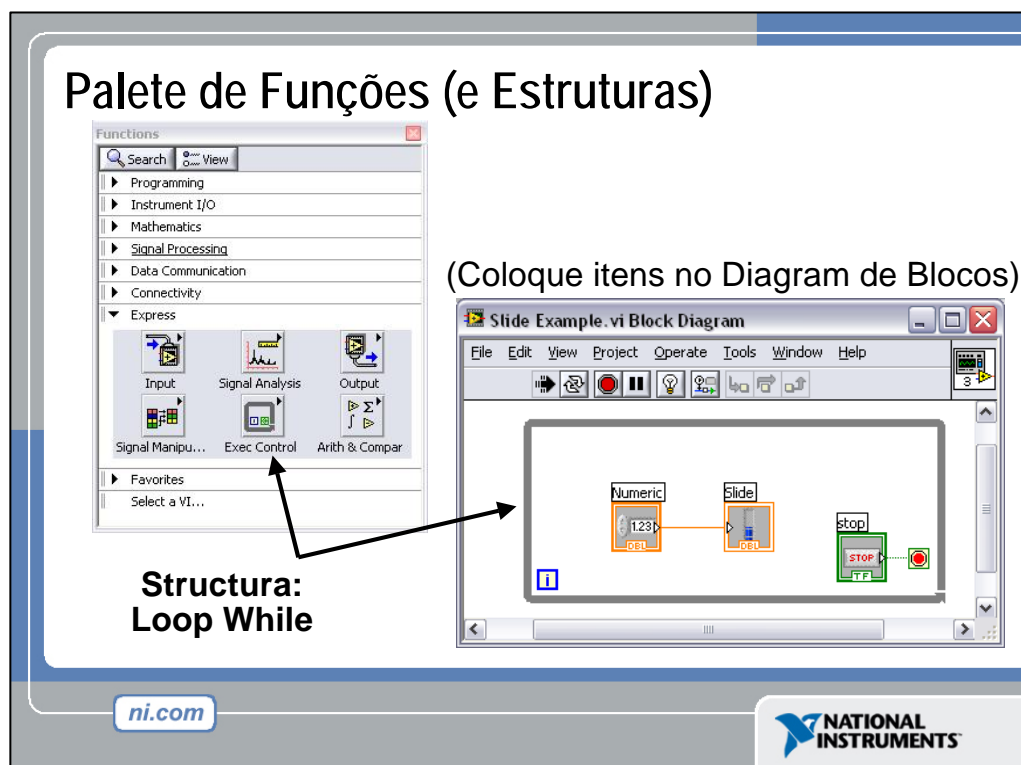
No LabVIEW, você pode construir uma interface de usuário usando um conjunto de ferramentas e objetos. A interface de usuário é conhecida como o painel frontal. Você pode acrescentar código usando representações gráficas de funções para controlar os objetos do painel frontal. O diagrama de blocos contém este código. De certa forma o diagrama de blocos se parece que um fluxograma.

Usuários interagem com o Painel Frontal quando o programa está em execução. Usuários podem controlar o programa, mudar entradas, e ver os dados serem atualizados em tempo real. Controles são usados para entradas para, ajustar uma barra de controle para configurar um valor para um alarme, ligar ou desligar um interruptor, ou parar o programa. Indicadores são saídas. Termômetros, luzes, e outros indicadores mostram os valores de saída do programa. Estes podem incluir dados, estados do programa, e outras informações.

Cada controle ou indicador do painel frontal tem um terminal correspondente no diagrama de blocos. Quando uma VI é executada, valores dos controles atravessam o diagrama de blocos, onde eles são usados em funções e o resultado é passado para outras funções através de fios.




Use o paleta **Controls** para colocar controles e indicadores no painel frontal. O paleta **Controls** está disponível somente no painel frontal. Para ver este paleta selecione **Window»Show Controls Palette**. Você pode visualizar o paleta **Controls** clicando com o botão direito do mouse em uma área aberta do painel frontal. Fixe o paleta **Controls** clicando no botão localizado no canto superior esquerdo do do paleta.



Use o palette **Functions** para criar seus diagrama de blocos. O palette **Functions** esta disponível somente no diagrama de blocos. Para ver o o palette, selecione **Window»Show Functions Palette**. Você pode visualizar o palette **Functions** clicando com o botão direito do mouse em uma área aberta do diagrama de blocos. Fixe o palette e **Functions** clicando no botão localizado no canto superiro esquerdo do palette.





Paleta de Ferramentas





- Recomendado: Ferramenta de Seleção Automática
- Ferramentas para operar e modificar objetos tanto do painel frontal quanto o diagrama de blocos

Automatic Selection Tool

Escolhe Automaticamente entre as ferramentas:

-  **Operating Tool**
-  **Positioning/Resizing Tool**
-  **Labeling Tool**
-  **Wiring Tool**

Se a ferramenta de seleção automática está ligada e você passa seu cursor sobre objetos do painel frontal e diagrama de blocos, o LabVIEW automaticamente seleciona a ferramenta correspondente do paleta **Tools**. Alterne entre seleção automática selecionado **Automatic Tool Selection** no paleta **Tools**.

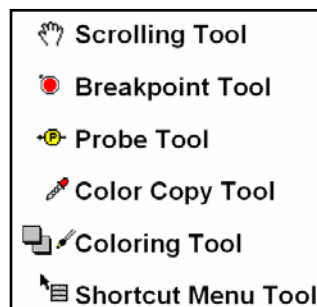
Use a **Operating tool** para mudar valores de controles ou selecionar texto dentro de um controle.

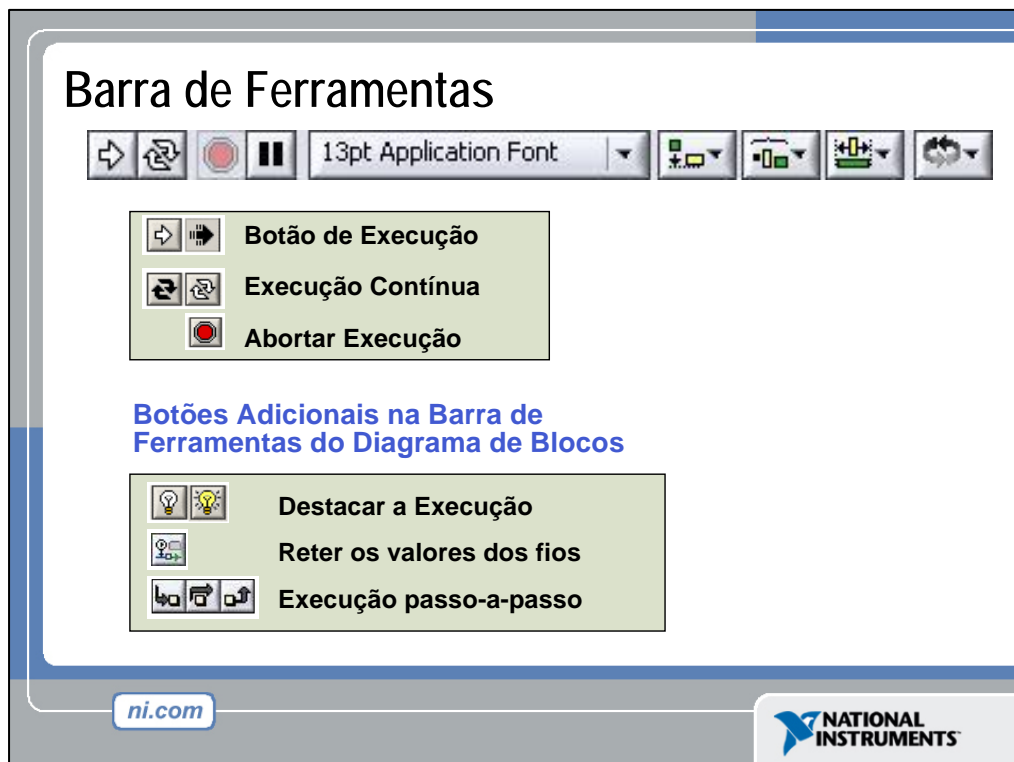
Use a **Positioning tool** para selecionar mover ou redimensionar objetos. A ferramenta de posicionamento muda de formato na quina de um objeto redimensionável.

Use a **Labeling tool** para editar textos e criar rótulos livres. A ferramenta de edição de texto muda para um cursor quando você tem um rótulo livre.

Use a **Wiring tool** para ligar objetos no diagrama de blocos.

Outras ferramentas importantes:





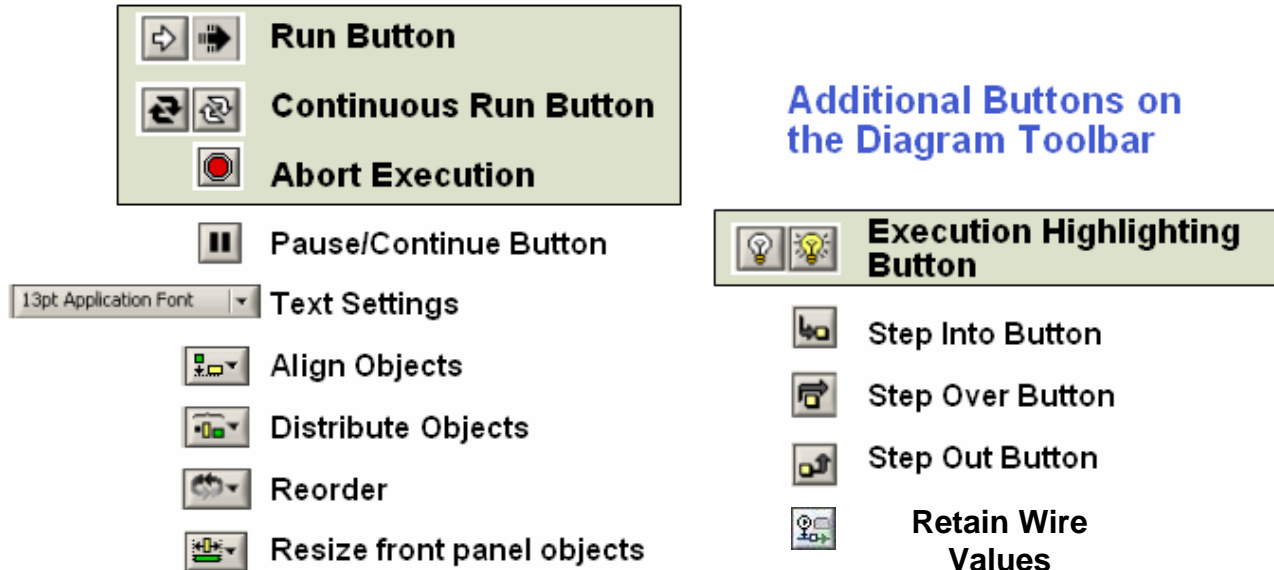
- Clique no botão **Run** para executar o VI. Enquanto o VI executa, o botão Run aparece como uma seta preta se o VI for um VI top-level, significando que ele não é um subVI.
 - Clique no botão **Continuous Run** para executar o VI até que você aborte ou pause o mesmo. Você pode clicar no botão novamente para desabilitar a execução contínua.
 - Enquanto o VI estiver em execução, O botão **Abort Execution** aparece. Clique neste botão para parar o VI imediatamente.
- Nota:** Evite usar o botão **Abort Execution** para parar um VI. Ou deixe o vi completar seu fluxo de dados ou desenvolva um método para parar o VI através do programa. Assim, o VI esta em um estado conhecido. Por exemplo, cloque um botão no painel frontal que para o VI quando este for acionado.
- Clique no botão **Pause** button para pausar um VI em execução. Quando você clica no botão **Pause**, oLabVIEW destaca a área do diagrama de blocos onde foi parada a execução. Clique novamente em **Pause** para continuar com a execução do VI.
 - Selecione o **Text Settings** e mude as configurações da fonte do VI, incluindo tamanho, estilo e cor.
 - Selecione **Align Objects** para alinhar objetos ao longo de eixos, incluindo vertical, topo, esquerdo, etc..
 - Selecione **Distribute Objects** para espaçar objetos homogeneamente incluindo espaços, compreensão,etc.
 - Selecione **Resize Objects** para alterar e largura de objetos no painel frontal

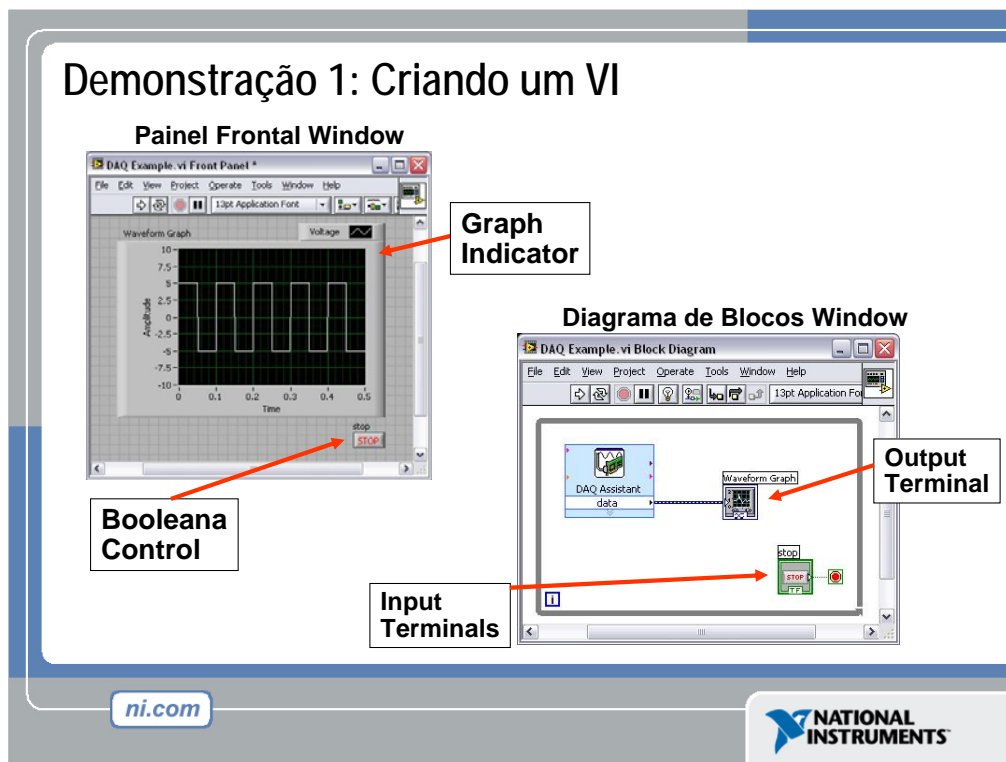
- Selecione o **Reorder** quando você tem objetos que ficam sobrepostos e voce quer definir qual fica na frente. Selecione um destes objetos com a ferramenta de posicionamento e selecione **Move Forward**, **Move Backward**, **Move To Front**, ou **Move To Back**.

Nota: Os seguintes itens aparecem somente no diagrama de blocos.

- Clique no botão **Highlight Execution** para ver o fluxo de dados no diagrama de blocos. Clique no botão novamente para desabilitar esta execução.
- Clique no botão **Retain Wire Values** para salvar o valor do fio em cada ponto no fluxo da execução para que quando você colocar um probe no fio, ele imediatamente mostrará o valor mais recente que passou pelo fio.
- Clique no botão **Step Into** para dar um único passo dentro do loop, subVI, etc. Single-stepping (execução passo a passo) de um VI faz uma execução nó a nó. Cada nó pisca quando este estiver pronto para ser executado.
- Clique no botão **Step Over** para passar por cima de um loop, SubVI, etc. Quando você utiliza o step over, você executa o nó sem a execução passo a passo daquele item.
- Clique no botão **Step Out** para sair de um loop, SubVI, etc. Clicando no step out do nó, você completa uma execução passo a passo do nó e passa para o próximo.

Ferramentas Adicionais:





Quando você cria um objeto no Painel Frontal, um terminal será criado no Diagrama de Blocos. Estes terminais lhe dão acesso aos objetos do Painel Frontal a partir do código do Diagrama de Blocos.

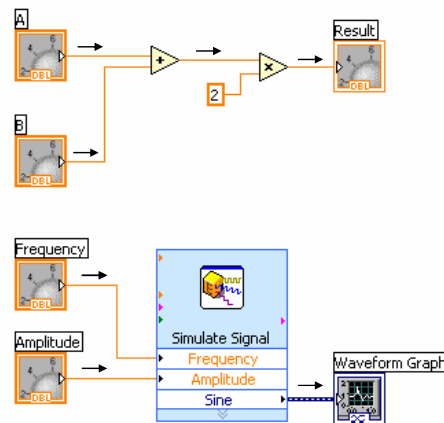
Cada terminal contém informações úteis sobre os objetos correspondentes do Painel Frontal. Por exemplo, a cor e símbolo trazem informação a respeito do tipo de dado. Por exemplo: O tipo de dado dinâmico é um tipo de dado polimórfico representado por terminais azul marinho. Terminais booleanos são verdes com letras do tipo TF.

Em geral, terminais azuis devem ser ligados a terminais da mesma azuis, verde com verde, etc. Isto não é uma regra que é válida sempre; o LabVIEW que um usuário ligue um terminal azul (dado dinâmico) a um terminal laranja (valor fracional), por exemplo. Na maioria dos casos, porém, procure combinar as cores.

Controles em uma seta no lado direito e têm uma borda espessa. Indicadores têm uma seta no lado esquerdo e uma borda fina. Regras de lógica se aplicam quando fazemos ligações com o LabVIEW: Cada fio tem uma (mas não necessariamente uma) fonte (ou controle), e cada fio pode ter múltiplos destinos (ou indicadores)

Programação com Dataflow

- Execução com Diagrama de Blocos
 - Dependente do fluxo de dados
 - Diagrama de Blocos NÃO executa da esquerda para direita
- Nó é executado quando dados estão disponíveis em TODOS os terminais de entrada
- Nós fornecem dados para todos os terminais de saída no final da execução



ni.com

NATIONAL
INSTRUMENTS

O LabVIEW segue um modelo de dataflow para executar VIs. Um nó no diagrama de blocos é executado quando todas as suas entradas estão disponíveis. Quando um nó completa a execução, ele fornece dados a seus terminais de saída e passa os dados de saída para o próximo nó no caminho do dataflow. Visual Basic, C++, JAVA, e a maioria das linguagens de programação baseadas em texto seguem um modelo de controle de fluxo para executar programas. Em controle de fluxo, a ordem sequencial dos elementos do programa determina a ordem de execução do mesmo.

Considere o diagrama de blocos acima. Ele soma dois números e em seguida multiplica o resultado da soma por 2. Neste caso, o diagrama de blocos executa da esquerda para a direita, não por que os objetos foram colocados nesta ordem, mas sim por que uma das entradas da função de multiplicação não é válida enquanto a função soma não tenha terminado e passado os dados para a função multiplicação. Lembre que um nó executa somente quando há dados em todos os terminais, e ele fornece dados para todos seus terminais de saída somente no final da execução. Em um segundo pedaço de código, o VI Simulate Signal Express recebe dados de controles e passa estes resultados para o Gráfico.

Você pode considerar que os códigos soma-multiplicação e simulador de sinal co-existem paralelamente em um mesmo diagrama de blocos. Isto significa que ambos começaram a execução ao mesmo tempo e funcionarão independentemente um do outro. Se o outro computador que estiver executando este código tivesse múltiplos processadores, estes dois pedaços de código poderiam ser executados separadamente (cada um em um processador), sem a necessidade de programação adicional.

Técnicas de Depuração

- **Encontrando Erros**



Clique no botão quebrado **Run**.

Uma janela mostrando os erros aparece

- **Execution Highlighting**



Clique no botão de **Execution Highlighting**; o fluxo de dados é animado usando bolhas.

Valores são mostrados nos fios.

- **Probes**



Clique com o botão da direita do mouse em um fio para mostrar um probe e veja os dados na medida que estes passam pelo segmento de fio.



Você pode selecionar, também, a ferramenta “Probe” no paleta de ferramentas e clicar em um fio.

ni.com



Quando seu VI não é executável, uma seta quebrada é mostrada no botão Run.

- **Encontrando Erros:** Para listar os erros, clique na seta quebrada. Para localizar os elementos defeituosos, clique na mensagem de erros.
- **Execution Highlighting:** Anima o digrama e rastreia o fluxo de dados, permitindo que você veja valores intermediários. Clique na lâmpada na barra de ferramentas.
- **Probe:** Usado para ver valores em arrays e clusters. Clique em fios com a ferramenta **Probe** ou clique com o botão direito no fio para colocar um probe.
- **Retenha valores dos fios:** Usado em conjunto com probes para ver os valores da última interação do programa.
- **Breakpoint:** Coloca pausas em diferentes lugares do diagrama. Clique em fios ou objetos com a ferramenta **Breakpoint** para colocar breakpoints.



Exercício 1.2 – Adquirindo um sinal com DAQ (Track A)

Nota: Antes de começar este exercício, copie a pasta Exercícios and Solutions para a área de trabalho do seu computador.

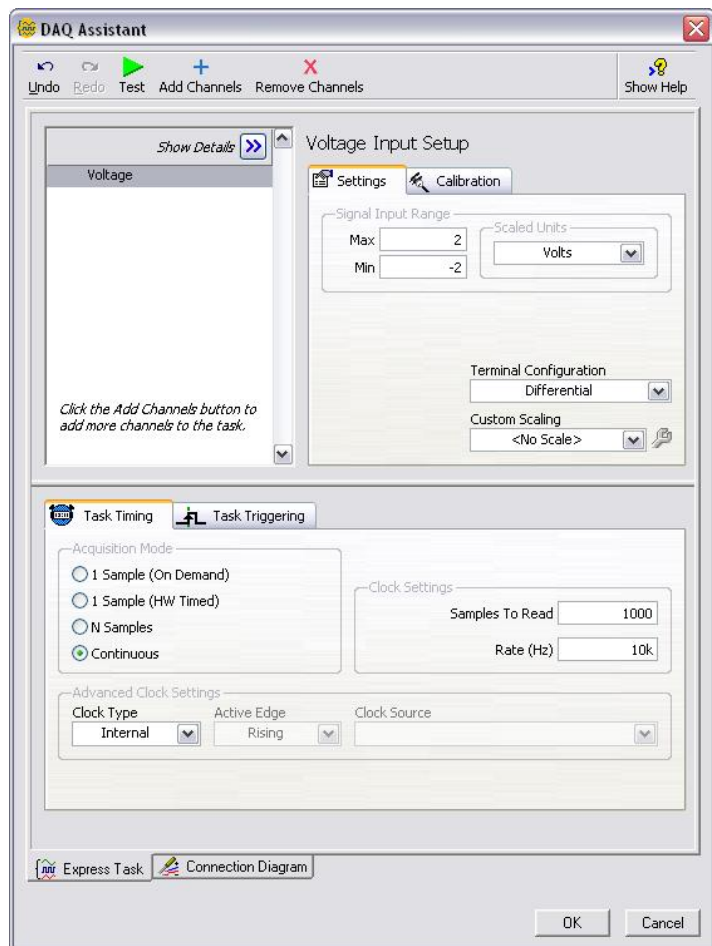
Complete os seguintes passos para criar uma VI que adquire dados continuamente de seu dispositivo DAQ.

1. Abra o LabVIEW.
2. Na janela **Getting Started**, clique em **New** ou no link **VI from Template** para mostrar a caixa de diálogo **New**.
3. Abra um template de aquisição de dados. Da lista Create New, selecione **VI»From Template»DAQ»Data Acquisition with NI-DAQmx.vi** e clique “OK”.
4. Abra o diagrama de blocos clicando nele ou selecionando **Window»Show Diagrama de Blocos**. Leia as instruções que estão escritas lá sobre como completar o programa.
5. Dê um clique duplo no DAQ Assistant para abrir o configuration wizard.
6. Configure uma operação de entrada analógica.
 - a. Escolha **Analog Input»Voltage**.
 - b. Escolha **Dev1 (USB-6009)»ai0** para adquirir dados de no canal 0 de entrada analógica e clique em “Finish.”
 - c. Na próxima janela você pode definir parâmetros de sua operação de entrada analógica. Escolha um intervalo de entrada que funcione bem com seu microfone, no tab setting, entre com **2 Volts** para o máximo e **-2 Volts** para o mínimo. No tab task timing, escolha “**Continuous**” para o modo de aquisição e entre **10000** em rate. Deixe as demais opções com seus valores padrões. Clique em “OK” para sair do wizard.
7. Coloque o VI Filter Express à direita de seus DAQ Assistant no diagrama de blocos. Do palette de funções selecione, **Express»Signal Analysis»Filter** e clique este VI no diagrama de blocos dentro de um loop while. Quando você abrir o palette de funções, aberte o pequeno botão no canto esquerdo superior do palette. Isto irá prender o palette para que ele não desapareça. Este passo será omitido nos demais exercícios mas deve ser repetido. Na janela de configuração dentro de Filtering Type, selecione “Highpass.” Dentro de Cutoff Frequency, use o valor 300 Hz. Clique “OK.”

8. Make the following connections on the Diagrama de Blocos by hovering your mouse over the terminal so that it becomes the wiring tool and cliqueing once on each of the terminals you wish to connect:
 - a. Connect the “Data” output terminal of the DAQ Assistant VI to the “Signal” input of the Filter VI.
 - b. Create a graph indicator for the filtered signal by Clique com o botão direito do mouseing on the “Filtered Signal” output terminal and choose **Create»Graph Indicator**.
9. Return to the Painel Frontal by selecting **Window»Show Paine Frontal** or by pressing <Ctrl+E>.
10. Run your program by cliqueing the run button. Hum or whistle into the microphone to observe the changing voltage on the graph.
11. Clique stop once you are finished.
12. Save the VI as “Exercício 2 – Acquire.vi” in your Exercícios folder and close it.

Nota: A solução deste exercício está no final deste manual

Tip: You can place the DAQ Assistant on your Diagrama de Blocos from the Functions Palette. Clique com o botão direito do mouse the Diagrama de Blocos to open the Functions Palette and go to **Express»Input** to find it.



(Fim do Exercício)



Exercício 1.2 – Adquirindo um sinal com DAQ (Track B)

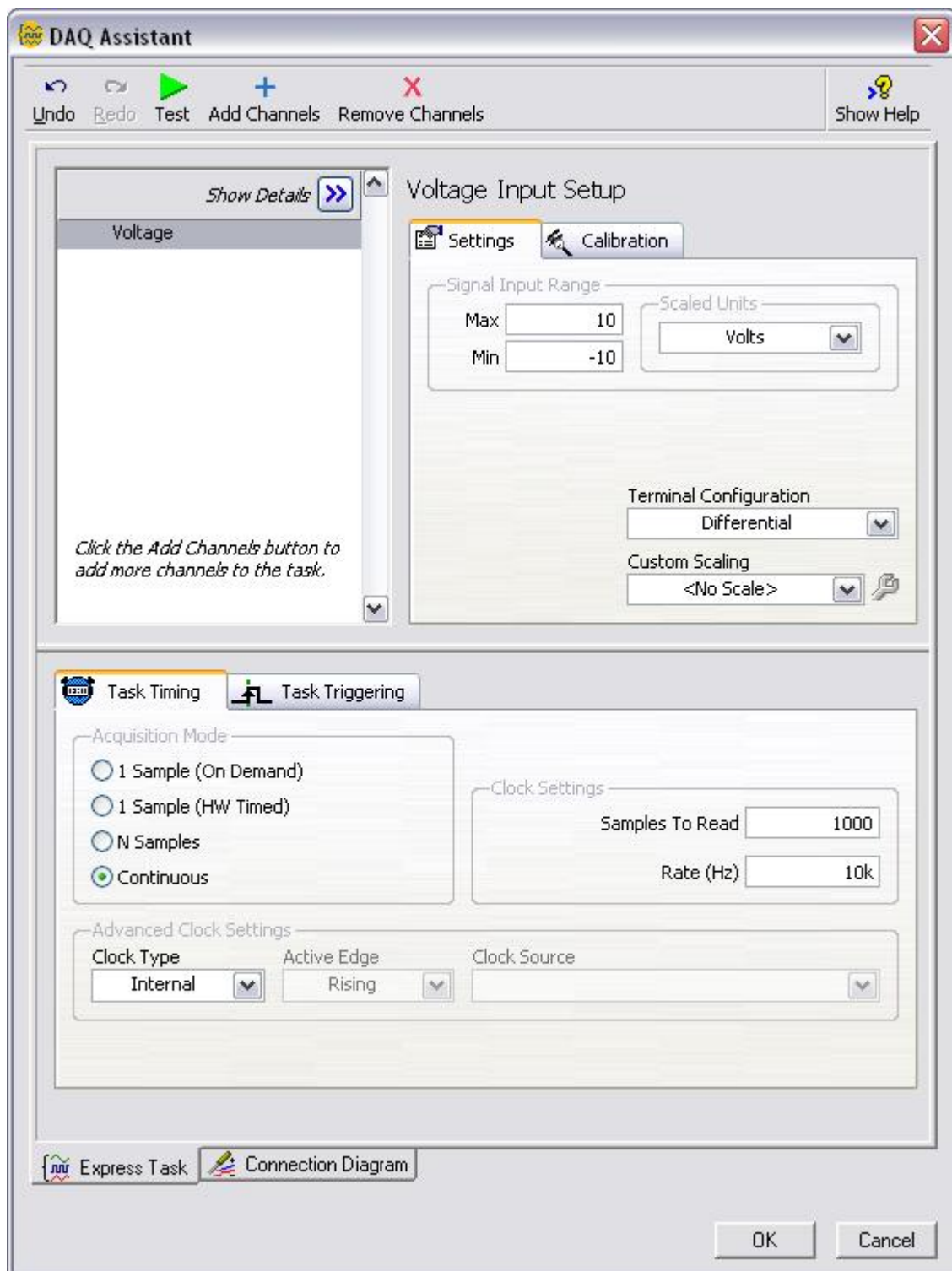
Nota: Antes de começar este exercício, copie a pasta Exercícios and Solutions para a área de trabalho do seu computador.

Complete os seguintes passos para criar uma VI que adquira dados continuamente de seu dispositivo DAQ.

1. Abra o LabVIEW.
2. Na janela **Getting Started**, clique em **New** ou no link **VI from Template** para mostrar a caixa de diálogo **New**.
3. Abra um template de aquisição de dados. Da lista **Create New**, selecione **VI»From Template»DAQ»Data Acquisition with NI-DAQmx.vi** e clique “OK”.
4. Abra o diagrama de blocos clicando nele ou selecionando **Window»Show Diagrama de Blocos**. Leia as instruções que estão escritas lá sobre como completar o programa.
5. Dê um clique duplo no DAQ Assistant para abrir o configuration wizard
6. Configure uma operação de entrada analógica.
 - a. Escolha **Analog Input»Voltage**.
 - b. Escolha **Dev1 (PCI-6220)»ai0** para adquirir dados no canal 0 de entrada analógica e clique em “Finish.”
 - c. Na próxima janela você irá definir os parâmetros de sua operação de entrada analógica. No tab **task timing**, escolha “**Continuous**” para o modo de aquisição, entre com **1000** para o número de samples a serem lidos, e **10000** para o rate. Deixe as demais opções com seus valores padrões. Clique “OK” para sair do wizard.
7. No diagrama de blocos, clique com o botão direito na seta preta para a direita para onde esta escrito “data.” Escolha **Create»Graph Indicator** do menu do botão direito.
8. Retorne ao painel frontal selecionando **Window»Show Painel Frontal** ou apertando <Ctrl+E>.
9. Execute seu programa clicando no botão Run. Observe a onda senoidal simulada no gráfico.
10. Clique em “stop” quando você terminar.
11. Salve o VI como “Exercício 2 – Acquire.vi” na pasta de Exercícios. Feche o VI.

Notas:

- A solução deste exercício está impressa no final deste manual.
- Você pode colocar o DAQ Assistant em seu diagrama de blocos a partir do paleta de funções. Clique com o botão direito do mouse no diagrama de blocos e abra o paleta de funções e entre em **Express»Input** para encontrar o DAQ Assistant. Quando você abrir o paleta de funções, aperte o pequeno botão no canto esquerdo superior do paleta. Isto irá prender o paleta para que ele não desapareça. Este passo será omitido nos demais exercícios mas deve ser repetido.



(Fim do Exercício)



Exercício 1.2 – Adquirindo um sinal com DAQ (Track C)

Nota: Antes de começar este exercício, copie a pasta Exercícios and Solutions para a área de trabalho do seu computador.

Complete os seguintes passos para criar uma VI que adquira dados continuamente de seu dispositivo DAQ.

1. Abra o LabVIEW.
2. Na janela **Getting Started**, clique no link **Blank VI**.
1. Abra o diagrama de blocos apertando <Ctrl+E> ou selecionando **Window»Show Diagrama de Blocos**.
2. Coloque o VI Acquire Sound Express no diagrama de blocos. Clique com o botão direito do mouse para abrir o paleta de funções e selecione **Express»Input»Acquire Sound**. Coloque o VI Expresso no diagrama de blocos.
3. Na janela de configurações em **#Channels**, selecione **1** da lista e clique “OK”.
6. Coloque o VI Filter Express à direita do VI Acquire Signal no diagrama de blocos. No paleta de funções, selecione **Express»Signal Analysis»Filter** e coloque-o no diagrama de blocos. Na janela de configuração dentro de **Filtering Type**, selecione “**Highpass**.” em Cutoff Frequency, use um valor de 300 Hz. Clique “OK.”
7. Faça as seguintes conexões no diagrama de blocos passando seu mouse por cima dos terminais para que ele se transforme na ferramenta de fiação e clique uma vez em cada um dos terminais que você deseja conectar:
 - a. Conecte o terminal de saída “Data” do VI à entrada “Signal” do VI Filter.
 - b. Crie um indicador gráfico para o sinal filtrado clicando com o botão direito do mouse no terminal de saída do “Filtered Signal” e escolha **Create»Graph Indicator**.
8. Volte para o painel frontal apertando <Ctrl+E> ou **Window»Show Painel Frontal**.
9. Execute seu programa clicando no botão Run. Faça um zumbido ou assobie em seu microfone e observe os dados adquiridos de sua placa de som.
10. Salve o VI como “Exercício 1.2 – Acquire.vi” na pasta de exercícios.
11. Feche o VI.

Nota: A solução deste exercício está impressa no final deste manual.

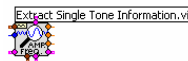
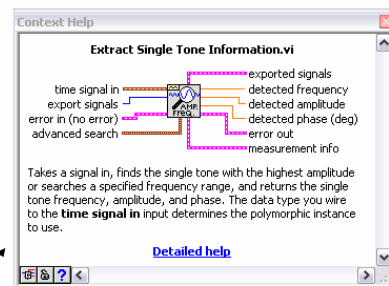
(Fim do Exercício)

Janela Context Help

- **Help»Show Context Help**, aperte as teclas <Ctrl+H>
- Passe o cursor por cima de um objeto para atualizar a janela

Ajuda Adicional

- Clique com o botão direito do mouse em um ícone de VI e escolha **Help**, ou
- Escolha “**Detailed Help**.” na janela context help



ni.com



A Janela **Context Help window** mostra informações básicas sobre os objetos do LabVIEW quando você passa o cursor sobre cada item. Objetos com ajuda de conteúdo incluem VIs, funções, constantes, estruturas, paletes, propriedades, métodos, eventos, e componentes da caixa de diálogo.

Para visualizar a janela Context Help, selecione **Help»Show Context Help**, aperte as teclas <Ctrl+H>, ou aperte o botão **Show Context Help Window** na barra de ferramentas

Conexões mostradas no Context Help:

Necessário – **negrito**

Recomendado – normal

Opcional – apagado

Ajuda Adicional

- **VI, Funções, & Ajuda “How-To”** também estão disponíveis.
 - **Help» VI, Function, Ajuda& How-To Help**
 - Clique com o botão direito do mouse no ícone do VI e escolha **Help**, ou
 - Escolha “**Detailed Help**.” na janela context help.
- **LabVIEW Help – Ajuda referenciada**
 - **Help»Search the LabVIEW Help...**

Dicas para Trabalhar com LabVIEW

- Teclas de Atalho

- <Ctrl+H> – Ativar/Desativar a Janela “Context Help”
- <Ctrl+B> – Retira fios quebrados do Diagrama de Blocos
- <Ctrl+E> – Alterna entre o Painel Frontal e o Diagrama de Blocos
- <Ctrl+Z> – Desfazer (Também no Menu “Edit”)

- **Tools»Options...** – Ajusta as preferências no LabVIEW

- **VI Properties**—Configura a aparência, documentação, etc. da VI.

ni.com



O LabVIEW possui diversos atalhos para facilitar seu trabalho. Os atalhos mais comuns estão listados acima.

Enquanto a ferramenta de seleção automática é ótima para escolher a ferramenta que você gostaria de usar no LabVIEW, há casos quando você quer usar o controle manual. Uma vez que a ferramenta de seleção manual for desligada, você pode usar a tecla Tab para alternar entre as quatro ferramentas mais comuns (Operar Valores, Posição/Tamanho/Selecione, Editar Texto, Ajustar Cores no painel frontal e Operar Valores, Posição/Tamanho/Selecione, Ligar fios no diagrama de blocos). Uma vez que você escolheu a ferramenta, você pode apertar <Shift+Tab> para voltar para ligar a ferramenta de seleção automática novamente.

No diálogo **Tools»Options...**, há diversas opções configuráveis para customizar seu painel frontal, diagrama de blocos, impressão e muito mais.

Parecido com as opções do LabVIEW, você pode configurar as propriedades específicas do VI em **File»VI Properties...** Lá você pode documentar o VI, mudar a aparência da janela, e customizar o VI de outras formas.

Seção II – Elementos de Programas Típicos

A. Loops

- Loop “While”
- Loop “For”

B. Funções e SubVIs

- Tipos de Funções
- Criando e Customizando Funções (SubVI)
- Paleta de Funções & Buscas

C. Tomando Decisões e E/S de Arquivos



- Estrutura “Case”
- Seleção (declaração simples de If)
- E/S de Arquivos

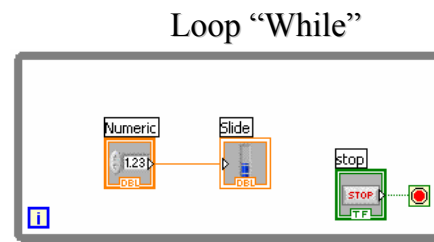
ni.com





Loops

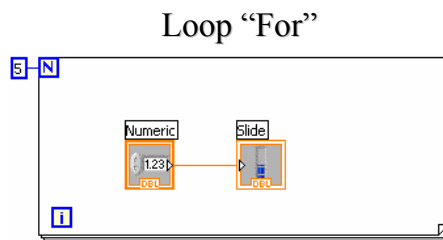
•Loops “While”

- **Terminal**  conta número de interações
- Roda pelo menos uma vez
- Roda até a condição de parada  for atingida



•Loops “For”

- **Terminal**  conta número de interações
- Roda de acordo com a entrada  do terminal de contagem



Tanto o Loop While quanto o For estão localizados no palette **Functions»Structures**. O loop For difere do loop While no fato de que o loop For executa um número fixo de vezes. O loop While para de executar o subdiagrama somente quando o valor no terminal condicional existe.

Loops While

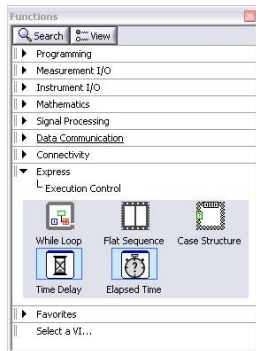
Similar a um Do Loop ou um loop Repita-Até em linguagem de programação baseada em texto, um loop While, mostrado no canto superior direito, executa um subdiagrama até que uma condição seja alcançada. O loop While executa o subdiagrama até o terminal condicional, um terminal de entrada, receber um valor Booleano específico. O comportamento e aparência padrão terminal condicional é **Stop If True**. Quando o terminal condicional é **Stop If True**, o loop while executa seu subdiagrama até o condicional recebe um valor TRUE. O terminal de interação (um terminal de saída), mostrado à esquerda, contém o número de interações completas. A contagem de interações sempre começa no zero. Durante a primeira interação, o terminal de interação retorna 0.

Loops For

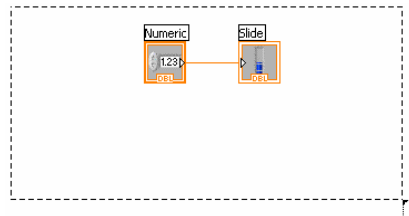
Um loop For, mostrado acima, executa o subdiagrama um número fixo de vezes. O valor no terminal de contagem (um terminal de entrada) representado por N, indica quantas vezes o subdiagrama será repetido. O terminal de interação (um terminal de saída), mostrado à esquerda, contém o número de interações completas. A contagem de interações sempre começa no zero. Durante a primeira interação, o terminal de interação retorna 0.

Desenhando um Loop

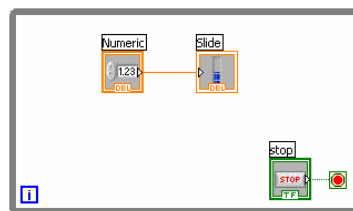
1. Selecione a estrutura



2. Selecione o código a ser repetido



3. Arraste nódulos adicionais e ligue

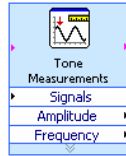


Coloque loops em seu diagram selecionado eles do paleta Structures localizado no paleta de funções:

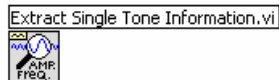
- Quando selecionado, o cursor do mouse se torna um ponteiro especial que você usa para rodear a seção de código que você quer repetir.
- Clique o botão do mouse para definir canto esquerdo superior, clique o mouse novamente no canto inferior direito, e a borda do loop While é criada em volta do código selecionado.
- Arraste ou largue nós adicionais dentro do loop While se for necessário.

3 Tipos de Funções (do Paleta de Funções)

Vis Expressa: Vis interativas com página de dialogo configurável(borda azul)



Vis Padrão: Vis modularizadas customizadas pela fiação (customizavel)



Funções: elemento de operação fundamental do LabVIEW; sem painel frontal ou diagrama de blocos (amarelo)



ni.com

NATIONAL
INSTRUMENTS

O LabVIEW 7.0 introduziu um novo tipo de subVI chamado VIs Expressas. Estes Vis interativos têm uma caixa de diálogo configurável que permite que usuários customizem a funcionalidade da VI Expressa. O LabVIEW gera em seguida uma subVI baseada nestas configurações.

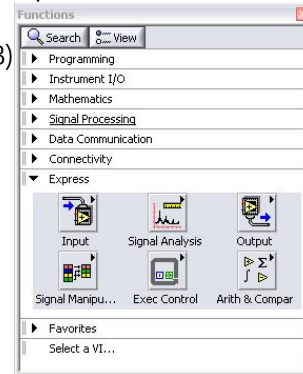
SubVIs são VIs (consistindo de um painel frontal e um diagrama de blocos) que são usados dentro de outra VI.

Funções são a base de todos os VIs. Funções não têm painel frontal ou diagrama de blocos.

Que tipo de Funções estão Disponíveis?

- **Entrada e Saída**
 - Simulação de Dados e Sinais
 - Adquirir e Gere Sinais Reais com DAQ
 - Assistência de E/S de Instrumentos (Serial & GPIB)
 - ActiveX para comunicação com outros programas
- **Análise**
 - Processamento de Sinais
 - Estatísticas
 - Formulas e Matemática Avançada
 - Resolvedor de Tempo Contínuo
- **Armazenamento**
 - E/S de Arquivos

Express Functions Palette



ni.com



O LabVIEW inclui algumas centenas de funções pré-estabelecidas para ajudar você a adquirir, analisar, e apresentar seus dados. Você geralmente usa estas funções como delineado no slide a cima.

Toolkits do LabVIEW

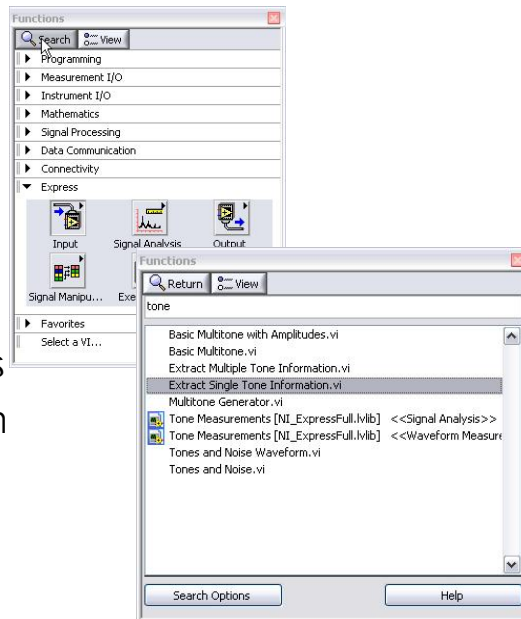
Toolkits adicionais estão disponíveis para acrescentar funções de domínio específico ao LabVIEW. Estas funções incluem:

Application Deployment and Targeting Modules <ul style="list-style-type: none"> * LabVIEW PDA Module * LabVIEW Real-Time Module * LabVIEW FPGA Module * LabVIEW Vision Development Module 	Signal Processing and Analysis <ul style="list-style-type: none"> * Sound and Vibration Toolkit * Advanced Signal Processing Toolkit * Modulation Toolkit * Spectral Measurements Toolkit * Order Analysis Toolkit * Digital Filter Design Toolkit 	Control Design and Simulation <ul style="list-style-type: none"> * Control Design and Simulation Bundle * LabVIEW Real-Time Module * System Identification Toolkit * Control Design Toolkit * LabVIEW Simulation Module * State Diagram Toolkit
Embedded System Deployment <ul style="list-style-type: none"> * DSP Test Integration Toolkit * Embedded Test Integration Toolkit * Digital Filter Design Toolkit * LabVIEW FPGA Module 	Software Engineering and Optimization Tools <ul style="list-style-type: none"> * Execution Trace Toolkit for LabVIEW Real-Time * Express VI Development Toolkit * State Diagram Toolkit * VI Analyzer Toolkit 	Image Processing and Acquisition <ul style="list-style-type: none"> * LabVIEW Vision Development Module * NI Vision Builder for Automated Inspection * NI-IMAQ for IEEE 1394

<http://www.ni.com/toolkits/>

Encontrando Controles, VIs, e Funções

- Paletes estão repletos de centenas de VIs
- Pressione o botão “search” para indexar todas as Vis para uma busca
- Clique e arraste um item da janela de busca para o diagrama de blocos
- De um clique-duplo em cima do item para abrir o palette que o contém



ni.com



Use os botões no topo das janelas dos paletes para navegar, pesquisar e editar os paletes.

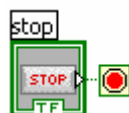
Você pode procurar controles, VIs, e funções que ou contém certas palavras ou começam com certas palavras. Dando um clique duplo em um resultado de busca abre o palette que contém estes resultados de busca. Você pode clicar e arrastar o nome do controle, VI, ou função diretamente para o painel frontal ou diagrama de blocos.



Exercício 2.1 – Analise (Track A, B, & C)

Crie um VI que produz uma onda senoidal com uma frequência específica e exiba os dados em uma tabela de forma de onda até finalizado pelo usuário.

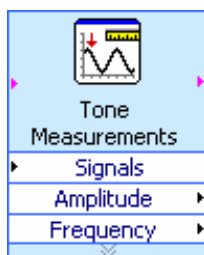
1. Abra um VI em branco a partir da tela Getting Started.
2. Coloque um gráfico no painel frontal. Clique com o botão direito do mouse para abrir o paleta de controles e selecione **Controls»Modern»Graph»Waveform Chart**.
3. Coloque um controle do tipo “dial” no painel frontal. Do paleta de controles selecione **Controls»Modern »Numérico»Dial**. Veja que quando colocamos o controle no painel frontal, o nome está destacado. Enquanto ele estiver destacado, digite “Frequency In” para dar um nome a este controle.
4. Vá ao diagrama de blocos(<Ctrl+E>) e coloque um loop while. Clique com o botão direito do mouse para abrir o paleta de funções e selecione **Express»Execution Control»While Loop**. Clique e arraste no diagrama de blocos para criar um loop while de tamanho correto. Selecione a tabela de formato de onda e o controle “dial” se eles não foram ainda. Veja que o botão “stop” já está conectado ao terminal condicional do loop while.



5. Coloque o VI Expresso Simulate Signal Express no diagrama de blocos. Do paleta de funções, selecione **Express»Signal Analysis»Simulate Signal** e coloque-o dentro do loop while do diagrama de blocos. Na janela de configuração, no item Timing, escolha “Simulate acquisition timing.” Clique “OK.”



6. Coloque um Tone Measurements Express VI no Diagrama de Blocos (**Express»Signal Analysis»Tone Measurements**). Na janela de configuração, escolha medidas de Amplitude e Frequência na seção Single Tone Measurements. Clique “OK.”



7. Faça as seguintes conexões ao diagrama de blocos passando seu cursor por cima do terminal para que este se transforme na ferramenta de fiação e clique uma vez em cada um dos terminais que você deseja ligar:
- Ligue o terminal de saída “Sine” do VI Simulate Signal ao terminal de entrada “Signal” do VI Tone Measurement. “Signals”
input of the Tone Measurements VI.
 - Ligue a saída “Sine” ao Waveform Chart.
 - Crie indicadores para medidas de amplitude e frequência clicando com o botão direito do mouse em cima de cada terminal Tone Measurements Express VI e selecionando **Create»Numérico Indicator**.
 - Ligue o controle “Frequency In” ao terminal “Frequency” do VI Simulate Signal.
8. Volte ao painel frontal e execute o VI. Use o controle “Frequency In” e observe a frequência do sinal. Clique no botão “stop” quando você tiver terminado.
9. Salve o VI como “Exercício 2.1 – Simulated.vi”.
10. Feche o VI.

Notas

- Quando você abrir o paleta de funções, aberte o pequeno botão no canto esquerdo superior do paleta. Isto irá prender o paleta para que ele não desapareça. Este passo será omitido nos demais exercícios mas deve ser repetido.
- A solução deste exercício está no final deste manual.

(Fim do Exercício)



Exercício 2.2 - Analise

Crie uma VI que meça a frequência e amplitude de sinal do seu dispositivo DAQ (simulado) e mostre o sinal adquirido em um gráfico de formato de onda. As instruções são as mesmas que no Exercício 2.1, mas o DAQ Assistant é usado no lugar do VI Simulate Signal. Tente fazer sem usar as instruções abaixo.

Create a VI that measures the frequency and amplitude of the signal from your (simulated) DAQ device and displays the acquired signal on a waveform chart. The instructions are the same as in Exercício 2.1, but a DAQ Assistant is used in place of the Simulate Signal VI. Try to do this without following the instructions!

1. Abra um VI em branco.
2. Coloque um gráfico no painel frontal. Clique com o botão direito do mouse para abrir o paleta de controles e selecione **Controls»Modern»Graph»Waveform Chart**.
3. Vá até o diagrama de blocos e coloque um loop while (**Express»Execution Control»While Loop**).
4. Coloque o DAQ Assistant no diagrama de blocos (**Express»Input»DAQ Assistant**). Escolha uma entrada analógica no canal 0 de seu dispositivo (simulado) e clique "Finish". No tab task timing, escolha "continuous" para o modo de aquisição. Se você estiver usando o USB-6009, mude o intervalo de entrada de -2 até 2 e o número de samples a serem lidos para 100.
5. Coloque o VI Filter Express à direita do DAQ Assistant no diagrama de blocos. No paleta de funções, selecione **Express»Signal Analysis»Filter** e coloque-o no diagrama de blocos dentro do loop while. Na janela de configuração, em "Filtering Type," selecione "Highpass." Em "Cutoff Frequency, use o valor 300 Hz e clique "OK."
6. Conecte o terminal de saída "Data" do DAQ Assistant à entrada "Signal" do VI Filter.
7. Conecte o terminal "Filtered Signal" no VI Filter ao Waveform Chart.
8. Coloque o VI Expresso Tone Measurements Express VI no diagrama de blocos (**Express»Signal Analysis»Tone**). Na janela de configuração, escolha medidas Amplitude e Frequência na seção de Single Tone Measurements.
9. Crie indicadores para medidas amplitude e frequência clicando com o botão direito do mouse em cada terminal do VI Expresso Tone Measurement Express e selecione **Create»Numérico Indicator**.
10. Conecte a saída do Filtro à entrada "Signals" do Tone Measurements Express VI.
11. Retorne ao painel frontal e execute o VI. Observe seu sinal adquirido e sua frequência e amplitude. Faça um zumbido ou assobie em seu microfone se você tiver um USB-6009 e observe a amplitude e frequência que você está produzindo.
12. Salve o VI como "Exercício 2.2 - Data.vi".
13. Feche o VI.

Nota: A solução deste exercício está no final deste manual.

(Fim do Exercício)



Exercício 2.2 – Analysis (Track C)

Create a VI that measures the frequency and amplitude of the signal from your sound card and displays the acquired signal on a waveform chart. The instructions are the same as in Exercício 2.1, but the Sound Signal VI is used in place of the Simulate Signal VI. Try to do this without following the instructions!

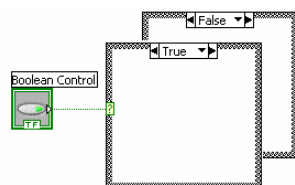
1. Open a blank VI.
2. Go to the Diagrama de Blocos and place a While Loop down (**Express»Execution Control»While Loop**).
3. Place the Acquire Sound Express VI on the Diagrama de Blocos (**Express»Input»Acquire Sound**).
4. Place a Filter Express VI on the Diagrama de Blocos. In the configuration window choose a highpass filter and a cutoff frequency of 300 Hz.
5. Place a Tone Measurements Express VI on the Diagrama de Blocos (**Express»Signal Analysis»Tone**). In the configuration window, choose Amplitude and Frequency measurements in the Single Tone Measurements section.
6. Create indicators for the amplitude and frequency measurements by Clique com o botão direito do mouseing on each of the terminals of the Tone Measurements Express VI and selecting **Create»Numérico Indicator**.
7. Connect the “Data” terminal of the Acquire Sound Express VI to the “Signal” input of the Filter VI.
8. Connect the “Filtered Signal” terminal of the Filter VI to the “Signals” input of the Tone Measurements VI.
9. Create a graph indicator for the Filtered Signal by Clique com o botão direito do mouseing on the “Filtered Signal” terminal and selecting **Create»Graph Indicator**.
10. Return to the Painel Frontal and run the VI. Observe the signal from your sound card and its amplitude and frequency. Hum or whistle into the microphone and observe the amplitude and frequency you are producing.
11. Save the VI as “Exercício 2.2-Data.vi”. Close the VI.

Nota: A solução deste exercício está no final deste manual.

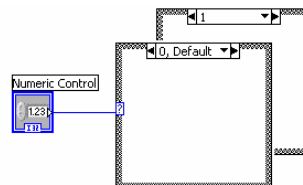
(Fim do Exercício)

How Do I Make Decisions in LabVIEW?

1. Case Structures

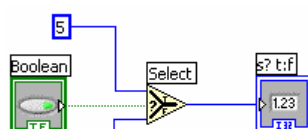


(a)



(b)

2. Select



(c)

ni.com

NATIONAL
INSTRUMENTS

Estrutura Case

A Estrutura Case tem um ou mais subdiagramas, ou casos, dos quais exatamente um será executado com a estrutura. O valor ligado ao terminal seletor Case Structure has one or more subdiagramas, or cases, exactly one of which executes when the structure executes. The value wired to the selector determina que caso executar e pode ser do tipo booleano, string, inteiro, ou enumerado. Clique com o botão direito do mouse na borda da estrutura para adicionar ou deletar casos. Use a ferramenta de texto para entrar com valores no rótulo do seletor de casos e configure os valores administrados por cada caso. Isto é encontrado em **Functions»Programming»Structures»Case Structure**.

Select

Retorna os valores ligados à entrada **t** ou saída **f**, dependendo do valor de **s**. Se **s** for TRUE, esta função retorna o valor ligado a **t**. Caso contrário, a função retorna o valor ligado a **f**. O painel de conexões mostra os tipos de dados default para esta função polimórfica. Ela se encontra em **Functions»Programming»Comparison»Select**.

- **Exemplo a:** entrada Booleana: Simple if-then case. If the Boolean input is TRUE, the true case will execute; otherwise the FALSE case will execute.
- **Exemplo b:** entrada Numérica. O valor de entrada determina que caixa deve ser executada. Se estiver for a do alcance dos casos, o LabVIEW executa o caso default.
- **Exemplo c:** Quando o Booleano passa um valor TRUE para o Select VI, o valor 5 é passado para o indicador. Quando o Booleano passa um valor FALSE para o Select VI, 0 é passado para o indicador.

E/S de Arquivo

- E/S de Arquivo – Allows recording or reading data in a file.
- LabVIEW creates or uses the following file formats:
 - Binary: underlying file format of all other file formats
 - ASCII: regular text files
 - LVM: LabVIEW measurement data file
 - TDM: created for National Instruments products

ni.com



Operações de E/S de Arquivo passam dados da memória para e de arquivos. No LabVIEW você pode usar as funções de E/S de Arquivo para:

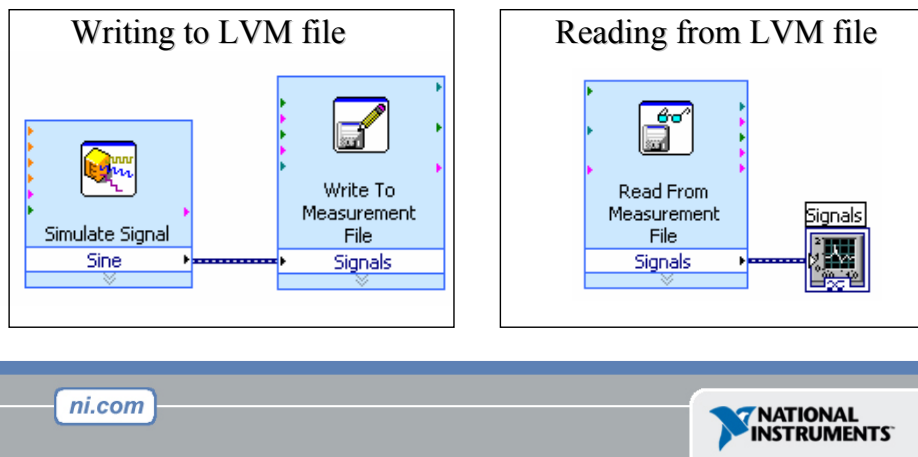
- Abrir e fechar arquivos de dados
- Ler e escrever dados em arquivos
- Ler e escrever dados para planilhas
- Mudar e renomear arquivos e diretórios
- Alterar características de arquivos
- Criar, modificar, e ler configurações do arquivo

Os formatos diferentes de arquivos que o LabVIEW pode usar ou criar são os seguintes:

- Binário – Arquivos binários são o formato fundamental dos demais formatos de arquivo.
- ASCII – Um arquivo ASCII é um tipo específico de arquivo binário que é um padrão usado na maioria dos programas. ASCII são também conhecidos como arquivos de texto.
- LVM – O LabVIEW measurement data file (.lvm) é um formato de arquivo delimitado por abas que você pode abrir com uma aplicação de planilha ou editor de texto. Este formato de arquivo é um tipo específico de ASCII criado pelo LabVIEW. O formato .lvm contém informação sobre os dados como data e hora que os dados foram gerados.
- TDM – Este formato de arquivo é um tipo específico de binário criado para produtos National Instruments. Ele consiste em dois tipos separados de arquivos: uma seção XML contendo atributos dos dados e uma seção binária para formato de ondas.

High Level E/S de Arquivo Functions

- Easy to use
- High Level of abstraction



E/S de alto nível de Arquivo : Estas funções trazem um nível maior de abstração para o usuário abrindo e fechando arquivos automaticamente antes e depois de ler ou escrever dados. Algumas destas funções incluem são:

- o **Escrever em uma planilha** – Converte um array de 1D ou 2D de números de precisão simples em um string e escreve este string em um novo arquivo ASCII ou acrescenta o string em um arquivo existente.
- o **Ler de uma planilha** – Le um número especificado de linhas ou colunas de um arquivo de texto numérico começando num caracter especificado e converte os dados em um array de 2D de números de precisão simples. O VI abre o arquivo antes de ler e fecha depois.
- o **Write to Measurement File** – VI Expresso que escreve dados em um arquivo de medidas baseado em texto (.lvm) ou formato de arquivo de medidas binário (.tdm).
- o **Read from Measurement File** – VI Expresso que lê dados de um arquivo de medidas baseado em texto (.lvm) ou formato de arquivo de medidas binário (.tdm) format. Você pode especificar o nome, formato e tamanho do segmento do arquivo.

Estas funções são fáceis de usar e são excelentes para aplicações simples. No caso em que você terá um fluxo constante para o arquivo devido a escrita ou leitura contínua do mesmo, pode haver um teto em algumas destas funções.

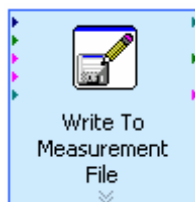
No próximo exemplo veremos como escrever e ler de arquivos*.lvm.



Exercício 2.3 – Decisões e Salvando dados (Track A, B, & C)

Crie um VI que permite que você salve seus dados em um arquivo se a frequência do seus dados estiver abaixo de um limite controlado pelo usuário.

1. Abra Exercício 3.2 – Data.vi.
2. Vá para **File»Save As...** e salve como “Exercício 3.3 – Decision Making and Saving Data”. Na caixa “Save As” selecione **substitute copy for original** e clique “Continue...”.
3. Acrescente uma estrutura case ao Diagrama de Blocos dentro do loop while (**Functions»Programming»Structures»Case Structure**).
4. Dentro do caso “true”, acrescente um Write to Measurement File Express VI (**Functions»Programming»E/S de Arquivo»Write to Measurement File**).

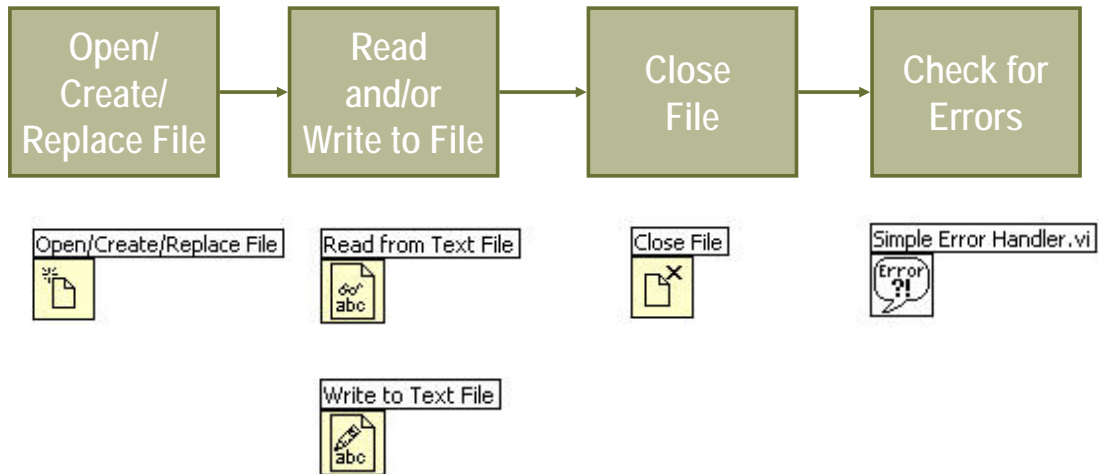


- a. Na janela de configurações que abre, escolha **Save to series of files (multiple files)**.” Note qual é o local default para salvar seu arquivo e mude se quiser.
 - b. Clique em “Settings...” e escolha “Use next available file name” em **Existing Files**.
 - c. Em **File Termination** escolha para começar um novo arquivo a cada 10 segmentos. Clique “OK” duas vezes.
5. Acrescente código para que a frequência computada pelo **Tone Measurements Express VI** que está abaixo do limite será salva. Dica: Vá para **Functions»Programming»Comparison»Less?**
 6. Lembre-se de conectar os dados do **DAQ Assistant** ou o **Acquire Sound Express VI** à entrada “Signals” do **Measurement File VI**. Se você precisar de ajuda refira à solução deste exercício.
 7. Vá ao Painel Frontal e execute seu VI. Varie seu limite de frequência e pare o VI.
 8. Vá até **My Documents»LabVIEW Data** e abra um dos arquivos que foram salvos. Examine a estrutura do arquivo e verifique que há 10 segmentos no arquivo.
 9. Salve e Feche seu VI .

Nota: A solução deste exercício está no final deste manual

(Fim do Exercício)

E/S de Arquivo Programming Model – Under the hood



ni.com

NATIONAL INSTRUMENTS

Modelo de Programação para os VIs de Arquivos Intermediários

Este mesmo modelo de programação se aplica a aquisição de dados, controle de instrumentos, E/S de arquivos, e outros esquemas de comunicação. Na maioria dos instantes você abrirá o arquivo ou canal de comunicação, ler e escrever várias vezes, e aí a comunicação será fechada ou encerrada. É boa prática de comunicação checar por erros no final. Lembre-se deste modelo de programação quando você passar para programação mais avançada ou ver DAQ, comunicação, ou E/S de arquivos de VIs Expressos.

VIs e Funções de E/S de Arquivo

Use as funções e VIs de E/S de Arquivo para abrir, fechar, ler e escrever em arquivos, criar diretórios e arquivos em que você especifica o caminho do controle, retirar informações sobre diretórios, e escrever strings, números, arrays e clusters em arquivos.

Use VIs de alto nível de E/S de Arquivo localizadas na fileira de cima do paleta para realizar operações comuns de E/S, como escrever ou ler vários tipos de dados. Tipos aceitáveis podem incluir caracteres, ou linhas em arquivos textos, arrays de 1D ou 2D (valores numéricos de precisão simples) em arquivos binários, ou arquivos binários de 16 bits.

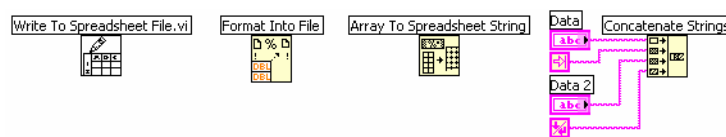
Use VIs e funções de baixo nível de E/S de Arquivo localizadas na fileira do meio do paleta, funções avançada para controlar cada operação de arquivo individualmente.

Use as principais funções de baixo nível para criar ou abrir, escrever, ler, e fechar arquivos. Você pode, também, criar diretórios mudar, copiar, ou deletar arquivos; listar conteúdo de diretórios; mudar características de arquivos; ou manipular caminhos.

Refira ao NI Developer Zone para mais informações sobre escolhendo um formato de arquivo.

planilha Formatting

- planilha files are ASCII files with a certain formatting
 - Usually tabs between columns and end of line constants between rows
 - LabVIEW includes VIs that perform this formatting or a string can be concatenated



ni.com

NATIONAL
INSTRUMENTS

planilhas são arquivos ASCII com um certo tipo de formatação. Dois métodos são valores separados por vírgulas (CSV) e delimitado por abas. Arquivos delimitados por abas, que são o mais popular, têm abas constantes entre colunas de dados e no final de cada linha de constantes e entre fileiras. O LabVIEW inclui VIs que realizam esta formatação :

Write to Spreadsheet File pega um array de 1D ou 2D de dados numéricos e formata estes dados, e escreve a informação em um arquivo.

Format Into File pega diversos tipos de dados (string, numérico, Booleano) e escreve esta informação em um arquivo, usando ou um caminho ou referencia de arquivo. Esta função pode ser redimensionada para incluir terminais de dados se necessário.

Array to Spreadsheet String é uma função de string que formata um array de dados em uma string para ser escrita em um arquivo texto.

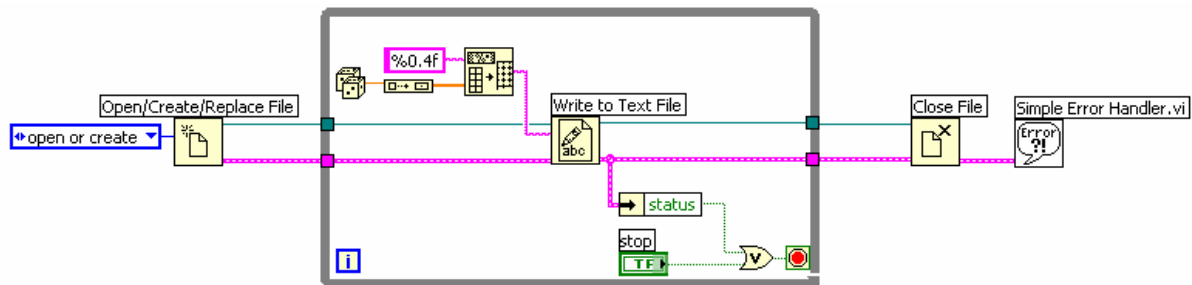
A função **Concatenate String** é usada para criar strings maiores a partir de outras menores e é mais flexível quando convertemos dados para uma string que pode ser escrita em um arquivo texto.



Exercício 2.4 – Escrever em uma planilha

1. Abra um VI novo da tela Getting Started.
2. Coloque uma função Open/Create/Replace File no Diagrama de Blocos. Clique com o botão direito do mouse no Diagrama de Blocos para abrir o palete de funções e selecione File I/O » Open/Create/Replace File.
3. Clique com o botão direito do mouse no terminal **operation** da função Open/Create/Replace File e selecione **Create » Constant** do menu de atalhos e selecione **open or create** do menu que desce.
4. Coloque um loop While do palete de estruturas no Diagrama de Blocos à direita da função Open/Create/Replace File. Clique com o botão direito do mouse no Diagrama de Blocos e selecione Structures » While Loop.
5. Coloque uma função Write Text File dentro do Loop While. Clique com o botão direito do mouse no Diagrama de Blocos e selecione File I/O » Write To Text File.
6. Ligue o terminal **refnum out** da função Open/Create/Replace File ao terminal **file (use dialog)** da função Write Text File.
7. Ligue o terminal **error out** da função Open/Create/Replace File ao **error in** da função Write Text File.
8. Coloque uma função Array to Spreadsheet String dentro do loop while e à esquerda da função Open/Create/Replace File. Clique com o botão direito do mouse no Diagrama de Blocos e selecione String » Array to Spreadsheet String.
9. Clique com o botão direito do mouse no terminal **format string** da função Array to Spreadsheet e selecione **Create » Constant** do menu de atalho e entre com **“%0.4f”** na constante de string para formatar os dados de entrada.
10. Coloque uma função Build Array no Diagrama de Blocos. Clique com o botão direito do mouse no Diagrama de Blocos e selecione Array » Build Array.
11. Coloque um Random Number dentro do loop while. Clique com o botão direito do mouse no Diagrama de Blocos e selecione Numeric » Random Number (0-1).
12. Ligue o **error out** do terminal da função Write Text File para um túnel de saída do loop while.
13. Coloque uma função Unbundle By Name dentro do loop while. Clique com o botão direito do mouse no Diagrama de Blocos para abrir o palete de funções e selecione **Cluster & Variant » Unbundle By Name**.
14. Ligue o error out da função Write Text File à função Unbundle By Name.
15. Coloque uma função Or no loop while. Clique com o botão direito do mouse no Diagrama de Blocos para abrir o palete de funções e selecionar **Boolean » Or**.
16. Passe para o Painel Frontal e coloque um botão stop. Clique com o botão direito do mouse no Painel Frontal para abrir o palete de controles e selecione **Boolean » Stop Button**.
17. No Diagrama de Blocos, ligue o elemento **status** do cluster de erros à entrada x da função Or e ligue o botão stop na entrada y.
18. Ligue a saída do Or no terminal condicional do loop while.

19. Coloque uma função Close File à direita do loop While. Clique com o botão direito do mouse no Diagrama de Blocos para abrir o palete de funções e selecione **File I/O » Close File**.
20. Ligue o tunel de saída refnum ao terminal de entrada refnum da função Close File.
21. Ligue o tunel de saída de erros ao terminal error in da função Close File.



22. Volte ao Painel Frontal e execute o VI. Você será advertido para “Choose or enter path of file to open”, entre: “planilha.xls”.
23. Clique no botão stop para terminar a execução do VI.
24. Abra o arquivo: “planilha.xls”.
25. Salve e feche o VI.

(Fim do Exercício)

Seção III – Apresentando seus Resultados

A. Exibindo dados no Painel Frontal

- Controles e Indicadores
- Gráficos e Tabelas
- Timing de Loop

B. Processamento de Sinais

- MathScript
- Arrays
- Clusters
- Formatos de Ondas

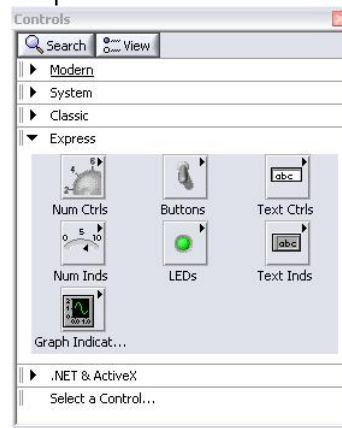
ni.com



Que tipos de controles e indicadores estão disponíveis?

- **Dados Numéricos**
 - Entrada e visualização de números
 - Controles analógicos deslizantes, marcadores, e bitolas
- **Dados Booleanos**
 - Botões e LEDs
- **Dados de Array & Matriz**
 - Display Numérico
 - Tabelas
 - Gráficos
 - Gráficos XY
 - Gráficos de Intensidade
 - Gráficos 3D: ponto, superfície, e modelo
- **Decoração**
 - Controle de Abas
 - Setas
- **Outros**
 - Strings e caixas de texto e text boxes
 - Display de Fotos/Imagens
 - Controles ActiveX

Express Controls Palette



ni.com

NATIONAL
INSTRUMENTS

Controles e Indicadores são itens do Painel Frontal que permitem usuários a interagirem com o programa para trazer informações de entrada e mostrar resultados. Você pode acessar Controles e Indicadores clicando com o botão direito do mouse no Painel Frontal.

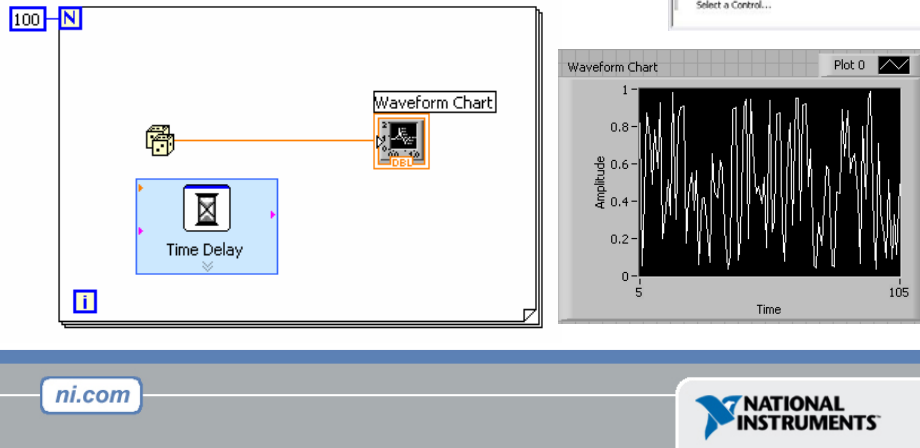
Além disto, você recebe controles e indicadores adicionais quando você instala módulos e toolkits.

Por exemplo, quando você instala as ferramentas de Control Design, você recebe gráficos especializados como o de Bode e Nyquist que não estão disponíveis por default.

Tabelas – Acrescente 1 ponto de dado por vez com histórico

- Tabela de formato de onda– indicador numérico especial que mostra um histórico de valores
- Tabela atualiza com cada ponto individual recebido

Functions»Express»Graph Indicators»Chart



A tabela de formato de onda é um indicador numérico que mostra um ou mais gráficos. A tabela de formato de onda é localizado no paleta **Controls»Modern»Graph**. Tabelas de formato de onda podem ser visualizados em um ou mais gráficos. O Painel Frontal seguinte mostra um exemplo de uma tabela com mais de um formato de onda.

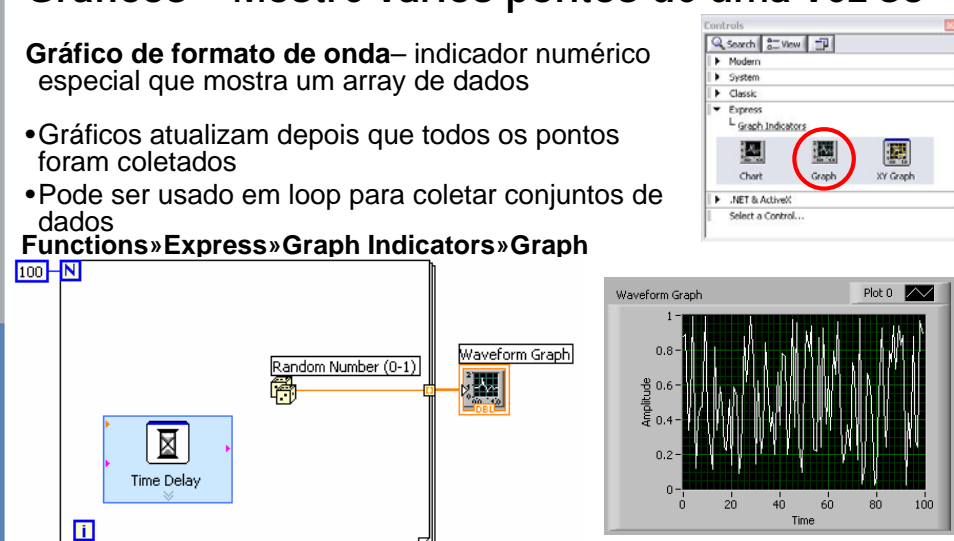
Você pode mudar o máximo e mínimo dos eixos x e y clicando duas vezes no valor com a ferramenta de rotulação e digitando novos valores. Similarmente, você pode mudar a legenda dos eixos. Você pode também clicar na legenda do gráfico e mudar, estilo, formato, e cor do traço que é mostrado na tabela.

Gráficos – Mostre vários pontos de uma vez só

Gráfico de formato de onda– indicador numérico especial que mostra um array de dados

- Gráficos atualizam depois que todos os pontos foram coletados
- Pode ser usado em loop para coletar conjuntos de dados

Functions»Express»Graph Indicators»Graph

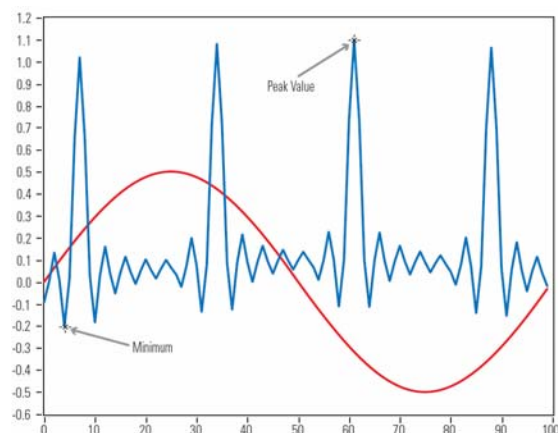
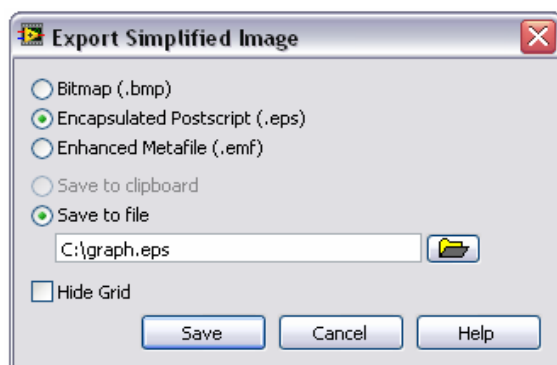


The image shows a LabVIEW interface. On the left, a block diagram features a 'Random Number (0-1)' block connected to a 'Waveform Graph' block. A 'Time Delay' block is connected to the 'Random Number' block, creating a feedback loop. On the right, the 'Waveform Graph' window is open, showing a plot of amplitude (y-axis, 0 to 1) versus time (x-axis, 0 to 100). The plot displays a noisy signal. The 'Controls' palette on the far right shows the 'Graph' icon selected under 'Graph Indicators'.

Gráficos são indicadores muito potentes do LabVIEW. Eles são altamente customizáveis, e podem ser usados consisamente para mostrar uma grande quantidade de informação.

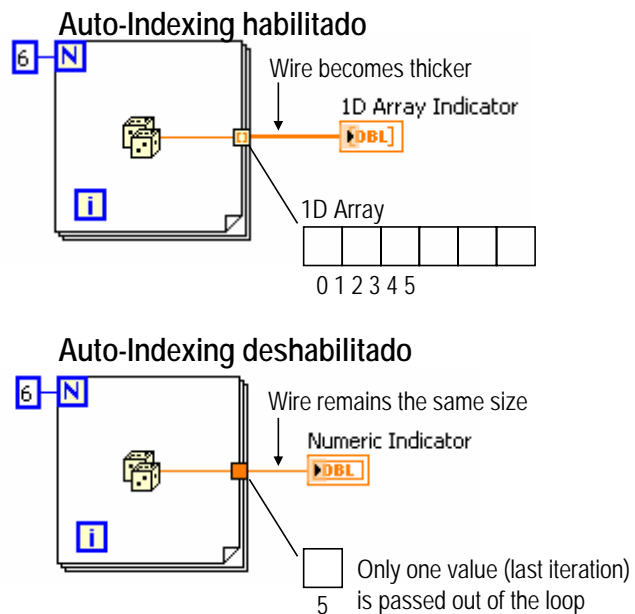
A página de propriedades permite que você mostre as configurações do tipo de gráfico, escala, e opções de cursor, e diversas outras opções do gráfico. Para abrir a página de propriedades, clique com o botão direito do mouse no gráfico no painel frontal e escolha **Properties**.

Gráficos permitem, também, que você crie gráficos para papéis técnico com a função “export simplified image”. Clique com o botão direito do mouse no gráfico, selecione **Data Operations»Export Simplified Image...**



Construindo Arrays com Loops (Indexação automática)

- Loops podem acumular arrays em suas bordas com indexação automática
- Loops For têm auto-index por default
- Loops While mostram somente o valor final por default
- Clique com o botão direito do mouse no tunel e habilite/desabilite o auto-indexing



ni.com

NATIONAL
INSTRUMENTS

Loops for e while podem ser indexados e acumular arrays em suas bordas. Isto é conhecido como indexação automática.

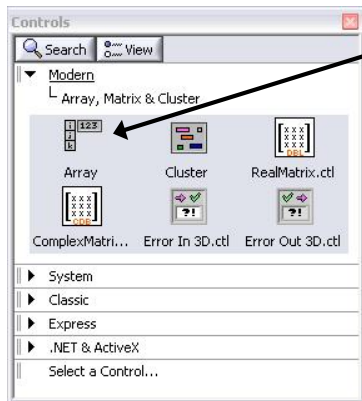
- O ponto de indexação na borda é conhecido com túnel.
- O loop for tem indexação automática habilitada por default.
- O loop while tem indexação automática desabilitado por default.

Exemplos:

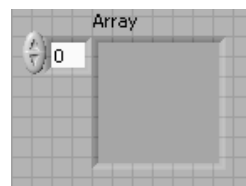
- Habilite a indexação automática para coletar valores dentro do loop para construir um array. Todos os valores são colocados em um array após a execução do loop.
- Desabilite a indexação automática se você estiver interessado somente no resultado final.

Criando um Array (Passo 1 de 2)

A partir do subpaleta **Controls»Modern»Array, Matrix, and Cluster**, selecione o ícone **Array**.



Coloque-o no painel frontal



ni.com

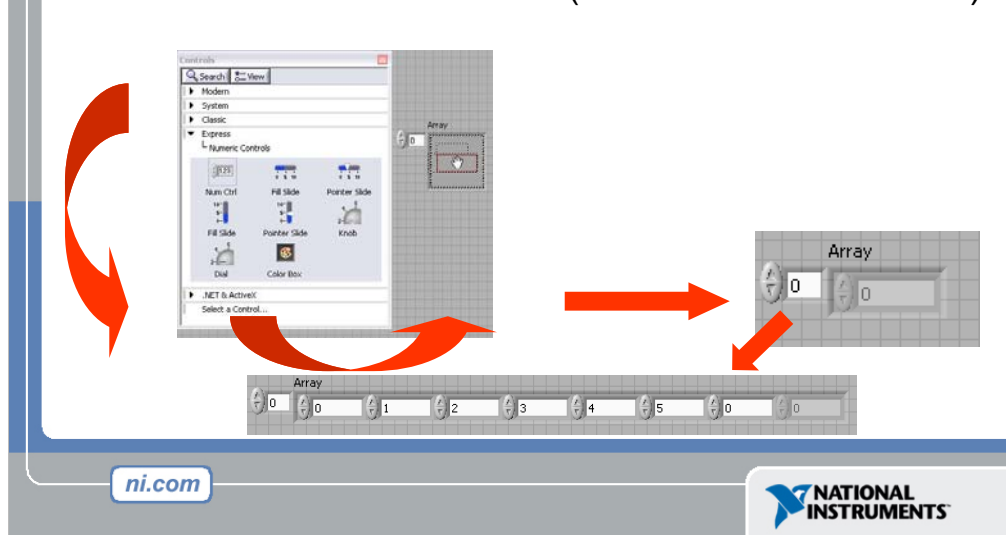
NATIONAL
INSTRUMENTS

Para criar um array de controles ou indicadores como mostrados, selecione um array no paleta **Controls»Modern»Array, Matrix, and Cluster**, coloque-o no Painel Frontal, e arraste um controle ou indicador em uma estrutura array. Se você tentar arrastar um controle ou indicador inválido como um gráfico XY, para dentro da estrutura, você não conseguirá coloca-lo dentro da estrutura.

Você deve inserir um objeto dentro da estrutura do array antes de usar o array no diagrama de blocos. Caso contrário, o terminal do array ficará preto e vazio.

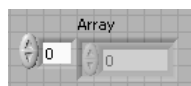
Criando um Array (Passo 2 de 2)

1. Coloque uma estrutura Array.
2. Insira dados na estrutura (i.e. Controle Numérico).

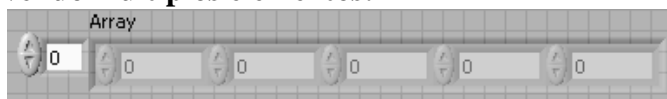


Para acrescentar dimensões a um array uma por vez, clique com o botão direito do mouse no display de índice e selecione **Add Dimension** do menu de atalhos. Você pode usar a ferramenta de posicionamento para redimensionar o tamanho do display de índice até conseguir a dimensão desejada

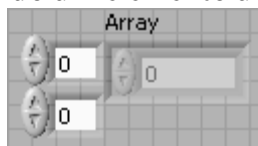
Array de 1D vendo um único elemento:



Array de 1D vendo múltiplos elementos:



Array 2D vendo um elemento único:



Array de 2D vendo múltiplos elementos:



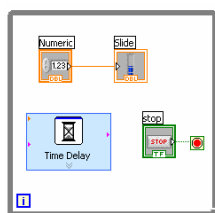
Como que crio um Time Loop?

1. Loop Time Delay

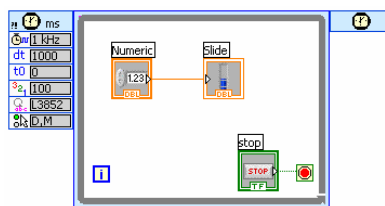
- Configure o Time Delay Express VI para esperar alguns segundos após cada interação (funciona para For e While loops).

2. Timed Loops

- Configure loops While especiais com dt específicos.



Time Delay



Timed Loop

ni.com

NATIONAL
INSTRUMENTS

Time Delay

O Time Delay Express VI atrasa a execução por um número especificado de segundos. Siga as regras de programação de fluxo de dados, o loop while não irá passar os dados adiante até que todas as tarefas dentro do loop tenham sido executadas, assim atrasando cada interação por alguns segundos.

Timed Loops

Executa cada interação como especificado previamente. Use o Timed Loop quando você quiser desenvolver VIs com capacidades de temporização com várias taxas, temporização precisas, feedback no loop de execução, caracterizações de temporização que mudam dinamicamente, ou vários níveis de prioridades de execução.

Dê um clique duplo no nó de entrada ou clique com o botão direito do mouse no nó de entrada e selecione **Configure Timed Loop** do menu de atalhos para ver a caixa Loop Configuration, onde você pode configurar o Timed Loop. Os valores que você entrar na caixa **Loop Configuration** que aparece como opções no nó de entrada.



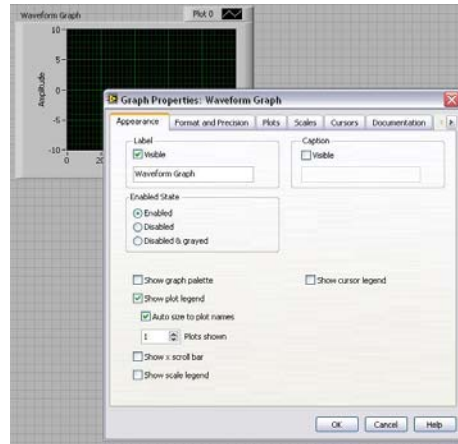
Wait Until Next ms Multiple

Espera até o valor de milissegundos do cronômetro seja um múltiplo especificado no **millisecond multiple**. Use esta função para sincronizar atividades. Você pode chamar esta função em um loop para controlar a taxa de execução do loop. É possível, porém, que o período do primeiro loop seja curto de mais. Esta função faz chamadas des-sincronizadas ao sistema, mas os nós funcionam de forma sincronizada. Assim, eles não completam a execução até que o tempo delimitado tenha passado

Functions»Programming»Timing»Wait Until Next ms Multiple

Propriedades de Controles & Indicadores

- Propriedades são características ou qualidades de um objeto
- Propriedades são encontradas clicando com o botão direito do mouse em controles ou indicadores
 - Propriedades Incluem:
 - Tamanho
 - Cor
 - Tipo e cor do gráfico
 - Características Incluem:
 - Cursores
 - Escalas



ni.com

NATIONAL
INSTRUMENTS

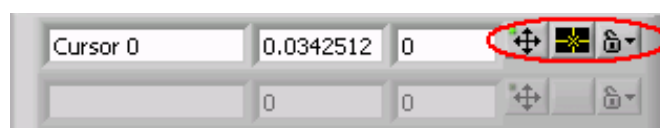
Propriedades são todas as qualidades dos objetos do painel frontal. Com propriedades, você pode configurar ou ler características como pano da frente e pano de fundo, formatação de dados e precisão, visibilidade, texto descritivo, tamanho, e lugar no painel frontal, e etc.



Exercício 3.1 – Analise Manual (Track A, B, & C)

Crie um VI que mostra os dados simulados no gráfico de formato de onda e meça a frequência e amplitude dos dados. Use o cursor do gráfico para verificar as medidas de frequência e amplitude.

1. Abra Exercício 2.1 – Simulated.vi.
2. Salve o VI como “Exercício 3.1 – Manual Analysis.vi”.
3. Vá para o Diagrama de Blocos e retire o loop while. Clique com o botão direito do mouse no canto do loop e escolha **Remove While Loop** para que o código dentro do loop não seja deletado
4. Delete o botão stop.
5. No Painel Frontal, troque a tabela de formato de onda por um gráfico. Clique com o botão direito do mouse na tabela e selecione **Replace»Modern»Graph»Waveform Graph**.
6. Faça a legenda do cursor disponível no gráfico. Clique com o botão direito do mouse no gráfico e selecione **Visible Items»Cursor Legend**.
7. Mude o valor máximo do “Frequency In” para 100. De um clique duplo no valor máximo e digite 100 uma vez que o texto estiver selecionado.
8. Ajuste o valor default do “Frequency In” ajustando o controle para o valor que você quiser, Clicando com o botão direito do mouse no controle e selecionando **Data Operations»Make Current Value Default**.
9. Execute o VI e observe o sinal do gráfico de formato de onda. Se você não conseguir ver o sinal, você pode precisar desligar a escalação automática do eixo-x. Clique com o botão direito do mouse no gráfico e selecione **X Scale»AutoScale X**.
10. Mude a frequência do sinal para que você possa ver alguns periodos do gráfico
11. Meça manualmente a frequência e amplitude do sinal no gráfico com os cursores. Para visualizar os cursores no gráfico, clique em um dos três botões da legenda do cursor. Uma vez que você conseguir ver os cursores, arraste estes pelo gráfico e suas coordenadas serão mostradas na legenda.



12. Lembre-se que a frequência de um sinal é inversamente proporcional a seu período ($f = 1/T$). Suas medidas se igualam aos indicadores do Tone Measurements VI?
13. Salve e feche seu VI.

Nota: A solução deste exercício está no final deste manual

(Fim do Exercício)

Matemática Textual no LabVIEW

- Integre scripts existentes com LabVIEW para desenvolvimento mais rápido
- Ambiente interativo, fácil de usar, hands-on
- Desenvolva algoritmos, explore conceitos matemáticos, e analise resultados usando um único ambiente
- Liberdade para escolher a sintaxe mais eficaz, seja ela textual ou gráfica dentro do mesmo VI

Ferramentas matemáticas suportadas:

MathScript

MathSoft software

Mathematica software

MATLAB® software

Maple software

Xmath software

ni.com

MATLAB® is a registered trademark of The MathWorks, Inc.



Revisão

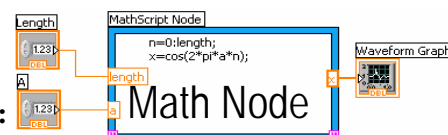
Com o lançamento do LabVIEW 8 da National Instruments, você tem a liberdade de escolher a sintaxe mais eficaz para computação técnica, se você estiver desenvolvendo algoritmos, explorando conceitos DSP, ou analisando resultados. Você pode instrumentar seus scripts e desenvolver algoritmos no diagrama de Blocos interagindo com softwares de terceiros como o MathWorks Inc. MATLAB software, Mathematica, Maple, Mathcad, IDL e Xmath. Uso destas ferramentas com o LabVIEW é alcançado de várias formas dependendo de seu programa listado abaixo:

Nó matemático textual do LabVIEW :

Nó MathScript , Nó de Formulas

Comunicação com outros softwares através de nós do LabVIEW:

Nó Xmath , nó MATLAB script, nó Maple*,nó IDL*



Comunicação com softwares através do VI Server:

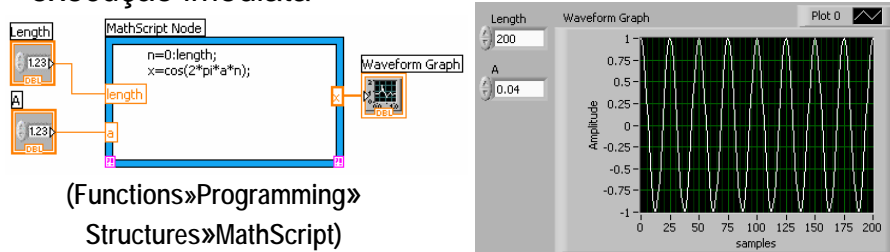
VIs Mathematica* , e VIs Mathcad*

No LabVIEW 8, você pode combinar a programação gráfica intuitiva do LabVIEW com o MathScript, uma linguagem textual orientada para matemática que é normalmente compatível com a linguagem popular m-file.

*toolkit LabVIEW especifica a ferramenta matemática deve ser instalada.

Matemática com o Nó MathScript

- Implemente equações e algoritmos textualmente
- Variáveis de entrada e saída criadas nas bordas
- Geralmente compatível com a linguagem popular m-file
- Termine declarações com ponto e vírgula para evitar execução imediata



Prototipe suas equações na janela interativa do **MathScript**.

ni.com

NATIONAL
INSTRUMENTS

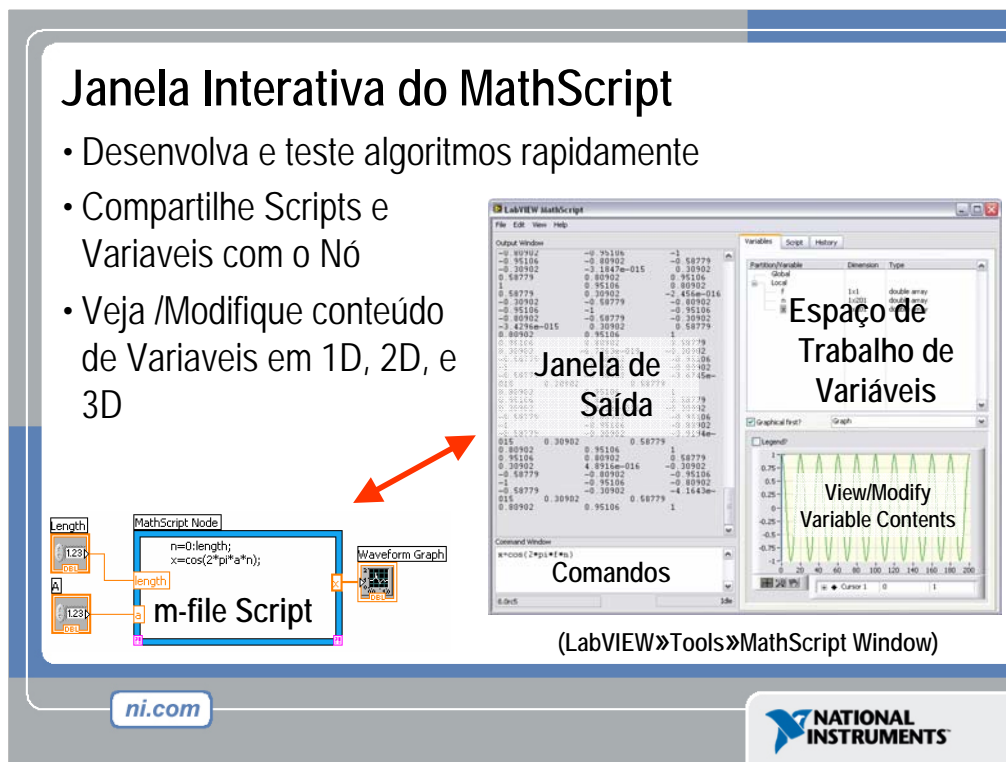
O nó MathScript melhora o LabVIEW acrescentando uma linguagem baseada em texto para algoritmos matemáticos no ambiente de programação gráfico do LabVIEW. Scripts M-file que você escreve e salvou na janela MathScript pode ser aberto no nó MathScript. Scripts m-file escritos em outros programas normalmente pode ser executadas no nó MathScript. O MathScript permite que você escolha a sintaxe que você se sentir mais confortável para resolver problemas. Equações podem ser instrumentadas com o MathScript para parâmetros de exploração, simulação, e lançamento em uma aplicação final.

O nó MathScript:

- Localizado no subpaleta **Programming»Structures**.
- Caixa redimensionável para entrar com computações textuais diretamente no Diagrama de Blocos.
- Para acrescentar variáveis, Clique com o botão direito do mouse e escolha **Add Input** ou **Add Output**.
- Nome de variáveis como elas são usadas na fórmula. (Nomes são diferenciáveis por letras maiúsculas e minúsculas).
- O tipo de dados de saída podem ser mudados clicando com o botão direito do mouse no nó de entrada ou saída.
- Declarações devem terminar com ponto e vírgula para evitar a execução imediata.
- Habilidade de importar & exportar m-files clicando com o botão direito do mouse no nó.

Janela Interativa do MathScript

- Desenvolva e teste algoritmos rapidamente
- Compartilhe Scripts e Variáveis com o Nó
- Veja /Modifique conteúdo de Variáveis em 1D, 2D, e 3D



A janela MathScript traz um ambiente interativo onde equações podem ser prototipadas e cálculos podem ser feitos. A janela e o Nó MathScript compartilham uma sintaxe comum e variáveis globais, fazendo uma mudança perfeita de prototipagem para a implementação. Os dados da janela de prévia traz um modo conveniente de ver dados variáveis como números, gráficos e áudio (com suporte da placa de som).

Ajuda para o MathScript

Ajuda para o ambiente pode ser acessada usando a Janela interativa do MathScript. Digite **Help** na janela de comandos para uma introdução ao help do MathScript. **Help** seguido de uma função mostrará ajuda específica a esta função.


Características da janela interativa MathScript:

- Prototipe equações e fórmulas através da janela
- Acesse arquivos de ajuda de funções facilmente digitando **Help <function>** na janela de comandos
- Selecione uma variável para mostrar seus dados na janela de prévia e até ouvir os resultados
- Escreva, salve, armazene, e execute m-files usando a aba script.
- Compartilhe dados entre o nó MathScript no LabVIEW e a Janela MathScript usando variáveis globais.
- Gráficos avançados e exportação de imagens

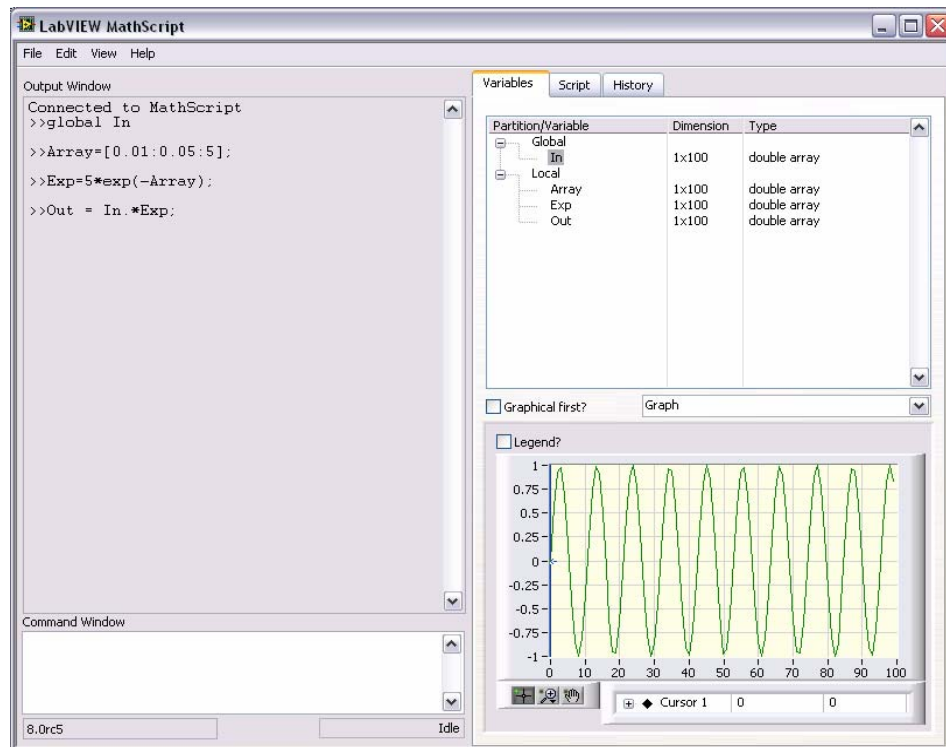


Exercício 3.2 – MathScript (Track A, B, & C)

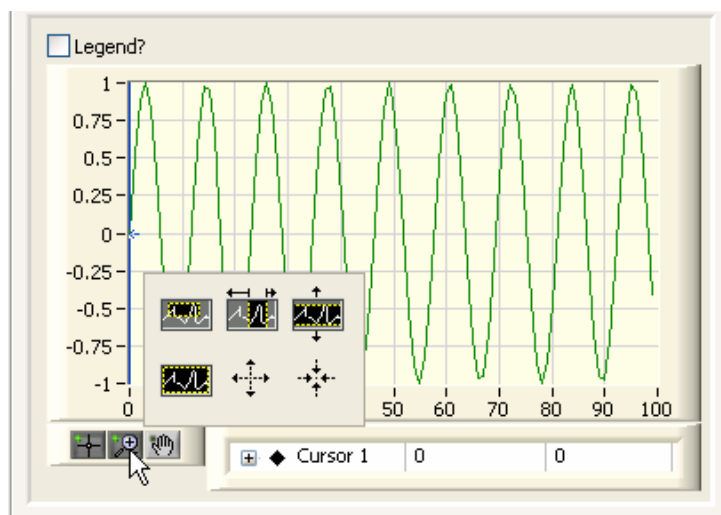
Crie um VI que usa o MathScript para alterar seu sinal simulado e plotar um gráfico do mesmo. Use a janela interativa do MathScript e altere os dados e carregue o script que você criou no nó MathScript.

1. Abra Exercício 3.1 – Manual Analysis.vi.
2. Salve o VI como “Exercício 3.2 – MathScript.vi”.
3. Vá ao Diagrama de Blocos e apague o fio conectando o Simulate signal ao Waveform Graph
4. Coloque o nó MathScript (**Programming»Structures»MathScript Node**).
5. Clique com o botão direito do mouse na borda esquerda do nó MathScript e selecione **Add Input**. Nomeie esta entrada de “In” digitando quando o nó estiver selecionado e preto.
6. Clique com o botão direito do mouse na borda direita do nó MathScript e selecione **Add Output**. Nomeie esta saída de “Out”.
7. Converta a saída Dynamic Data Type do Simulate Signals VI para um array de 1D de escalares para a entrada do nó MathScript. Coloque um Convert from Dynamic Data Express VI no Diagrama de Blocos (**Express»Signal Manipulation»Convert from Dynamic Data**). Por default, o VI é configurado corretamente então clique “OK” na janela de configuração.

8. Ligue a saída “Sine” do Simulate Signal VI à entrada “Dynamic Data” do Convert from Dynamic Data VI.
9. Ligue a saída “Array” do Convert from Dynamic Data VI ao nó “In” do nó MathScript.
10. Para usar os dados do Simulate Signal VI na Janela Interativa do MathScript é necessário declarar a variável de entrada como uma variável global. Dentro do Nó MathScript digite “global In;”.
11. Retorne ao Painel Frontal e aumente a frequência para um valor entre 50 e 100. Execute o VI.
12. Abra a Janela Interativa do MathScript(**Tools»MathScript Window...**).
13. Na Janela Interativa do MathScript, a janela de comandos pode ser usada para digitar os comandos desejados. Digite “global In” e aperte “Enter”. Isto permitirá que você veja os dados que passam pela variável “In” no nó MathScript.

14. Veja que todas as variáveis declaradas no script junto com suas dimensões e tipos estão listadas na aba “Variables”. Para ver os dados plotados, clique uma vez na variável In e mude o a opção do menu de “Numeric” para “Graph”.

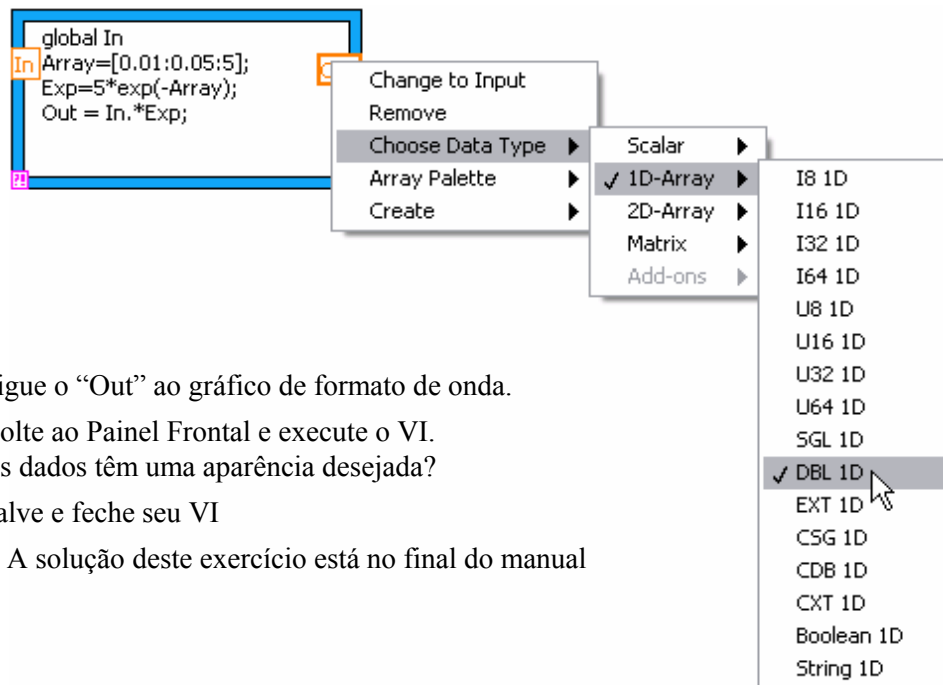


15. Use o paleta do gráfico para aumentar o zoom em seus dados.



16. Clique com o botão direito do mouse em “Cursor 1” e escolha **Bring to Center**. O que que isto faz?
17. Arraste o cursor pelo gráfico. O cursor não se moverá com o zoom selecionado.
18. Clique com o botão direito do mouse no gráfico e escolha **Undock Window**. O que que isto faz? Feche esta nova janela quando você tiver terminado.

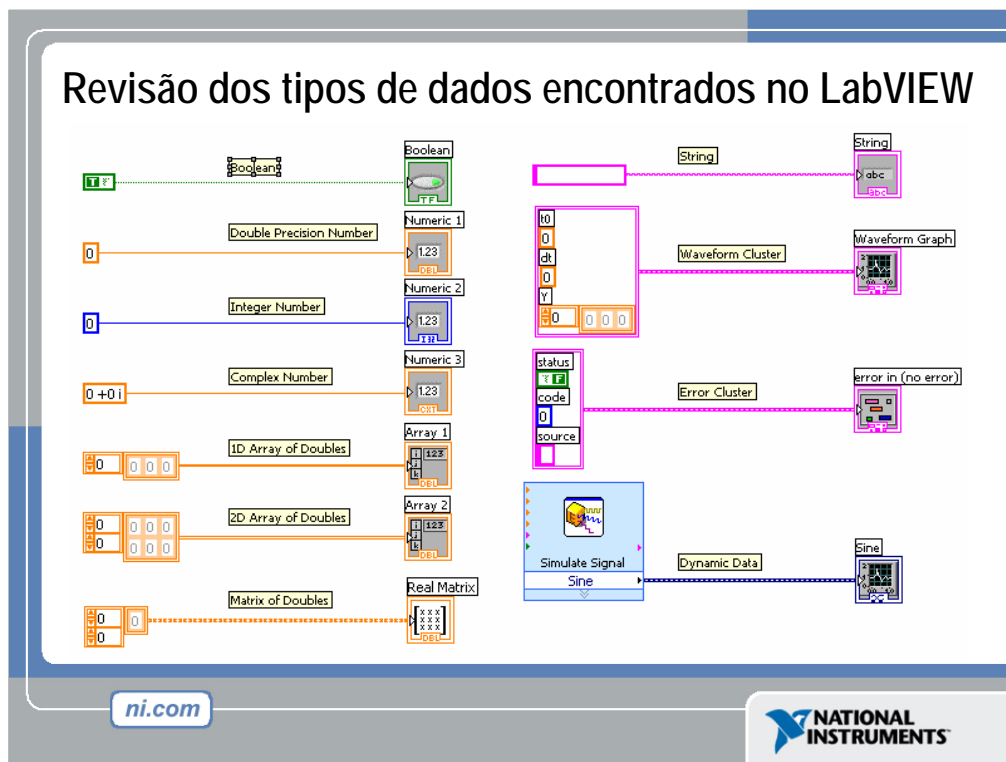
19. Multiplique os dados por um decaimento exponencial. Siga estes passos:
 - a. Crie um array de 100 elementos de dados que constitue uma função rampa indo de 0.01 a 5 digitando “Array = [0.01:0.05:5];” na janela de comandos e apertando “Enter”. Que tipo de variável é o “Array”?
 - b. Crie um array contendo o decaimento exponencial. Digite :
“Exp = 5*exp(-Array);” e aperte “Enter”.
 - c. Agora multiplique os arrays Exp e In elemento por elemento digitando
“Out = In.*Exp;” e aperte “Enter”.
 - d. Veja o gráfico da variável “Out”.
20. Vá a aba History e use o Ctrl-clique para escolher os 4 comandos que você acabou de digitar. Copie estes comandos usando <Ctrl-C>.
21. No tab Script, cole os comandos no editor de scripts usando<Ctrl-V>.
22. Salve seu script clicando em “Save” na parte inferior da janela . Salve como “myscript.txt”
23. Feche a Janela MathScript.
24. Volte para o Diagrama de Blocos do Exercício 4.2 – MathScript. Carregue o script que você acabou de criar clicando com o botão direito do mouse na borda do nó MathScript e selecionando **Import...** Navegue até myscript.txt, selecione-o, e clique “OK”.
25. Clique com o botão direito do mouse na variavel “Out” e selecione **Choose Data Type**
Type»1D-Array»DBL 1D. Tipos de dados de saída precisam ser ajustados manualmente no MathScript.



26. Ligue o “Out” ao gráfico de formato de onda.
27. Volte ao Painel Frontal e execute o VI.
Os dados têm uma aparência desejada?
25. Salve e feche seu VI

Nota: A solução deste exercício está no final do manual

(Fim do Exercício)



LabVIEW utiliza diversos tipos de dados. Estes incluem:

Booleanos, Numérico, Arrays, Strings, Clusters, e mais.

A cor e símbolo de cada terminal indica o tipo de dado do controle ou indicador. Terminais de controle têm uma borda mais grossa. Além disso, as setas aparecem nos terminais do painel frontal para indicar se o terminal é controle ou indicador. Uma seta aparece no lado direito se for controle, e no lado esquerdo se for indicador.

Definições

- **Array:** Arrays agrupam dados do mesmo tipo. Um array consiste em elementos e dimensões. Elementos são dados que constituem o array. A dimensão é o comprimento, altura, ou profundidade do array. Um array pode ter mais de uma dimensão e até (2^{31}) – elementos 1 por dimensão, se a memória permitir.
- **Cluster:** Clusters agrupam dados de elementos de diversos tipos, com um grupo de fios dentro de um cabo de telefone, onde cada fio representa um elemento diferente do cluster.

Veja **Help»Search the LabVIEW Help...** para mais informações. O *LabVIEW User Manual* no site ni.com traz referências adicionais para tipos de dados do LabVIEW.

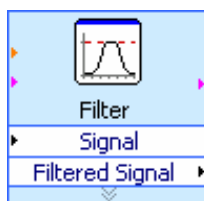


Exercício 3.3 – Aplique o que você aprendeu(Track A, B, & C)

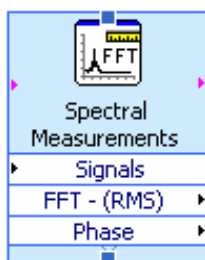
Neste exercício, você irá criar um VI que usa o que você aprendeu até aqui.

Desenvolva um VI que faz o seguinte:

1. Adquire dados de seu dispositivo e plota um gráfico (do seu dispositivo DAQ, ou dispositivo simulado ou placa de som).
2. Filtra estes dados usando o Filter Express VI (**Functions»Express»Signal Analysis»Filter**). Deve haver um controle no painel frontal para definir uma frequência de corte.



3. Faça um Fast Fourier Transform (FFT) para obter informações a respeito da frequência de seus dados filtrados e plote um gráfico com os resultados. Use o Spectral Measurements Express VI (**Functions»Express»Signal Analysis»Spectral**).



4. Encontre a frequência dominante de seus dados filtrados usando o Tone Measurements Express VI.
5. Compare esta frequência àquela escolhida pelo usuário. Se a frequência estiver acima deste limite, ascenda um LED. Se você estiver usando o USB-6009, ascenda o LED de seu hardware usando o DAQ Assistant. Você precisará inverter a linha digital para ascender o LED quando o valor passar do limite. Você pode especificar isto na janela de configuração do DAQ Assistant ou com uma função booleana “not”.
6. Se você se atrapalhar, abra a solução ou veja a mesma no final deste manual.

(Fim do Exercício)

Seção IV – Tópicos Adicionais do LabVIEW

- A. Tipos Adicionais de Dados
 - Cluster
- B. Construções de Fluxo de Dados
 - Shift Register
- C. SubVIs
- D. Máquinas de Estados
- E. Variáveis Locais
- F. Produtor/Consumidor

ni.com



Introdução a Clusters

- Estrutura de dados de agrupamento
- Dados podem ser de tipos diferentes
- Análogo ao *struct* em C
- Elementos precisam ser todos indicadores ou todos controles
- Pense como fios dentro de um cabo
- **Ordem é importante**



ni.com

NATIONAL
INSTRUMENTS

Clusters agrupam componentes iguais e diferentes. Eles são equivalentes ao record em Pascal e struct em C.

Componentes do cluster podem ser de tipos diferentes de dados.

Exemplos:

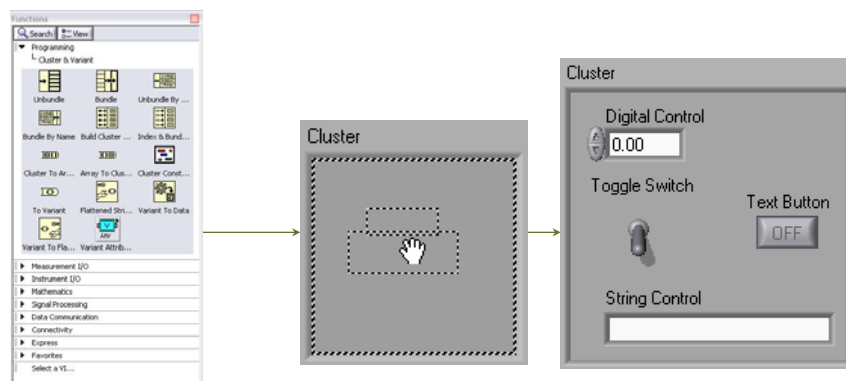
- Informação de erros — Agrupamos bandeiras booleanas de erro, um código numérico de erro, e uma string indicando a fonte do erro para especificar o erro exato.
- Informação de usuário—Agrupamos uma string com o nome do usuário, e um número de identificação para especificar seu código de segurança.

Todos os elementos do cluster precisam ser ou controles ou indicadores. Você não pode ter um controle de string e um indicador booleano no mesmo cluster. Pense em clusters como um agrupamento de fios individuais em um único cabo.

Criando um Cluster

1. Selecione uma estrutura cluster.
2. Coloque objetos dentro da estrutura.

Controls»Modern»Array, Matrix & Cluster



Um cluster de objetos no painel frontal pode ser criado escolhendo **Cluster** do paleta **Controls»Modern»Array, Matrix & Cluster**.

- Esta opção traz a estrutura do cluster (parecida com a estrutura do array).
- Você pode dimensionar a estrutura cluster ao colocá-la no painel frontal.
- Clique com o botão direito do mouse dentro da estrutura e acrescente objetos de qualquer tipo.

Nota: Você pode ter inclusive um cluster de clusters.

O cluster se torna um cluster de controles ou indicadores dependendo do tipo de dados que forem colocados dentro do mesmo.

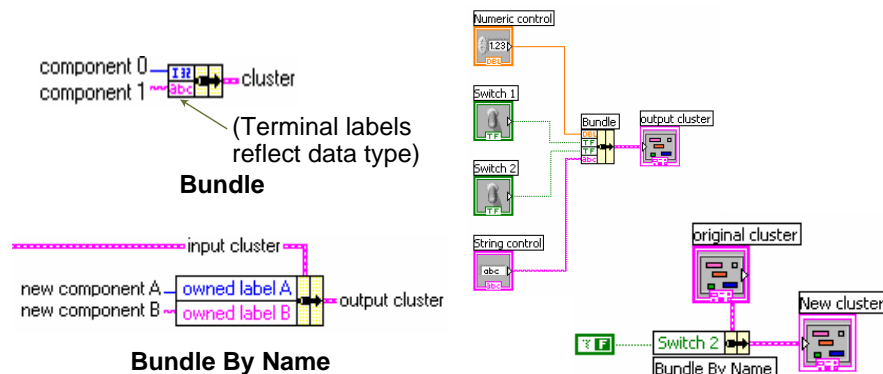
Você pode criar um cluster de constantes no diagrama de blocos usando o **Cluster Constant** localizado no paleta **Cluster**.

- Isto traz uma estrutura vazia.
- Você pode dimensionar o cluster ao colocá-lo no diagrama de blocos.
- Coloque as constantes dentro da estrutura.

Nota: Você não pode colocar terminais para o painel frontal em um cluster de constantes, nem colocar constantes “especiais” como constantes de aba ou strings vazias dentro do cluster de constantes.

Funções Cluster

- No sub-paleta **Cluster & Variant** do palette **Programming**
- Também pode ser acessado clicando com o botão direito do mouse no terminal cluster



ni.com

NATIONAL
INSTRUMENTS

Os termos Bundle e Cluster são fortemente relacionados no LabVIEW.

Exemplo: Você usa uma função Bundle para criar um cluster. Você usa uma função Unbundle para extrair os objetos do cluster.

Função Bundle —Forma um cluster contendo os dados objetos (explique o exemplo).

Função Bundle by Name—Atualiza os valores dos objetos específicos do cluster (o objeto deve ter seu próprio rótulo).

Nota: Você precisa ter um cluster existente ligado no terminal central da função Bundle by Name.

Usando Arrays e Cluster com gráficos

O tipo de dado Waveform contém 3 tipos de dados:

- t_0 = Tempo inicial
- dt = Tempo entre Samples
- Y = Array de magnitudes de Y

Duas formas de criar um cluster de waveform:



ni.com

NATIONAL INSTRUMENTS

O tipo de dado waveform traz os dados, tempo de início e Δt do formato de onda. Você pode criar formatos de onda usando a função Build Waveform. Vários VIs e funções que você usa para adquirir ou analisar formas de onda aceitam e retornam este tipo de dado por default. Quando você liga um tipo de dado waveform a um gráfico ou uma tabela de formato de onda, estes plotam a onda automaticamente baseado nos dados, tempo inicial e Δx da onda. Quando você liga um array de dados waveform a um gráfico ou tabela, estes automaticamente plotam todas as ondas disponíveis.

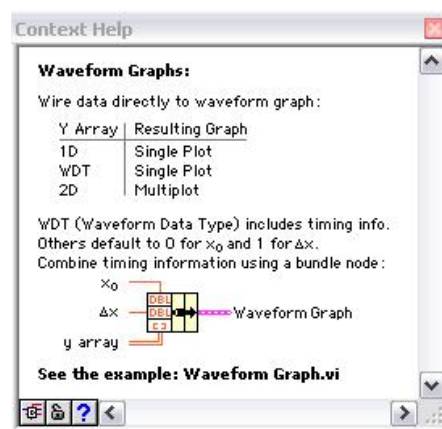
Build Waveform

Cria uma onda ou modifica uma onda existente com tempo inicial representado como um TimeStamp absoluto.

TimeStamps são horários e datas do mundo real e são muito úteis para gravação de dados.

Bundle

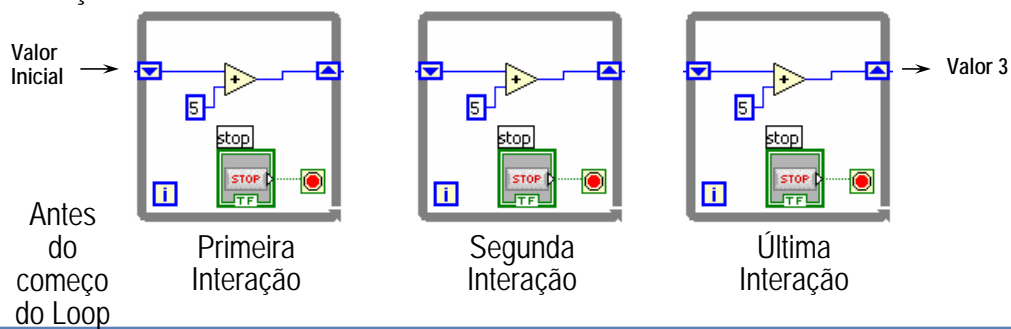
Cria uma onda ou modifica uma onda existente com um timestamp relativo. A entrada para t_0 é um DBL. Criando uma onda usando o bundle permite que dados sejam plotados em um eixo X (tempo) negativo.



Shift Register – Acesse Dados anteriores do loop

Disponível à esquerda e direita da estrutura do loop

- Clique com o botão direito do mouse na borda e selecione **Add Shift Register**
- Terminal da direita guarda informação no final de cada interação
- Terminal da esquerda fornece informação armazenada para a próxima interação



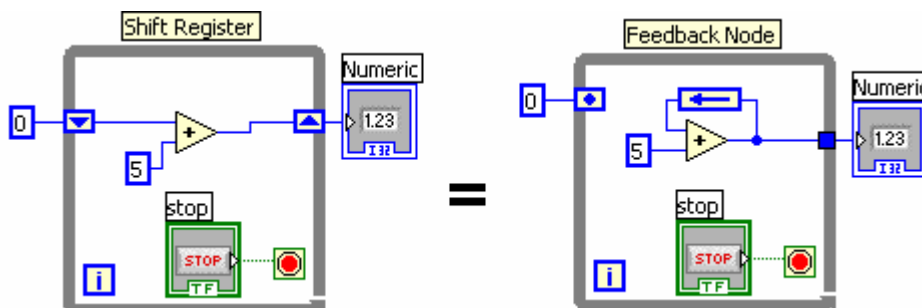
ni.com

NATIONAL INSTRUMENTS

Shift registers transfer data from one iteration to the next:

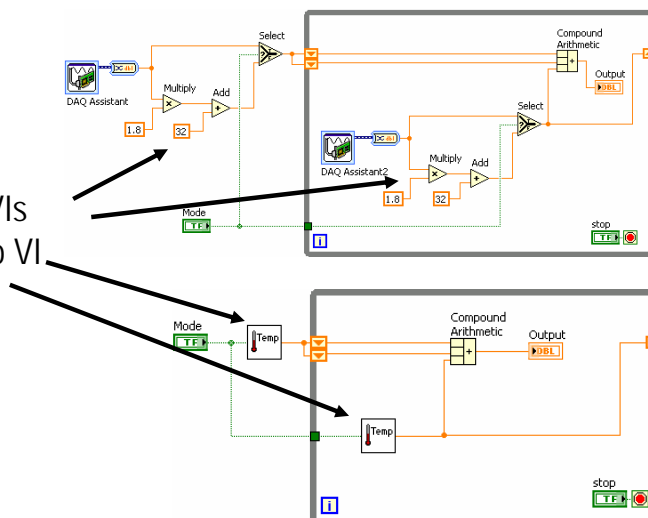
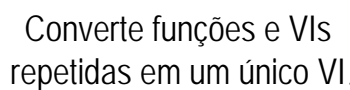
- Clique com o botão direito do mouse no lado esquerdo ou direito do loop for ou while para acrescentar um Shift Register.
- O terminal direito armazena dados no final da interação. Os dados aparecem no terminal esquerdo no começo da próxima interação.
- O shift register se adapta ao tipo de dado ligado a ele.

Uma entrada de 0 resultará em uma saída de 5 no final da primeira interação, 10 após a segunda, e 15 após a terceira interação. Dito de outra forma, um shift register é usado para reter o valor de uma interação até a próxima. Eles são muito úteis para aplicações que tem memória ou feedback entre estados. O nó de feedback é outra representação do mesmo conceito (mostrado abaixo) Ambos os programas tem o mesmo comportamento.



Veja **Help»Search the LabVIEW Help...** para mais informações.

Modularidade no LabVIEW – SubVIs



nj.com



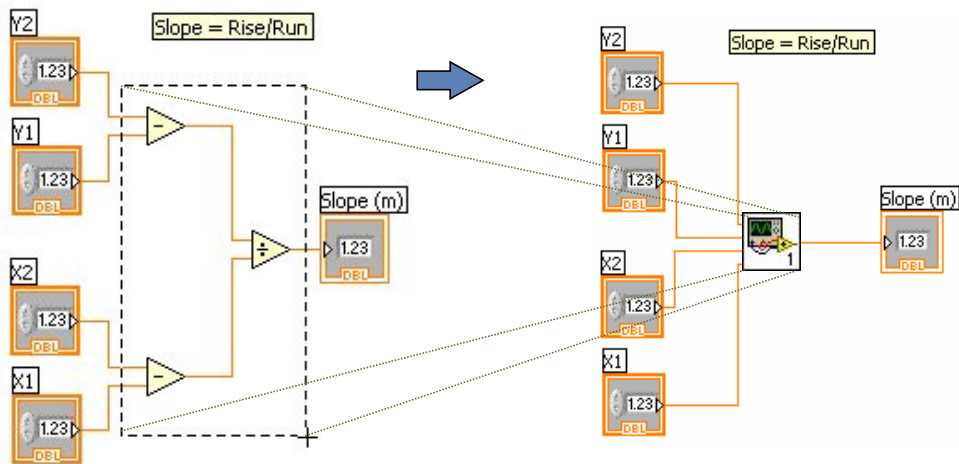
Modularidade define o grau em que seu VI é composto por componentes discretos, tais que uma mudança a um tem impacto mínimo nos demais componentes. No LabVIEW estes componentes separados são chamados de subVIs. Criar um subVIs com seu código aumenta a legibilidade e a capacidade de reutilização de seus VIs.

Na imagem superior, repetimos um código para permitir que o usuário trabalhe com diversas escalas de temperatura. Como a porção de código é idêntica para ambos os casos, criamos um subVI para ela. Isto tornará o código mais legível, por ser menos poluído, e permitirá que reaproveitamos o código facilmente. Como podemos ver, o código ficou muito menos bagunçado, e alcança a mesma finalidade e se for necessário, a escala de temperatura pode ser reutilizada em outras aplicações.

Qualquer porção de código LabVIEW pode se tornar um subVI que por sua vez pode ser usado por outros códigos do LabVIEW.

Crie SubVI

- Cerque a área a ser convertida em subVI.
- Selecione **Edit»Create SubVI** do Menu Edit.



ni.com

NATIONAL
INSTRUMENTS

Criando um SubVIs

Um nó de subVI corresponde a uma chamada de sub-rotina em linguagem de programação textual. O diagrama de blocos que contém vários subVIs idênticos chama o mesmo subVI diversas vezes.

Os controles e indicadores do subVI recebem e retornam informação para o diagrama de blocos do VI principal. Clique no ícone **Select a VI** ou texto no paleta de funções, navegue, e clique duas vezes em um VI, coloque este no diagrama de blocos para criar um subVI para que chama este VI.

Os terminais de entrada e saída e o ícone de um subVI são facilmente customizáveis. Siga as instruções abaixo para criar um subVI rapidamente.

Criando um SubVIs a partir de seções de um VI

Transforme uma seção de seu VI em um subVI usando a ferramenta de posição para selecionar a seção de seu diagrama de blocos e selecione **Edit»Create SubVI**. Um ícone para o novo subVI tomará o lugar do código selecionado no diagrama de blocos. O LabVIEW cria controles e indicadores para seu novo subVI, automaticamente configura o painel de conectores baseado no número de controles e indicadores selecionados e liga o subVI a fios existentes.

Veja **Help»Search the LabVIEW Help...»SubVIs** para mais informações.

Funções e SubVIs do LabVIEW operam como Funções em outras linguagens

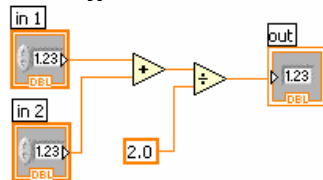
Função Pseudo Code

```
function average (in1, in2, out)
{
  out = (in1 + in2)/2.0;
}
```

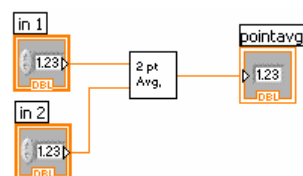
Chamando programa Pseudo Code

```
main
{
  average (in1, in2, pointavg)
}
```

SubVI Diagrama de Blocos



Chamando VI para o Diagrama de Blocos



ni.com

NATIONAL
INSTRUMENTS

Um nó do subVI corresponde uma chamada de subrotina em uma linguagem de programação baseada em texto. O nó não é o subVI, somente uma chamada de subrotina em um programa não é a subrotina propriamente dito. O diagrama de blocos que contém vários nós identicos de subVIs chama o mesmo subVI diversas vezes. A abordagem modular faz com que aplicações sejam mais fáceis de depurar e manter. A funcionalidade do subVI não importa para este exemplo. O ponto importante é transformar duas entradas numéricas em uma única entrada.

Ícone e Painel de Conexões

- Use este layout de painel como padrão
- Terminais superiores são reservados normalmente para referencias, como referencias de arquivo
- Terminais inferiores são reservados para clusters de erros

ni.com

NATIONAL INSTRUMENTS

O ícone e painel de conexões permite que você defina os dados sendo transferidos para dentro e para fora do subVI além de sua aparência do no código principal do LabVIEW. Cada VI tem um ícone no canto superior direito do painel frontal e diagrama de blocos. Depois de construir seu VI, crie um ícone e painel conector para que você possa usar o VI como subVI.

O ícone e painel de conexões corresponde ao prototipo da função em linguagem baseada em texto. Há diversas opções para o painel de conexões, mas alguns padrões gerais estão exemplificadas a cima. Em geral, reservar os terminais superiores para referencias e os inferiores para cluster de erros.

Para definir um painel de conexões, clique com o botão direito do mouse no ícone e selecione **Show Connector** no menu de atalhos. Cada retângulo no painel de conexões representa um terminal. Use os terminais para atribuir entradas e saídas. Selecione padrões diferentes clicando com o botão direito do mouse e selecionando **Patterns** no menu de atalhos.

Ícone e Painel de Conexões – Crie um Ícone

- Crie ícones customizados clicando com o botão direito do mouse no ícone no canto superior direito da tela do painel frontal ou diagrama de blocos e selecionando **Edit Ícone** ou clicando duas vezes no ícone
- Você pode arrastar gráficos de qualquer lugar de seu sistema de arquivos e soltar para usar como ícone
- Refira ao [Ícone Art Glossary](http://ni.com) at ni.com para gráficos padrões para usar como ícone de seu VI



ni.com

**NATIONAL
INSTRUMENTS**

Um ícone é uma representação gráfica do VI. Se você usar um VI como subVI, o ícone identifica o subVI no diagrama de blocos. O editor de ícones é uma utilidade que vem embutido no LabVIEW 8 e permite usuários a customizar totalmente a aparência de seus subVIs. Isto permitirá que programadores distinguem visualmente seus subVIs e aumentarão a usabilidade de subVIs em grandes porções de código.

Depois de ter definido o painel de conexões e customizar seu ícone, você está pronto para colocar seu subVI em outro código de LabVIEW. Há duas formas de fazer isto:

Para colocar um subVI no Diagrama de Blocos

1. Clique no botão **Select a VI** no paleta de funções
2. Navegue até o VI que você deseja usar como subVI
3. Dê um clique duplo para coloca-lo no diagrama de blocos

Para colocar e abrir um VI no diagrama de blocos de outro VI

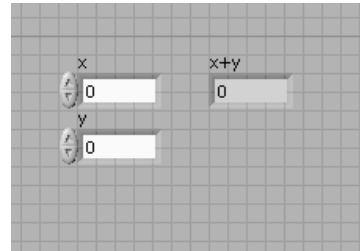
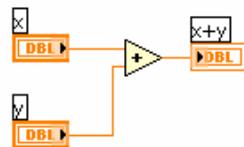
1. Use a ferramenta de posicionamento para clicar no ícone do VI que você deseja usar como subVI
2. Arraste o ícone para o diagrama de blocos de outro VI



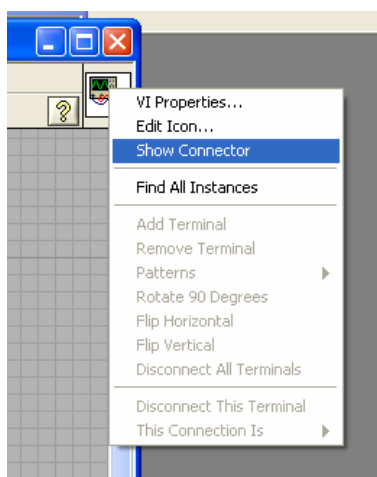
Exercício 4.1 – Criando um SubVI

Crie um subVI a partir de um VI novo, que soma duas entradas e fornece a soma

1. Abra um novo VI (**Ctrl+N**).
2. Coloque a função Add (**Programming » Numérico**) no diagrama de blocos.
3. Crie controles e indicadores clicando com o botão direito do mouse e selecionando **Create » Control** or **Indicator**. O diagrama de blocos e painel frontal devem parecer com os de baixo



4. No painel frontal clique com o botão direito do mouse no ícone no canto superior direito da janela e selecione **Show Connector** para revelar o painel de conexões.

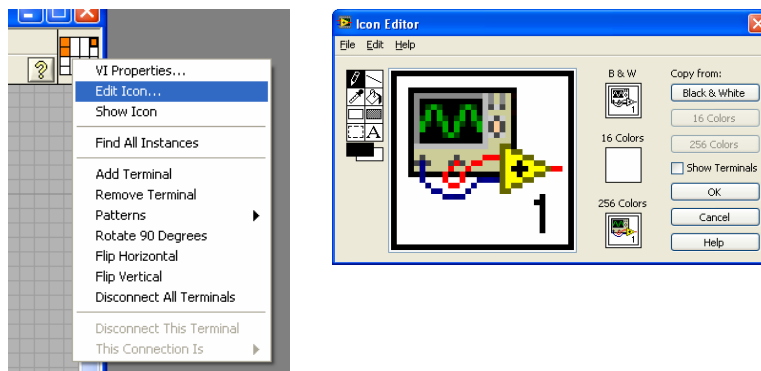


5. Atribua terminais aos dois controles e ao indicador clicando primeiro no terminal e depois no controle/indicador desejado

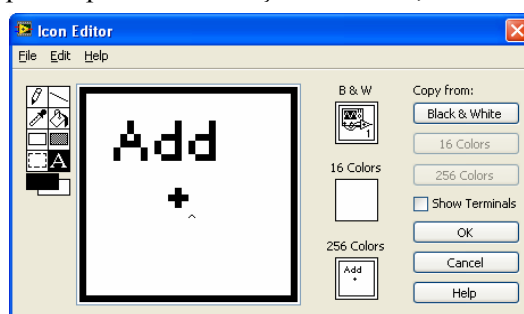
Nota: Convenção geral é ter os controles como dados de entrada no lado esquerdo e indicadores como saída no lado direito do ícone.



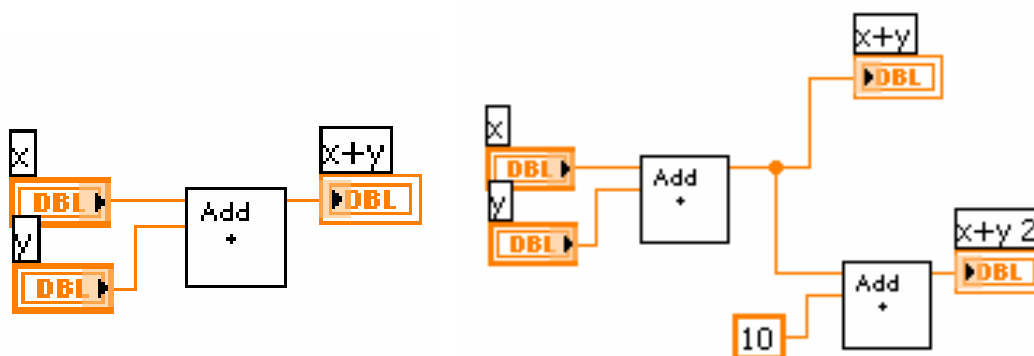
6. Clique com o botão direito do mouse no painel de conexões e selecione **Edit Ícone....**. Isto abrirá o editor.



7. Modifique os gráficos para representar a função do subVI, neste caso Soma.



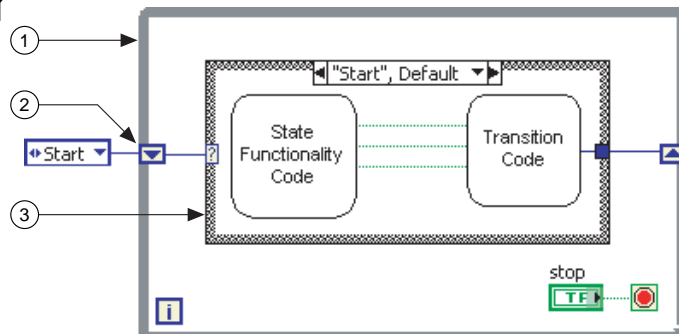
8. Salve oSubVI. Ele agora pode ser usado como qualquer outro VI para realizar qualquer função, neste caso a soma de dois números.



(Fim do Exercício)

Maquinas de Estados

- LoopWhile
- Estrutura Case
- Shift Register



ni.com

NATIONAL
INSTRUMENTS

Você usar uma maquina de estados para implementar um algoritmo que pode ser descrito com um diagrama de estados ou fluxograma. Uma máquina de estados consiste em uma série de estados e transições que te guiam para o próximo estado.

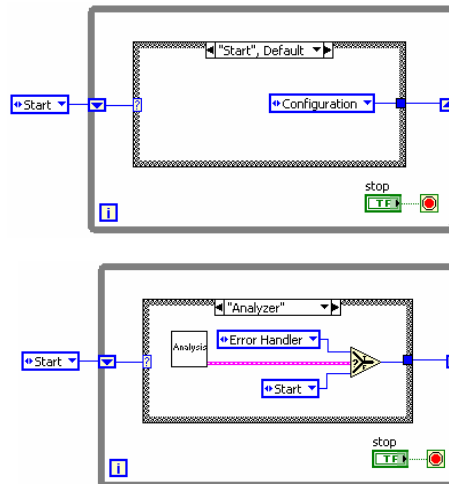
Cada estado leva a um ou mais estados ou encerra o fluxo.

Uma aplicação comum para máquinas de estados é para criar interfaces. Em uma interface de usuário, as ações do usuário mandam informação para a interface em outro segmento do processamento. Cada segmento haje como um estado.

Testes de processos é outra aplicação comum da máquina de estado. Para um teste de processo, um estado representa um segmento do processo. Dependendo do resultado de cada estado de teste, um estado diferente pode ser chamado.

Transições de Máquinas de Estados

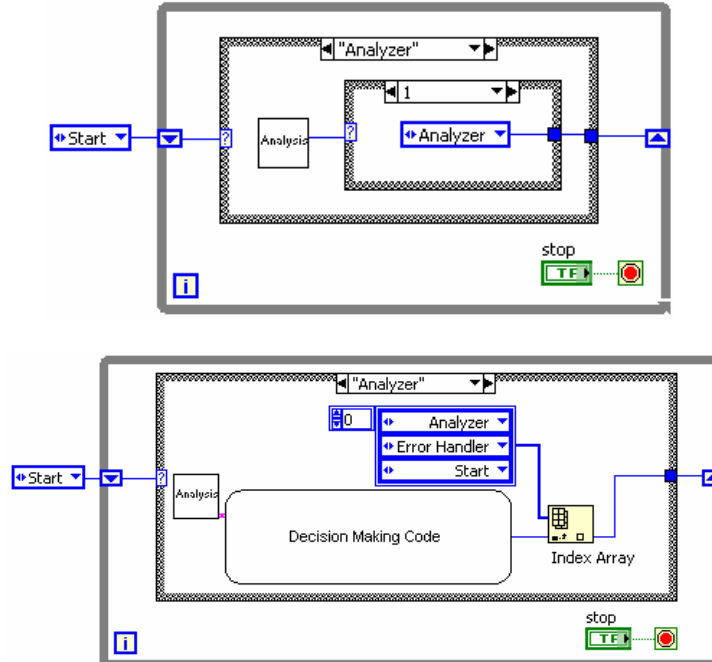
- Diversas técnicas de programação existe para fazer transições entre estados no LabVIEW usando Máquinas de Estados
- Transição padrão implica que um estado, sempre será seguido de outro
- Transições entre dois estados potenciais pode ser gerenciado pela função Select



ni.com

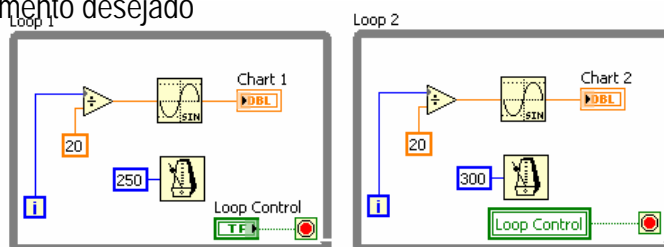
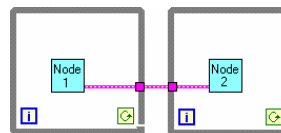
NATIONAL INSTRUMENTS

Se um estado pode transitar para vários estados potenciais, uma estrutura case pode ser usada. Outra abordagem é usar um array de estados possíveis e permitir que o código escolha o melhor estado para passar para o shift register.



Comunicação entre loops

- Comunicação entre loops usando fluxo de dados é impossível
 - O loop esquerdo sempre completará antes do direito
- Variáveis são necessárias para comunicação entre fios quando estes não têm um comportamento desejado



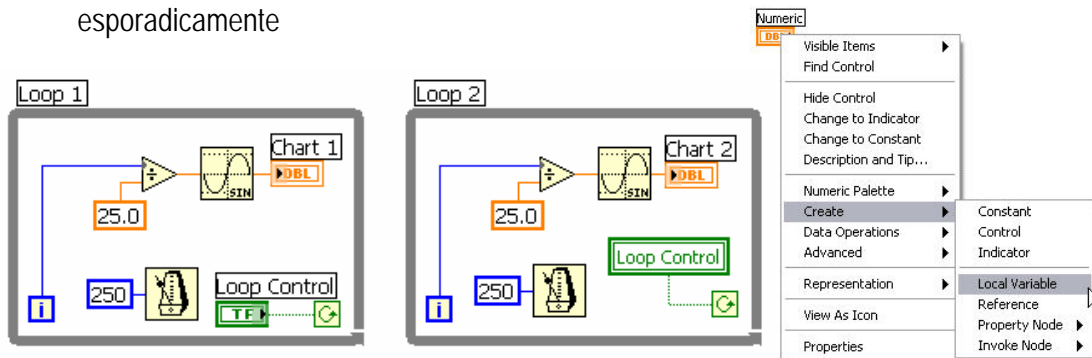
ni.com

NATIONAL
INSTRUMENTS

É impossível comunicar entre loops paralelos usando fluxo de dados. Dados não podem entrar ou sair de uma estrutura enquanto esta ainda estiver em execução via dataflow. Variáveis são elementos do diagrama de blocos que permite que você acesse dados armazenados em outro lugar. Variáveis locais guardam dados em controles e indicadores do painel frontal. Variáveis permitem que você contorne o fluxo de dados normal passando dados de um lugar para outro sem conectar os lugares por um fio.

Variáveis Locais

- Variáveis Locais permitem que dados sejam passados entre loops paralelos.
- Um controle ou indicador simples pode ler ou escrever em mais de um lugar do programa
 - Variáveis locais quebram o paradigma do dataflow e devem ser usadas esporadicamente



Variáveis locais são encontradas no subpalette **Structures** do palette **Functions**.

Quando você coloca uma variável local no diagram, ele contém o nome padrão (nome próprio) do primeiro objeto que você colocou no painel frontal.

Você usa uma variável local selecionando primeiro o objeto que você quer acessar. Você pode ou clicar na variável local com sua ferramenta de operação e selecionar o objeto, ou abrir a aba de local variable e selecionar **Select Item**.

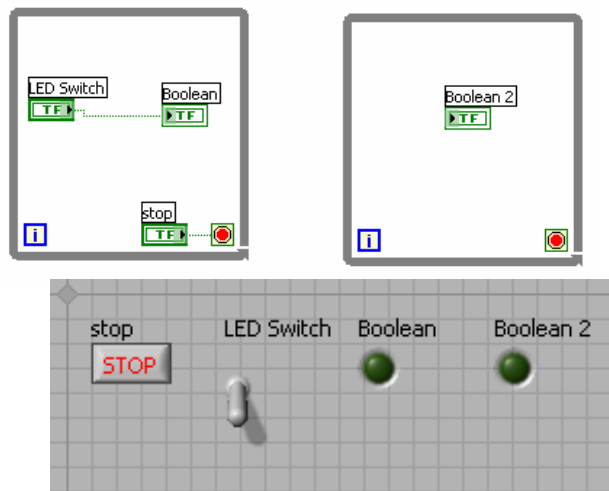
Em seguida, você deve decidir ler ou escrever o objeto. Clique com o botão direito e escolha **Change To Read** ou **Change to Write**.



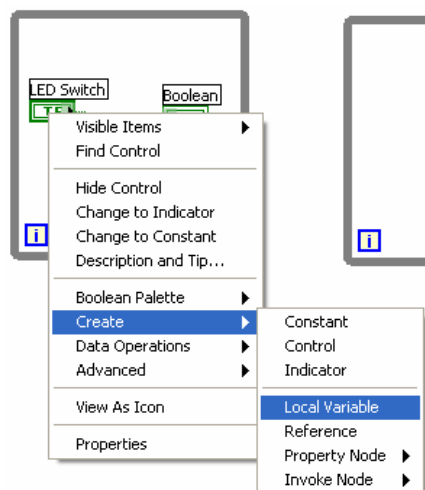
Exercício 4.2 – Criando uma variável local

Crie um VI que comunica entre dois loops paralelos usando uma variável local that communicates between two parallel while loops using a Local Variable.

1. Abra um novo VI.
2. No painel frontal, coloque um LED e dois indicadores booleanos.
3. No diagrama de blocos, coloque dois loops while e crie um botão stop clicando com o botão direito do mouse na condição de saída e selecionando **Create » Control**.
4. Arrange o código da seguinte maneira:

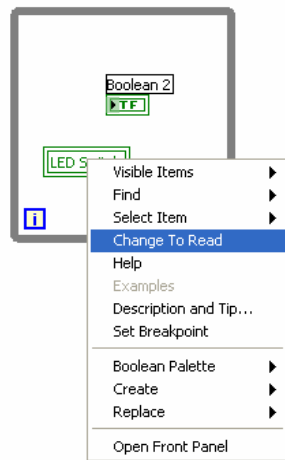


5. Clique com o botão direito do mouse no LED no diagrama de blocos e selecione **Create » Local Variable**.

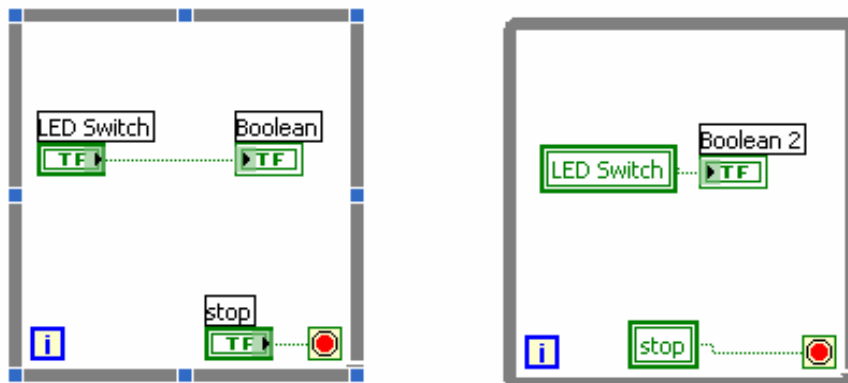


6. Coloque a variável local no segundo loop while.

7. Clique com o botão direito do mouse na variável e selecione Change To Read. Isto significa que em vez de escrever dados na variável nós iremos ler os dados já escritos nela



8. Repita o processo com o botão Stop.
9. Clique com o botão direito do mouse no botão Stop no painel frontal e mude a ação mecânica para Switch When Released. Variáveis locais não guardam dados booleanos presos. O código completo deve parecer com o seguinte:



10. Execute o VI. Veja como podemos controlar o LED e para os dois loops com um só controle.
11. Salve o VI.

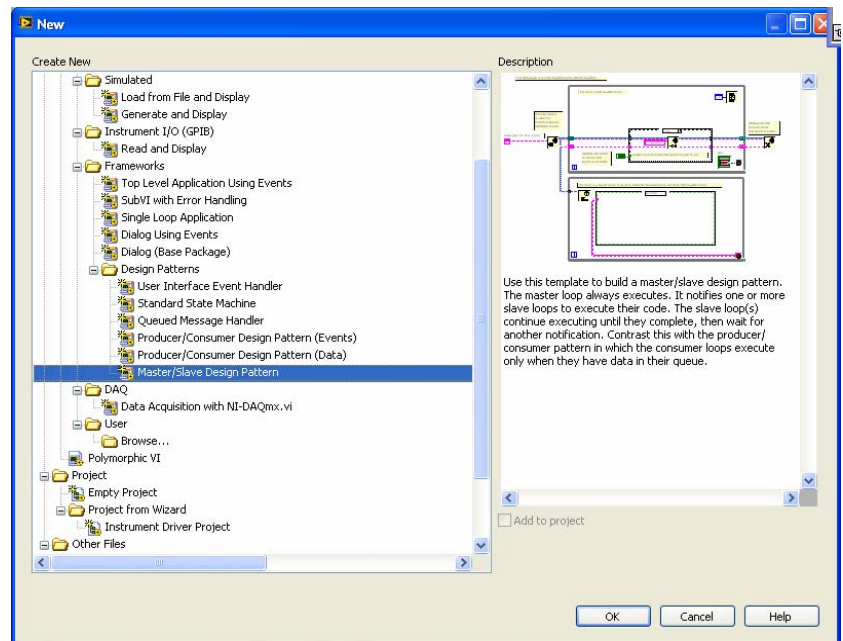
(Fim do Exercício)

Padrão de Design Produtor/Consumidor



Além de variáveis, há diversos outros métodos de transferência de dados entre loops paralelos. Isto é obtido usando as funções Notifier e Queue. Notificadores podem ser usados para implementar modelos Mestre/Escravo e Queues podem ser usados para implementar modelos Produtor/Consumidor. Ambos permitem que programadores compartilhem dados entre loops.

Selecione **File » New** e navegue para **VI » From Template » Frameworks » Design Patterns** para ver um resumo de ambos modelos.



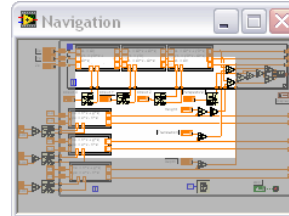
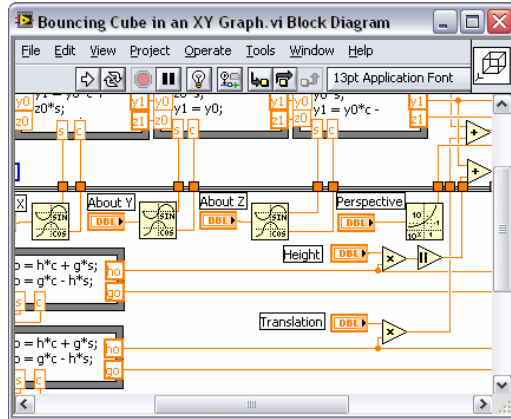
V. Desenvolvimento de Projetos Grandes

- A. Janela de Navegação
- B. Projeto LabVIEW
- C. Shared Variable

ni.com



Janela de Navegação do LabVIEW



- Mostra região de vista comparada com o Painel Frontal ou Diagrama de Blocos inteiro
- Ótimo para grandes programas

* Organize e reduza o tamanho visual de um programa com subVIs

ni.com

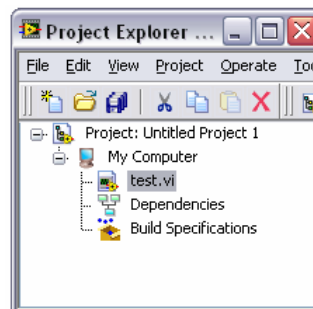
NATIONAL
INSTRUMENTS

Selecione **View»Show Navigation Window** para ver a janela.

Use a janela para navegar por um grande painel frontal ou diagrama de blocos. Clique em uma área da imagem na **Navigation Window** para mostra esta área do painel frontal ou diagrama de blocos. Você pode clicar e arrastar a imagem da **Navigation Window** para se deslocar pelo painel frontal ou diagrama de blocos.

LabVIEW Project

- Agrupe e organize VIs
- Gerenciamento de Hardware e E/S
- Gerencie VIs para alvos múltiplos
- Crie bibliotecas e executáveis
- Gerencie grandes aplicações
- Habilite rastreamento e gerenciamento de versões



(LabVIEW»Project»New)

ni.com

NATIONAL
INSTRUMENTS

Projecto LabVIEW

Use projectos para agrupar arquivos (tanto do LabVIEW quanto de outros programas), criar especificações de construção, e lançar ou baixar arquivos para alvos. Alvos são dispositivos ou máquinas onde uma VI é executada. Quando você salva um projeto, o LabVIEW cria um arquivo de projeto (.lvproj), que inclui informações de configurações, construção, e lançamento, referências a arquivos, e muito mais.

Você deve usar um projeto para construir uma aplicação autônoma e bibliotecas compartilhadas. Você deve usar um projeto, também, para trabalhar com RT, FPGA, ou PDA. Refira a módulos específicos para mais informações sobre como usar projetos com os módulos de Real-Time, FPGA, e PDA.

Drivers de Instrumentos Plug and Play para o LabVIEW usam as características de projetos e bibliotecas projetos no LabVIEW 8.0. Você pode usar drivers com estilo de projeto da mesma forma que os demais drivers Plug and Play do LabVIEW.

Project Explorer Window

Use a janela do Project Explorer para criar e editar projetos. Selecione **File»New Project** para ver esta janela. Você pode selecionar, também, **Project»New Project** ou selecionar **File»New** e aí selecionar **Empty Project** na nova caixa de diálogos para ver a janela do Project Explorer.

Shared Variables

- Shared Variables são usadas para mandar dados entre VIs.
- Tipos:
 - Single Process: compartilha dados entre VIs e computador local.
 - Network-published: comunica entre VIs, computadores remotos, e hardware através da Shared Variable Engine.
- Shared Variable deve existir dentro da biblioteca do projeto.
- Shared Variable deve ser lançada para estar disponível em outros projetos e computadores remotos.



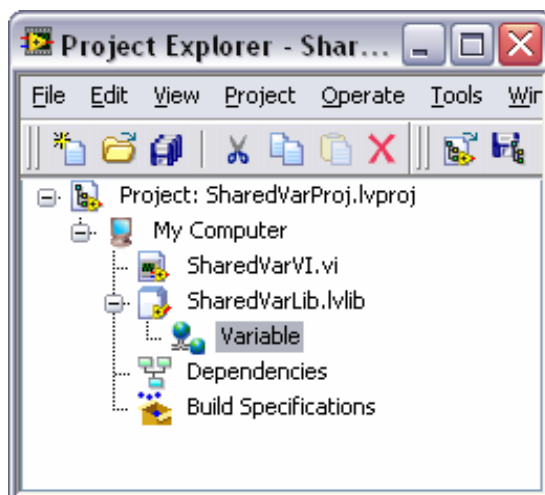
ni.com

NATIONAL
INSTRUMENTS

Shared variables são usadas para compartilhar dados entre VIs ou entre lugares em uma aplicação que não podem ser conectadas com fios. Existem dois tipos de variáveis:

- Single Process: crie shared variables que você quer ler e escrever em um único computador.
- Network-published: crie shared variables que você quer ler e escrever em computadores remotos da mesma rede.

Estes shared variables devem estar dentro da biblioteca de projetos. Se você criar uma shared variable de um alvo ou pasta que não estiver dentro da biblioteca, o LabVIEW cria uma nova biblioteca e coloca a shared variable dentro da mesma. Você deve lançar a shared variable para que esta esteja disponível em outros projetos e computadores remotos. Você consegue fazer isto executando o VI que contém a shared variable. Você pode também clicar com o botão direito do mouse na biblioteca do projeto e selecionar **Deploy**

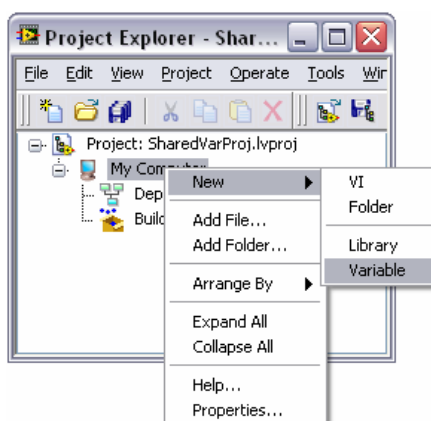




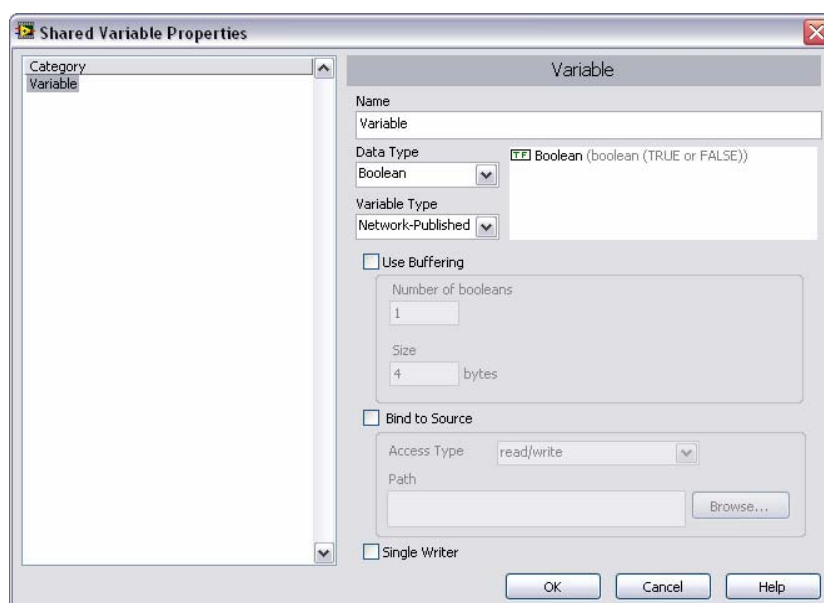
Exercício 5.1 – Shared Variable

Crie uma Shared Variable a partir de um projeto e use esta variável em vez da variável local do exercício anterior.

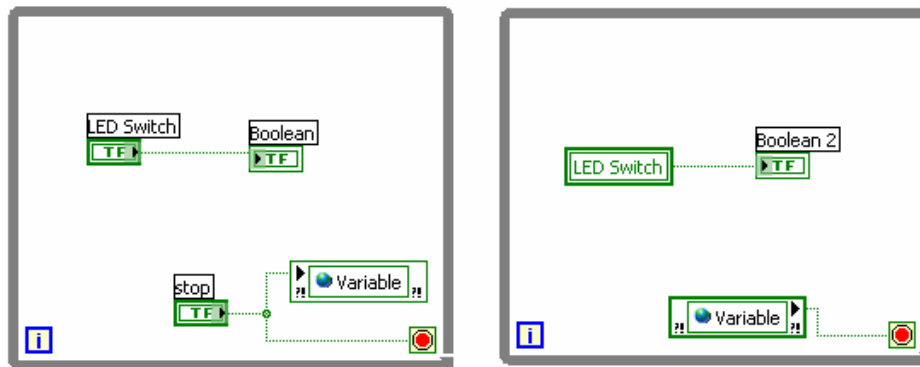
1. Abra o VI de variável local criado no Exercício 4.2.
2. Selecione **Project » New Project** no Menu Bar. Isto irá criar um novo projeto Quando pedido, selecione **Add** para acrescentar o VI aberto ao projeto.
3. Salve o projeto selecionando **Project » Save Project** na janela Project Explorer.
4. Crie uma Shared Variable clicando com o botão direito do mouse em My Computer e selecionando **New » Variable**.



5. Na janela de configuração, nomeie a variável e selecione “Boolean” no Data Type. Deixe as demais opções com seus valores default e clique em **OK**.



6. Como Shared Variables precisam estar dentro de uma biblioteca, o LabVIEW cria uma. Salve esta biblioteca clicando com o botão direito do mouse e selecionando **Save**.
7. Shared Variables podem ser usadas facilmente clicando e arrastando elas do Project Explorer ao VI. Clique e arraste a Shared Variable que você criou para o diagrama de blocos do Local Variable VI.
8. Delete a variável local que controla o botão stop no segundo loop.
9. Coloque a variável no segundo loop e ligue-a ao terminal de saída.
10. Coloque outra Shared Variable no primeiro loop. Esta Shared Variable irá escrever a informação que é lida no segundo loop.
11. Mude a Shared Variable para escrever clicando com o botão direito do mouse e selecionando **Change To Write**, e ligue a shared variable de tal forma que o botão stop seja escrito dentro da. O código completo deve parecer com o seguinte:



12. Renomei o VI selecionando **File » Save As...** e **Rename**.
13. Execute o VI. Veja que quando clicamos no botão Stop ambos os loops e o VI param.

(Fim do Exercício)

Seção VI – Controle de Instrumentos

- A. Resumo de Controle de Instrumentos
- B. GPIB
- C. Serial
- D. Assistente de Instrumento E/S
- E. VISA
- F. Drivers de Instrumentos e IDNET

ni.com



Que tipo de Instrumentos podem ser controlados?

- GPIB
- Serial
- Instrumentos Modulares
- Instrumentos Modulares PXI
- Aquisição de Imagem
- Controle de Movimento
- USB
- Ethernet
- Portas Paralelas
- CAN

ni.com



Quando configuramos um sistema de teste é frequentemente necessário, misturar e combinar instrumentos de várias categorias. Há diversos tipos de instrumentos, incluindo instrumentos GPIB, seriais, modulares, modulares PXI, de aquisição de imagem, de controle de movimento, USB, Ethernet, portas paralelas e CAN. Quando você usa um PC para comunicar com qualquer tipo de instrumento você deve estar familiarizado com as propriedades daquele instrumento, como por exemplo o protocolo de comunicação.

GPIB

- General Purpose Interface Bus (GPIB)
- GPIB é usado em instrumentos de bancada autônomos para controlar medidas e comunicar dados
- Interface de comunicação digital de 8-bits paralelos
- IEEE 488.1 e 488.2 definem padrões para GPIB

ni.com

NATIONAL
INSTRUMENTS

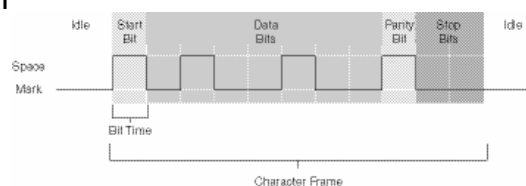
GPIB, ou General Purpose Interface Bus, é definido pelos padrões ANSI/IEEE 488.1-1987 e 488.2-1992 e descreve uma interface padrão para comunicação entre instrumentos e controladores de diversos fabricantes. Normalmente usado com instrumentos autônomos de bancada para controlar medidas e comunicar dados. Comunicação GPIB é uma interface digital paralela de 8 bits com handshaking de 3 fios e pode alcançar taxas de transferência de 1 Mbyte/s ou mais.

Refira ao site de suporte da National Instruments para GPIB www.ni.com/support/gpibsupp.htm para mais informações sobre GPIB.



Serial

- Comunicação serial transmite um bit por vez através de uma linha de transmissão
- Normalmente não requer hardware externo
- Quatro parâmetros: taxa baud, bits de dados, bit de paridade, bits de parada



ni.com

NATIONAL
INSTRUMENTS

Comunicação serial transmite dados entre um computador e um dispositivo periférico. O protocolo de comunicação serial usa transmissores para enviar dados um bit por vez em uma única linha de comunicação para um receptor. Este método é o melhor para taxas baixas de transferência, ou quando precisamos transmitir dados a grandes distâncias. Como a maioria dos computadores tem pelo menos uma porta serial, hardware adicional não é necessário.

Quatro parâmetros devem ser especificados para comunicação serial, taxa baud, bits de dados, bit de paridade, e os bits de parada. Uma estrutura de caracteres transmite cada caractere como um bit inicial, seguido de um bit de dado como mostrado acima para o caractere M.

Há diversos padrões para portas seriais, estes, porém, são os mais comuns:


RS-232 (ANSI/EIA-232 Standard) (mais popular)

RS-422 (AIA RS-422A Standard)

RS-485 (EIA-485 Standard)

Instrument I/O Assistant

- LabVIEW Express VI usado para comunicar com instrumentos baseados em mensagens
- Comunica com um instrumento que usa uma interface serial, de Ethernet, ou GPIB.
- Use o Instrument I/O Assistant quando o driver do instrumento não está disponível



ni.com

NATIONAL INSTRUMENTS

O Instrument I/O Assistant é um VI Expresso do LabVIEW Express que pode ser usado para comunicar com instrumentos baseados em mensagens e converter respostas de dados crus para representação ASCII. Você pode comunicar com um instrumento que usa uma interface serial, Ethernet, ou GPIB e deve ser usado quando o driver do instrumento não está disponível.

O Instrument I/O Assistant organiza a comunicação entre instrumentos em passos organizados. Para usar o Instrument I/O Assistant, coloque os passos em sequência. Na medida que acrescentamos passos, ele aparecem na janela Step Sequence. O LabVIEW acrescenta terminais de entrada e saída ao Instrument I/O Assistant Express VI no diagrama de blocos que correspondem aos dados recebidos do instrumento.

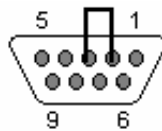


Exercício 6.1 – Teste de loop back com o Instrument I/O Assistant

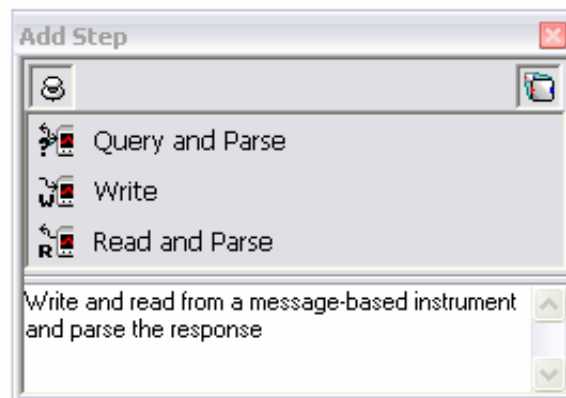
Nota: Este exercício usa a porta paralela e requer um cabo serial e um fio. A maioria dos PCs têm uma porta serial disponível. O Instrument I/O Assistant pode ser usado facilmente para comunicar com um dispositivo GPIB, mas isto requereria hardware GPIB ao invés de uma porta serial.

Complete os seguintes passos para configurar o I/O Assistant para realizar um teste de loopback usando a porta serial.

1. Conecte o cabo serial à porta COM do computador.
2. Conecte a transmissão e recepção de linhas do cabo serial nos pinos 2 e 3 como mostrado abaixo.

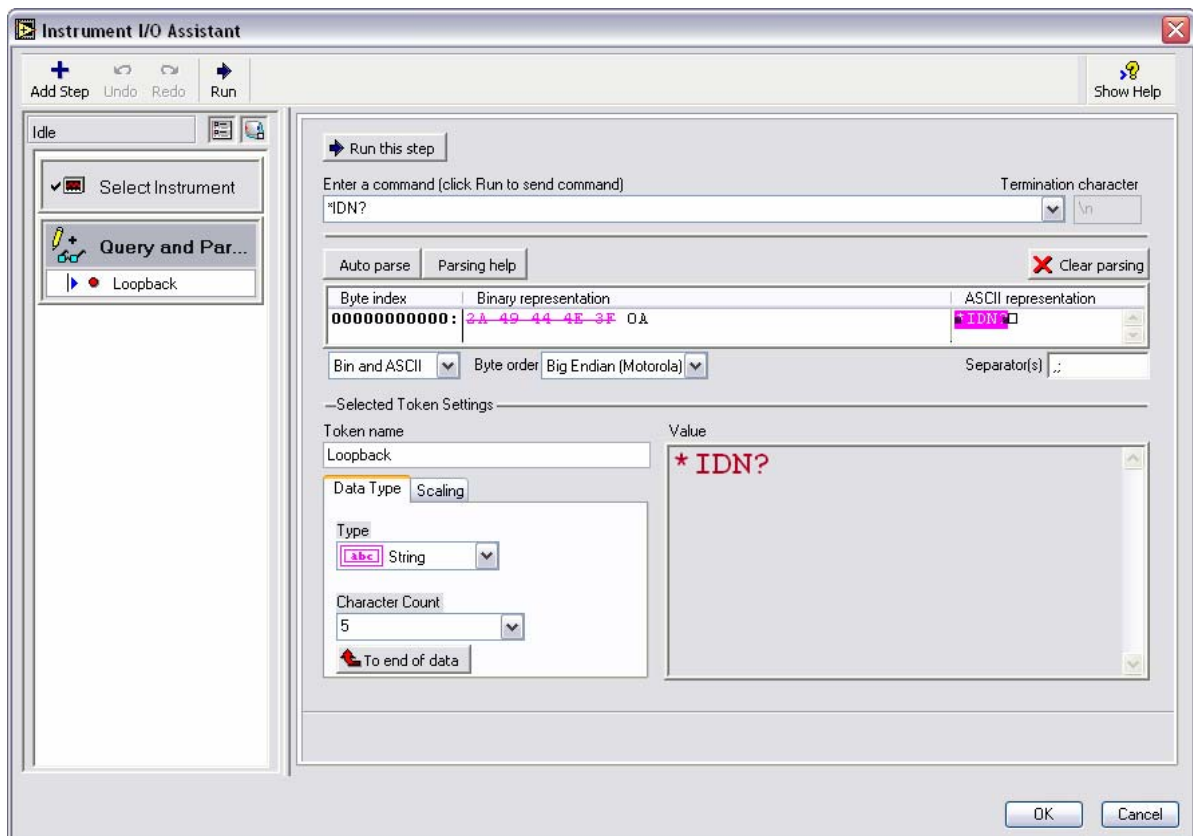


3. Abra um VI em branco da tela Getting Started.
4. Coloque o Instrument I/O Assistant no diagrama de blocos. Clique com o botão direito do mouse para abrir o paleta de funções e selecione **Instrument I/O » Instrument I/O Assistant**.
5. Abra a caixa de configuração Instrument I/O Assistant clicando duas vezes no ícone caso a caixa não apareça.
6. Selecione COM1 da caixa Select an Instrument.
7. Clique em Add Step para criar um novo passo e selecione Query and Parse.



8. Configure o passo Query and Parse:
 - a. Digite *IDN? No campo **Enter a Command**.
 - b. Nomeie a saída do Instrument I/O Assistant entrando com Loopback no campo **Token name**.
 - c. Clique no botão **“Run this Step”** para executar o teste de loopback.
 - d. Clique no botão **“Auto Parse”** para converter os dados crus para ASCII.

9. Clique no botão OK para sair da janela de configuração e gerar o código.
10. Clique com o botão direito do mouse na saída do string do Instrument I/O Assistant e crie um indicador.
11. Volte ao painel frontal e execute o VI. Observe o texto no indicador. Como estamos realizando um teste de loopback o texto do indicador deve ser o mesmo do comando que foi digitado no Instrument I/O Assistant. *IDN? é um comando padrão que retorna a informação do dispositivo, mas qualquer texto pode ser usado para um teste de loopback.
12. Salve e feche o VI.



(Fim do Exercício)

VISA

- Virtual Instrumentation Software Architecture (VISA)
- API de alto nível que chama drivers de baixo nível
- Pode controlar VXI, GPIB, serial, ou instrumentos baseados em computador
- Faz chamadas apropriadas aos drivers dependendo do instrumento usado.

ni.com



Virtual Instrument Software Architecture (VISA) é a base para os drivers de instrumentos do LabVIEW. VISA não traz diretamente a capacidade de programação de instrumentos, mas serve como um API de alto nível que chama drivers de baixo nível. VISA pode controlar VXI, GPIB, serial ou outros instrumentos baseados em computador e faz as chamadas corretas aos drivers dependendo do tipo de instrumento.

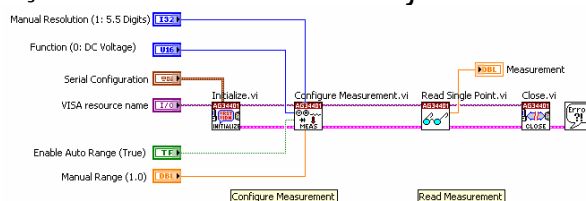
No LabVIEW, VISA é uma única biblioteca de funções que se adapta a diferentes instrumentos, para que não seja necessário usar paletes separados de E/S. A seguinte terminologia é usada para programação com VISA:

- Resource – Qualquer instrumento no sistema incluindo portas seriais e paralelas
- Session – Canais de comunicação que são usados pelo VISA para identificar uma referencia específica para aquele instrumento.
- Instrument Descriptor – Nome exato do instrumento (veja lista abaixo)

Interface	Syntax
Asynchronous serial	ASRL [board] [: : INSTR]
GPIB	GPIB [board] : : primary address [: : secondary address] [: : INSTR]
VXI instrument through embedded or MXIbus controller	VXI [board] : : VXI logical address [: : INSTR]
GPIB-VXI controller	GPIB-VXI [board] [: : GPIB-VXI primary address] : : VXI logical address [: : INSTR]

Instrument Drivers

- Drivers Plug and Play são um conjunto de VIs que controlam um instrumento programável
- VIs correspondem a operações do instrumento: configuração, triggering, e leitura de medidas
- Ajudam a começar uma vez que o protocolo de programação de cada instrumento já é conhecido



ni.com

NATIONAL
INSTRUMENTS

Um driver de instrumento Plug and Play do LabVIEW é um conjunto de VIs que controlam um instrumento programável. Cada VI no driver corresponde a uma operação específica do instrumento, como configuração, triggering e leitura de medidas. Isto reduz drasticamente o tempo de desenvolvimento permitindo que você use o instrumento no LabVIEW sem conhecimento profundo de seu protocolo de comunicação.

Abaixo temos um exemplo de um driver de instrumento para o Agilent 34401 digital multimeter (DMM) que inicializa, configura, le e mede, fecha a sessão de medidas com o instrumento e verifica se houveram erros.

IDNET

- Instrument Driver Network (IDNET)
- Instrument Driver Finder dentro do LabVIEW

Tools » Instrumentation » Find Instrument Drivers

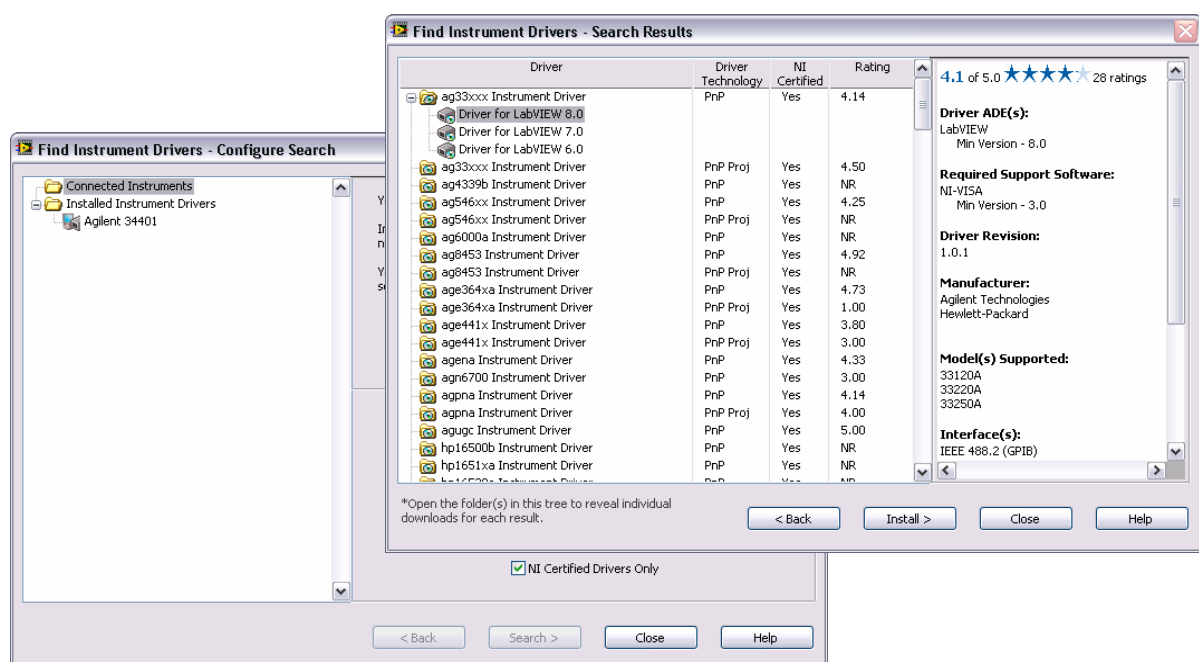
Help » Find Instrument Drivers

- Pode ser encontrado online em www.ni.com/idnet

ni.com



A maioria dos drivers de instrumentos Plug and Play do LabVIEW são encontrados no Instrument Driver Finder dentro do LabVIEW, que pode ser acessado clicando em Tools » Instrumentation » Find Instrument Drivers or Help » Find Instrument Drivers. O Instrument Driver Finder se conecta a www.ni.com e encontra o driver do instrumento. O Finder permite que você veja instrumentos conectados e drivers instalados, assim como fazer buscas de drivers por fabricante e palavras chaves.



Recursos Adicionais

- NI Academic Web & Student Corner

- <http://www.ni.com/academic>

- Conexões: Full LabVIEW Training Course

- www.cnx.rice.edu

- Or search for “[LabVIEW basics](#)”



- LabVIEW Certification

- LabVIEW Fundamentals Exam (free on www.ni.com/academic)

- Certified LabVIEW Associate Developer Exam (industry recognized certification)

- Get your own copy of LabVIEW Student Edition

- www.ni.com/academic



By [Robert H Bishop](#).

Published by [Prentice Hall](#).

ni.com

NATIONAL INSTRUMENTS

[view cart](#) | [help](#) | [search](#) [GO](#)

[MyNI](#) | [Contact NI](#) | [Products & Services](#) | [Solutions](#) | [Support](#) | [NI Developer Zone](#) | **Academic** | [Events](#) | [Company](#)

You are here: [NI Home](#) > [Academic Programs](#) English

Academic Programs and Resources

Questions? [Call \(800\) 531-5066](#)

[Academic Community](#)
[Grants & Donations](#)
[Distance Learning](#)
[LabVIEW Training](#)
[Curriculum Resources](#)
[Discipline Depot](#)
[Hardware](#)
[Software](#)

NI LabVIEW Hands-On Campus Workshop
Learn More >>

National Instruments provides products and resources to enhance engineering and science education. Explore [hardware](#), [software](#), and curriculum resources, including [Lab Share](#), [Courseware](#), [Journals](#), and [Experiments](#).

Main Academic Resources

Educators

- [Lab Share](#)
- [Courseware](#)
- [Experiments](#)
- [Journals](#)
- [LabVIEW Training](#)
- [Academic Discounts](#)
- [Subscribe to NI News - Academic Edition](#)

Researchers

- [Discounts](#)
- [Products](#)
- [Training](#)
- [Tutorials](#)
- [Technical Support](#)
- [User Groups](#)

Students

- [LabVIEW Zone Student Corner](#)
- [LabVIEW Fundamentals Exam](#)
- [LabVIEW Student Edition](#)
- [Technical Support](#)
- [Jobs at NI](#)

Technical Resources

- [Example Code](#)
- [Application Notes](#)
- [Product Manuals](#)
- [Development Library](#)
- [White Papers](#)
- [Academic Resources CD](#)
- [Technical Support](#)

Spotlights

Attend NIWeek 2005
Learn more about registration discounts and academic-specific workshops.
[GO >>](#)

Are You On A LabVIEW Campus?
Find out if your campus holds a LabVIEW site license.
[Learn More >>](#)

Services and Training

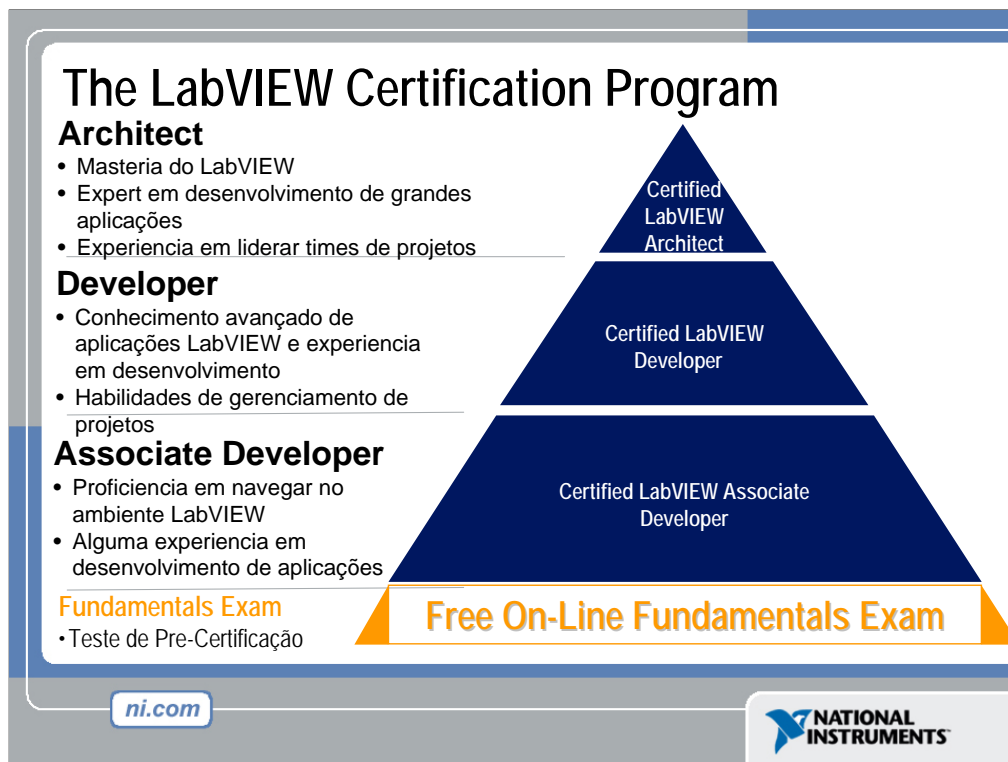
- [Training](#)
- [Tutorials](#)
- [LabVIEW Interactive Training CD](#)

Highlights

Lab Share
A compilation of lab manuals from schools successfully integrating NI technology

Courseware
A comprehensive set of instructor notes, equipment lists, and example code for a variety of topics and courses

Community



Hoje, cada vez mais, companhias estão gerentes de contratação estão pedindo experiencia com LabVIEW em suas entrevistas. O programa de certificação foi construido em cima de uma série de exames proficionais. Certificação LabVIEW é usado para validar experiencia e habilidades com LabVIEW para oportunidades de emprego e lances de projetos.

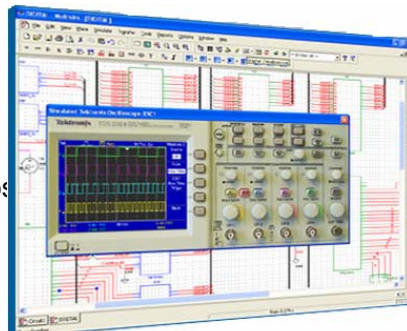
O Certificado LabVIEW Associate Developer é o primeiro passo de certificação em LabVIEW e demonstra uma fundação forte para usar o LabVIEW e seu ambiente. Como alunos, seu certificado de Associate Developer diferencia suas habilidades com o LabVIEW para oportunidades de emprego e também traz reconhecimento de sua experiencia com o LabVIEW. O CLAD é um exame de multipla escolha que dura uma hora e é conduzido nos centros de teste Pearson VUE em diversos lugares do pais. O exame cobre vários tópicos do ambiente LabVIEW incluindo conceitos de dataflow, estruturas de programação, Tecnicas avançadas de E/S de arquivos, Pratica de programação Modular, Propriedades de VIs e referências de controle.

Pensando conseguir seu certificado CLAD? Faça o exame LabVIEW Fundamentals Exam gratuitamente online.

Developer e Architect certificado em LabVIEW são certificados proficionais que validam um conhecimento avançado de LabVIEW e experiencia em desenvolvimento de aplicações. Adicionalmente, o certificado de arquiteto demonstra habilidades em liderar times de projetos e experiencia desenvolvendo grandes aplicações. Estes exames, são provas práticas que duram quatro horas e são conduzidos pela National Instruments.

Electronics Workbench e Multisim

- Software mais popular do mundo para a aprendizagem de eletrônica
- 180,000 usuários industriais e acadêmicos
- Produtos incluem:
 - Multisim: Simulation and Capture
 - Multi-MCU: Microcontroller Simulation
 - MultiVHDL: VHDL Simulation
 - Ultiboard: PCB Layout
 - Electronics CBT: Computer-based training
- Edição de baixo custo para estudantes disponível
- www.electronicsworkbench.com



ni.com

NATIONAL
INSTRUMENTS

Os produtos da Electronics Workbench são os softwares mais usados mundialmente no ensino de engenharia elétrica e eletrônica. Como a única companhia a desenvolver produtos especificamente para o mercado educacional, nossos software têm se tornado a ferramenta principal de aprendizagem e ensino para milhares de educadores.

MULTISIM — SIMULAÇÃO E CAPTURE

Multisim é um programa de simulação e captura intuitivo, com esquemática arrasta-e-solta que permite que educadores e alunos criem circuitos completos rapidamente contendo tanto componentes analógicos quanto digitais.

MULTIMCU — CO-SIMULAÇÃO DE MICROCONTROLADORES

MultiMCU acrescenta uma unidade de co-simulação de microcontroladores ao Multisim, permitindo que você inclua um MCU programado e código de assembly, dentro de seu circuito modelado com SPICE (e VHDL opcional) modeled circuit.

MULTIVHDL — CO-SIMULAÇÃO DE VHDL

MultiVHDL acrescenta capacidades patenteadas de co-simulação de VHDL co-simulação ao Multisim. Esta aplicação potente e fácil de usar é perfeita para o ensino de programação de HDL, ou para incluir dispositivos programados em VHDL no seu projeto Multisim.

ULTIBOARD — PCB LAYOUT

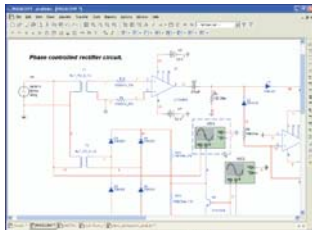
Ultiboard permite que alunos ganhem exposição às implementações físicas e manufatureis de circuitos em PCBs. A esquemática do Multisim pode ser importada para uma Ultiboard com um único clique do mouse.

ELECTRONICS CBT — TREINAMENTO BASEADO NO COMPUTADOR

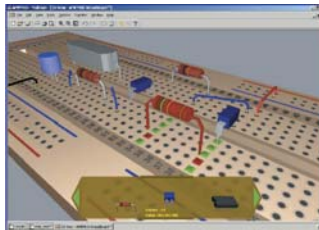
Electronics CBT oferece uma introdução completa e autônoma à eletrônica para auxiliar em suas palestras ou agir como peça central de seu curso. E-CBT é repleto com mais de 400 exercícios e experimentos que rodam diretamente do ambiente virtual do Multisim

Multisim Integrado com LabVIEW

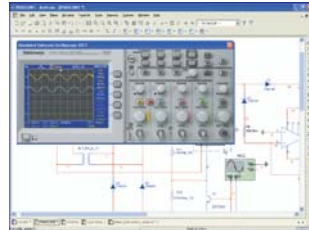
1. Crie Esquematicas



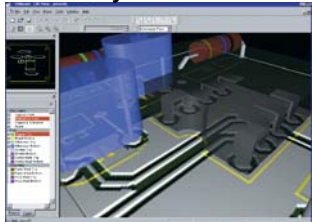
2. Breadboard Virtual



3. Simule



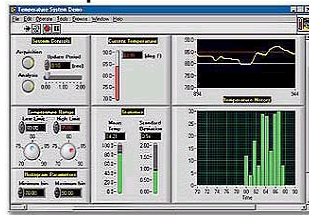
4. PCB Layout



5. Teste



6. Compare



ni.com

NATIONAL
INSTRUMENTS

1. Multisim - Schematics

- Esquemática simples de usar
- Simplesmente clique e arraste
- Partes animadas 3D
- Arraste fios sem quebrar conexões

2. Multisim –breadboard Virtual

- técnicas de Breadboarding
- Sincronizado com esquemática
- Relatório de fiação para o Elvis (step 5)

3. Multisim – Simulação

- biblioteca de 13.000 partes
- 20 instrumentos virtuais
- Mudanças a toque de caixa
- Novo simulador de microcontroladores
- Partes Animadas (LEDs, e displays de 7-segmentos)

4. Ultiboard – PCB Layout

- Integrado com Multisim
- Interface amigável
- Vista 3D
- Checagem de regras de design
- Auto-roteamento embutido

5. Elvis – Test

- Instrumentação
- Aquisição de dados
- Prototipagem

6. LabVIEW – Compare

- Importe automaticamente:
 - dados virtuais do Multisim
 - dados reais do ELVIS
- Compare dados reais e ideais

Seu próximo passo...

Faça o LabVIEW Fundamentals Exam em
ni.com/academic

Seu primeiro passo para se certificar com LabVIEW!

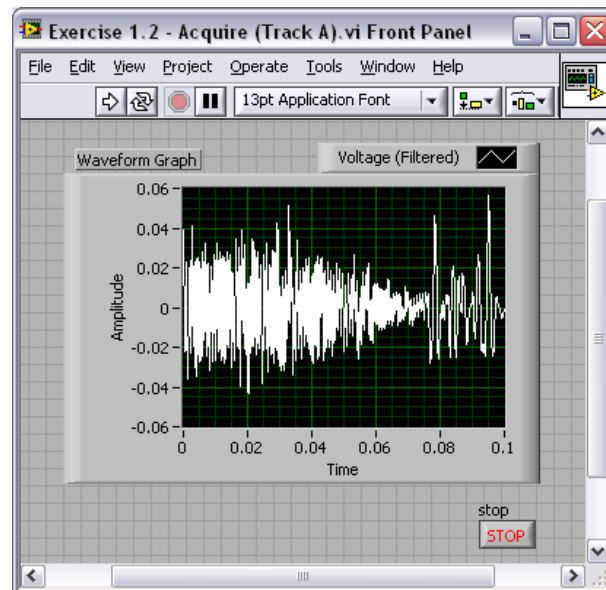


ni.com

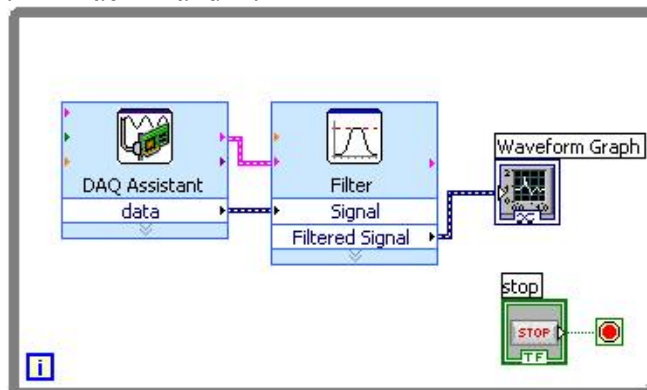


Solutions Section

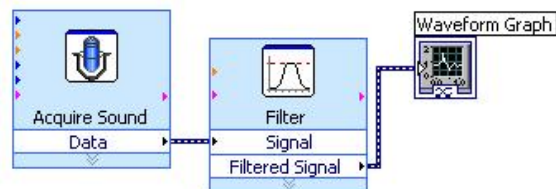
Exercício 1.2:



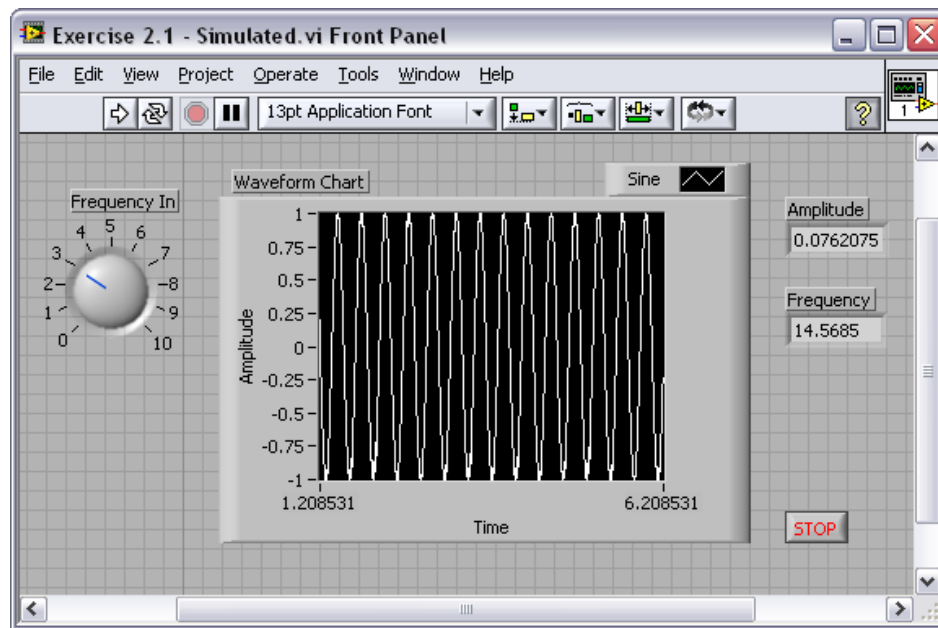
Exercício 1.2 – Track A and B:



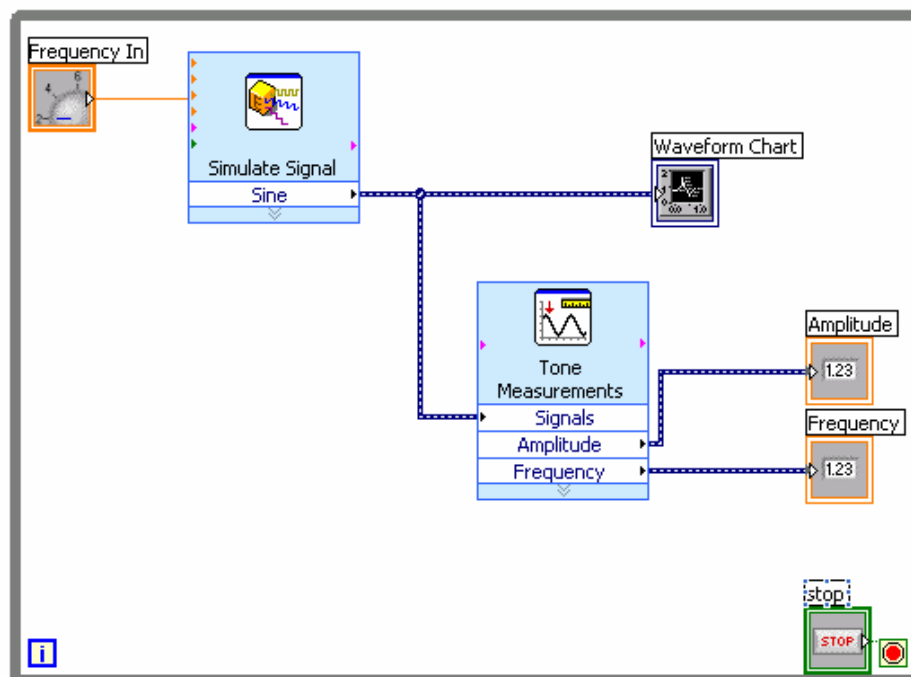
Exercício 1.2 – Track C:



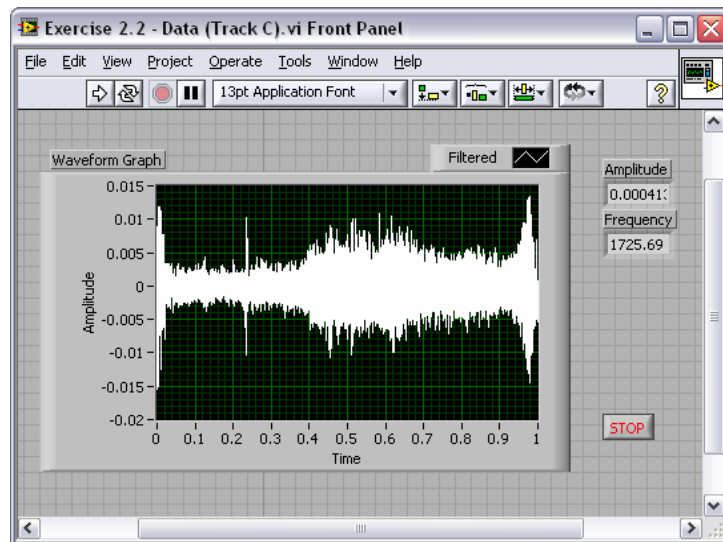
Exercício 2.1:



Exercício 2.1 – Track A, B, and C:

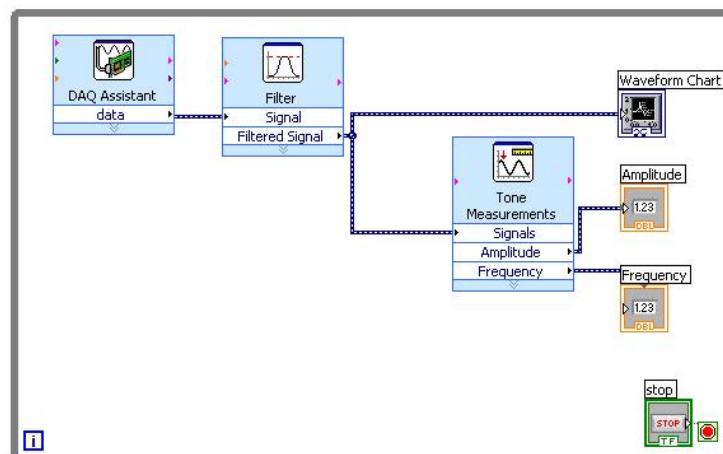


Exercício 2.2:

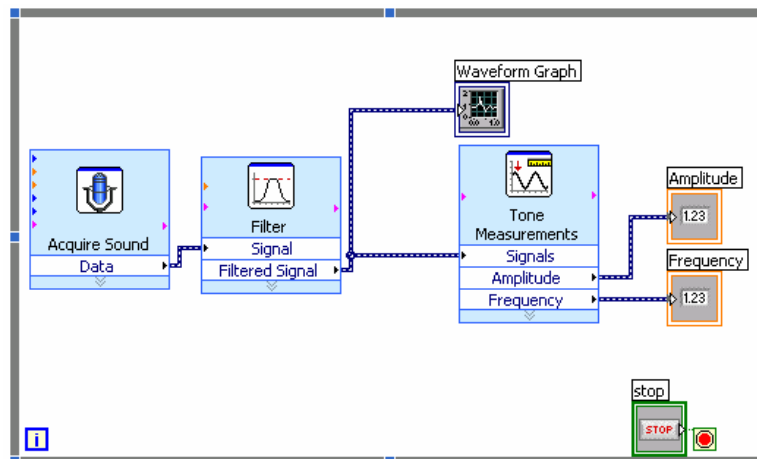


Note: Waveform Graphs and Charts have been used interchangeably in Exercício 3.2 and 3.3.

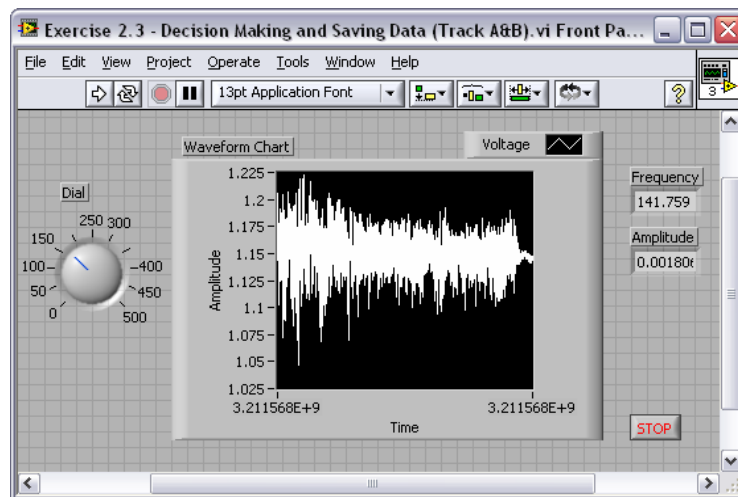
Exercício 2.2 – Track A and B:



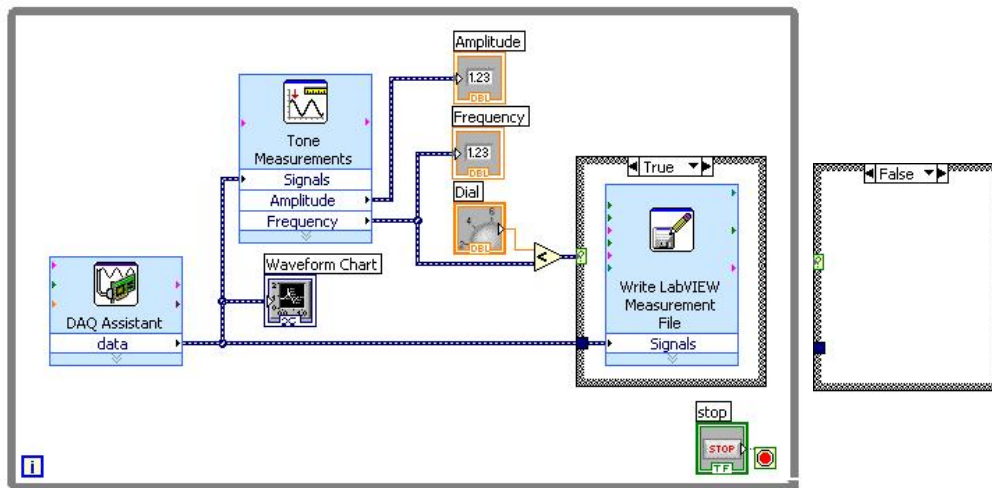
Exercício 2.2 – Track C:



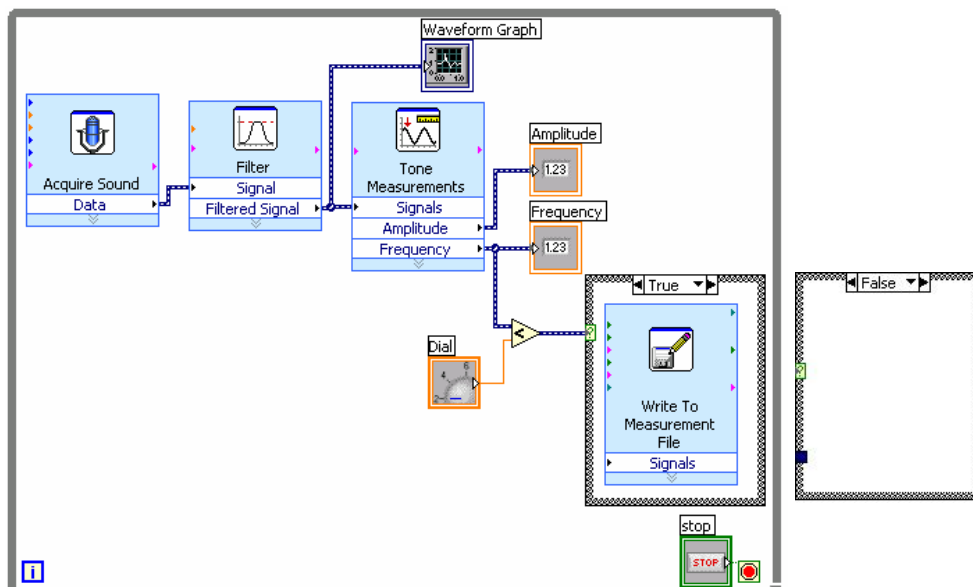
Exercício 2.3:



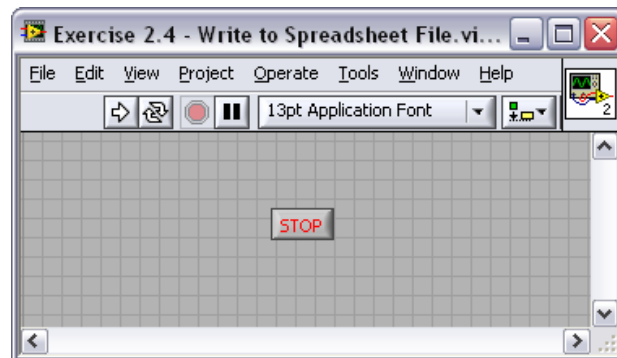
Exercício 2.3 – Track A and B:



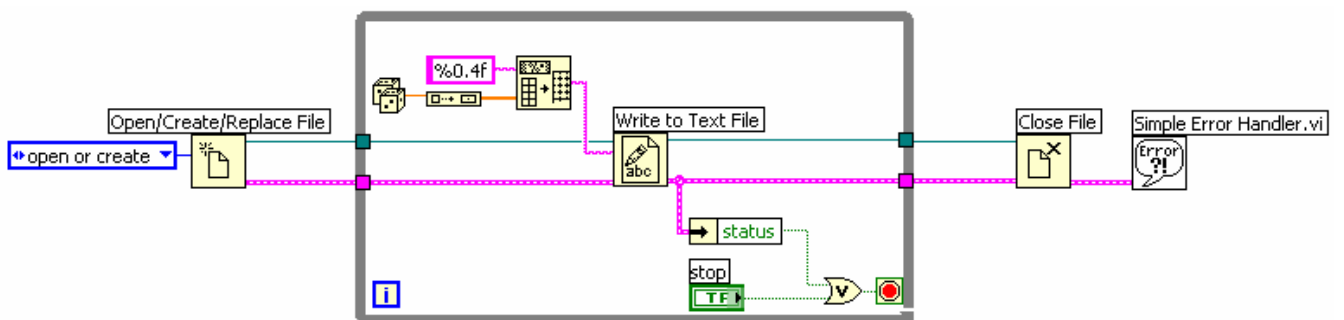
Exercício 2.3 – Track C:



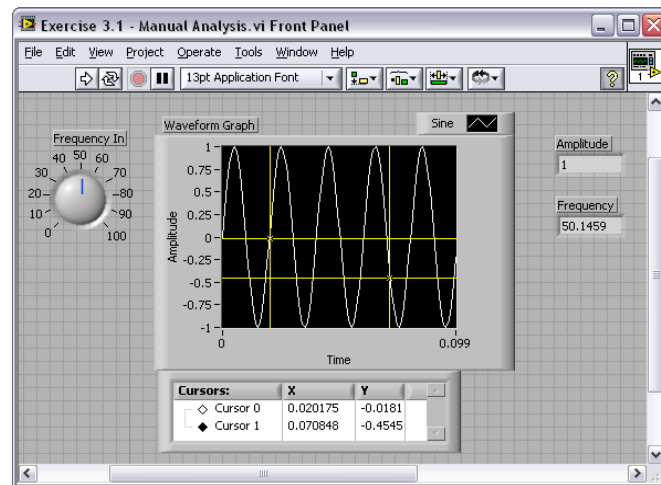
Exercício 2.4:



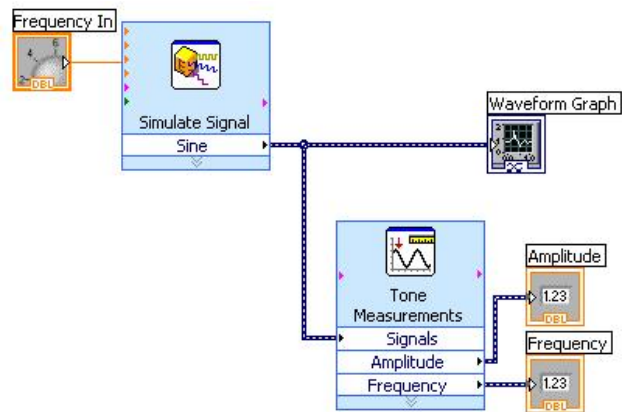
Exercício 2.4



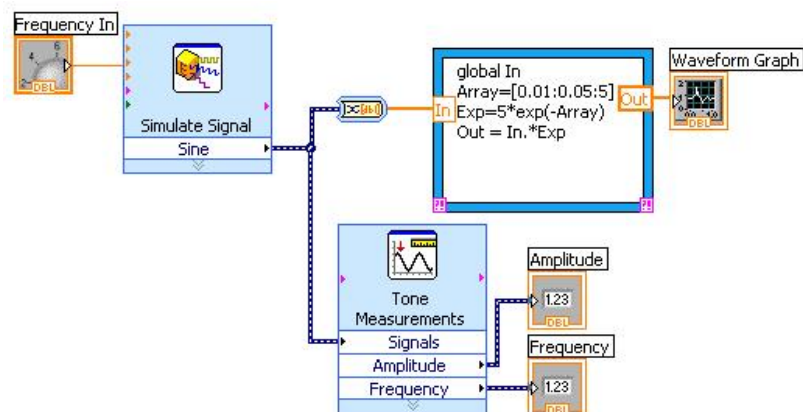
Exercício 3.1 and 3.2:



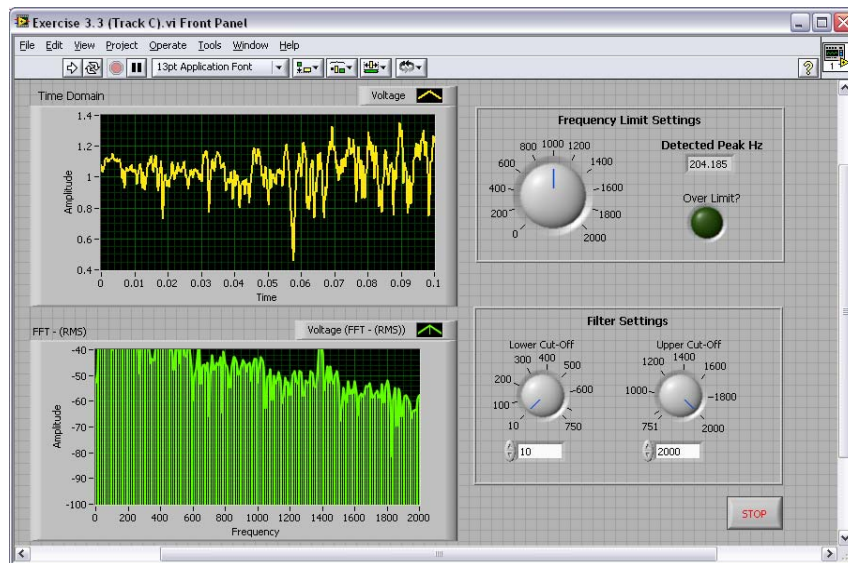
Exercício 3.1 – Track A, B, and C:



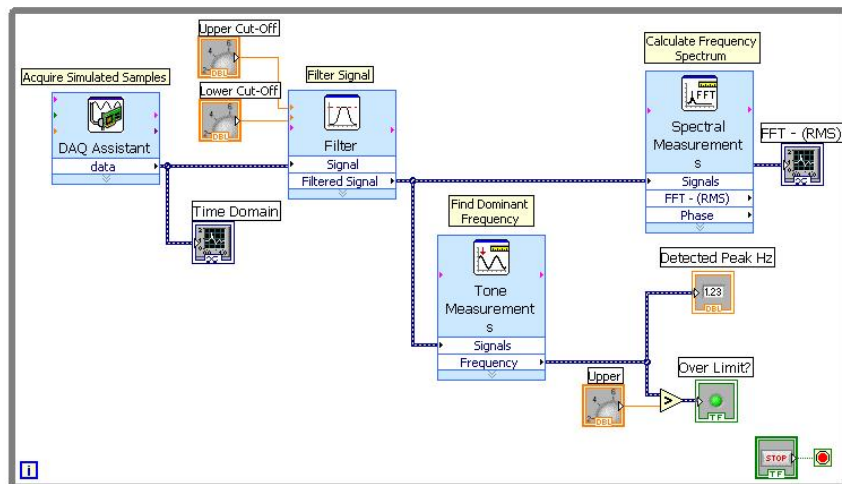
Exercício 3.2 – Track A, B, and C:



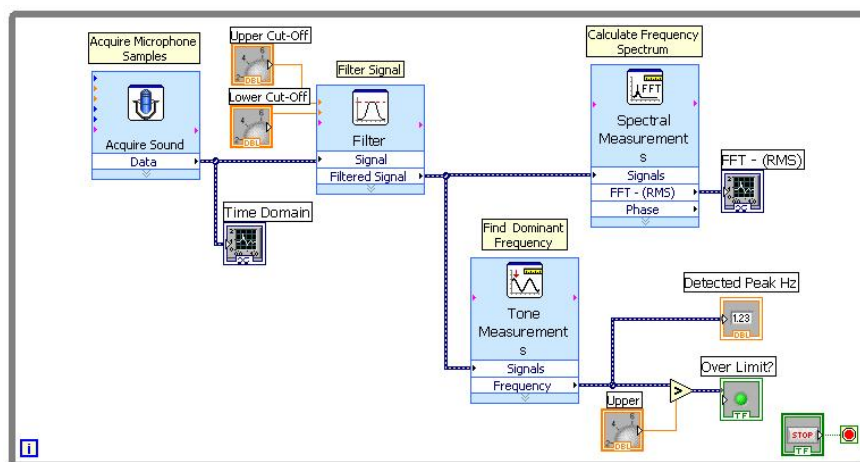
Exercício 3.3:



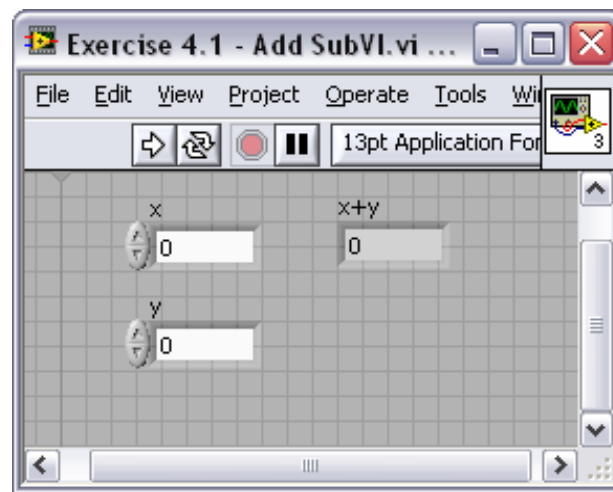
Exercício 3.3 – Track A and B:



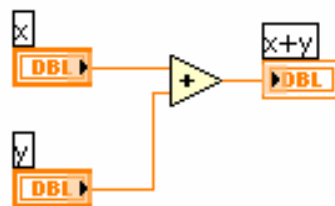
Exercício 3.3 – Track C:



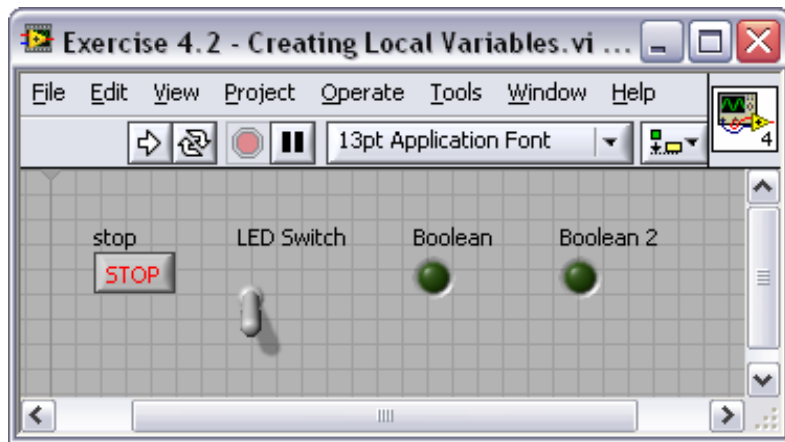
Exercício 4.1:



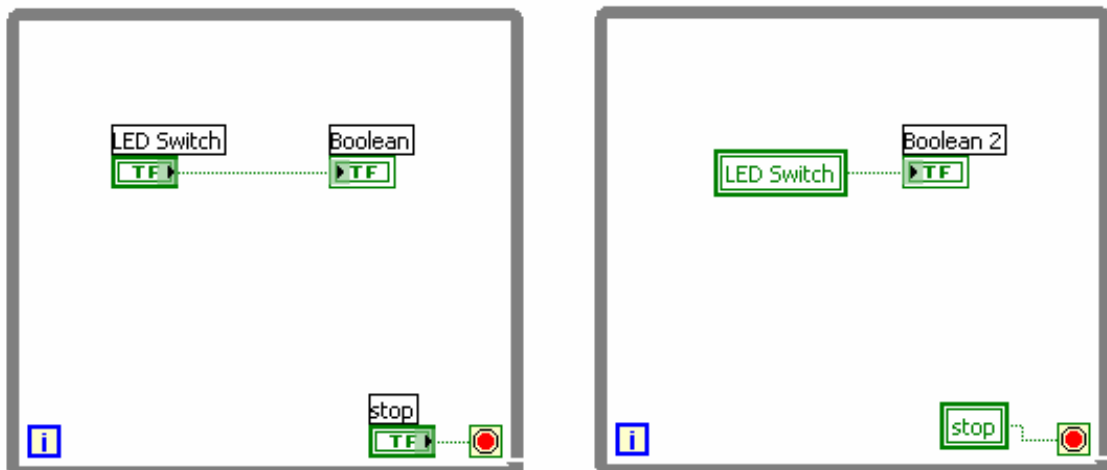
Exercício 4.1:



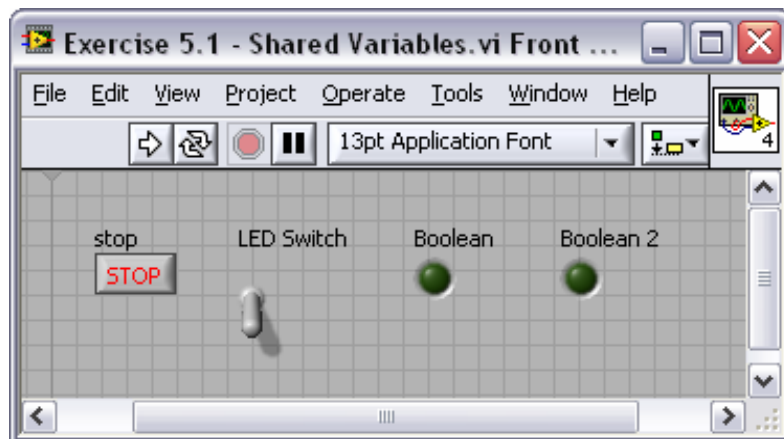
Exercício 4.2:



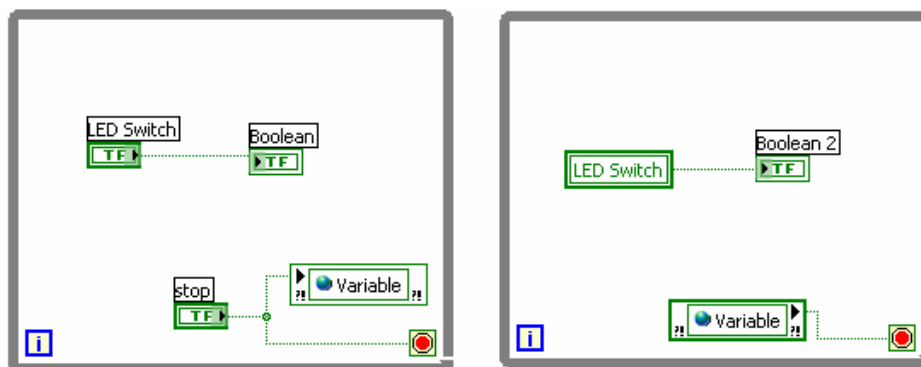
Exercício 4.2:



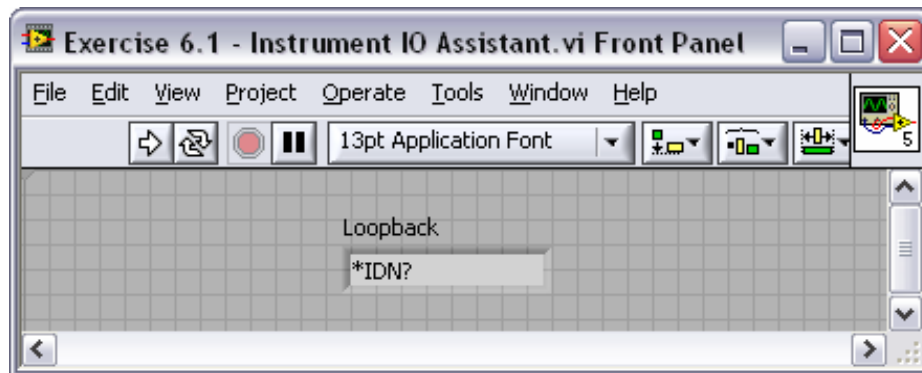
Exercício 5.1:



Exercício 5.1:



Exercício 6.1:



Exercício 6.1:

