

Introdução ao LabVIEW



Programação Gráfica
Para Engenheiros e Cientistas

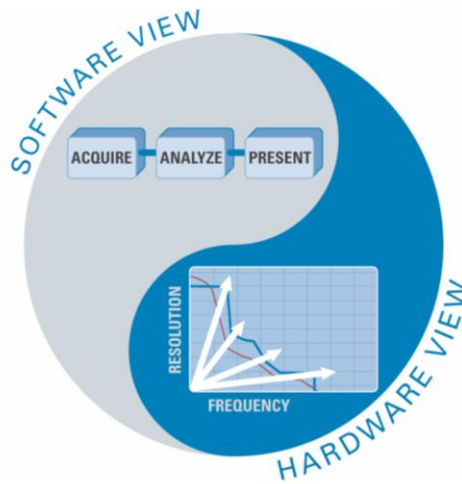
NATIONAL INSTRUMENTS
LabVIEW™ 2010

Treinamento prático de 6 horas

ni.com



A abordagem da Instrumentação Virtual



ni.com

3



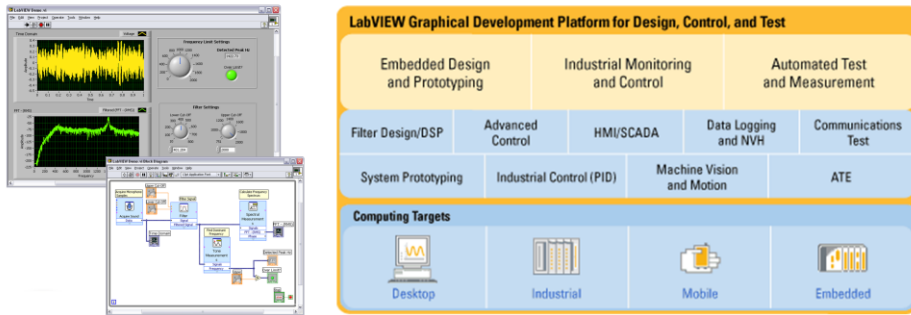
Instrumentação Virtual

Por mais de 30 anos, a National Instruments tem revolucionado o modo como engenheiros e cientistas na indústria e em órgãos governamentais e acadêmicos abordam temas como medição e automação. Utilizando os avanços em PCs e na tecnologia comercial, a instrumentação virtual aumenta a produtividade e reduz custos para aplicações de teste, controle e projeto através de ferramentais de software de fácil integração como o NI LabVIEW, e hardware de controle e medição modular para diversos barramentos, como PXI, PCI, USB e Ethernet.

Com a instrumentação virtual, engenheiros usam um software de programação gráfica para criar soluções definidas pelo usuário para atender suas necessidades específicas, o que é uma ótima alternativa para a instrumentação tradicional (proprietária e de funcionalidade única). Além disso, a instrumentação virtual utiliza o desempenho cada vez maior de computadores pessoais. Por exemplo, em aplicações de teste, medidas e controle, engenheiros têm usado a instrumentação virtual para reduzir as dimensões dos ATE (automated test equipment) enquanto experimentam um ganho na produção de até 10 vezes com um custo que representa uma fração de uma solução de instrumentação tradicional. No último ano, 25.000 companhias em 90 países investiram mais de 6 milhões em canais de instrumentação virtual da National Instruments.

LabVIEW - Ambiente gráfico de desenvolvimento

- O código pode ser compilado para múltiplos Sistemas Operacionais e dispositivos
- Eficiente em diversas aplicações



ni.com

4



O LabVIEW da National Instruments é uma ferramenta de software líder na indústria para o projeto de sistemas de teste, medida e controle. Desde sua introdução em 1986, engenheiros e cientistas ao redor do mundo que confiaram desenvolvimento gráfico do NI LabVIEW para projetos por todo o ciclo do produto ganharam melhor qualidade, menor tempo de mercado e maior eficiência na engenharia e na fabricação. Usando o ambiente integrado do LabVIEW para realizar a interface com sinais do mundo real, analisar os dados para informações importantes e compartilhar resultados, pode aumentar a produtividade para toda a organização. Como o LabVIEW tem a flexibilidade de uma linguagem de programação combinada com ferramentas integradas que foram projetadas especialmente para teste, medida e controle, pode-se criar aplicações que vão desde uma simples monitoração de temperatura até sistemas sofisticados de simulação e controle. Independente do que seja o seu projeto, o LabVIEW tem a ferramenta necessária para atingir os objetivos rapidamente.

Aplicações para a Instrumentação Virtual

- **Projeto**
 - Processamento de Sinal e Imagem
 - Programação de Sistemas Embarcados
 - (PC, DSP, FPGA, microcontroladores)
 - Simulação e Prototipagem
 - E mais...
- **Controle**
 - Controle automático e sistemas dinâmicos
 - Mecatrônica e robótica
 - E mais...
- **Medidas**
 - Circuitos e eletrônica
 - Medidas e instrumentação
 - E mais ...

Uma única plataforma de desenvolvimento gráfico



ni.com

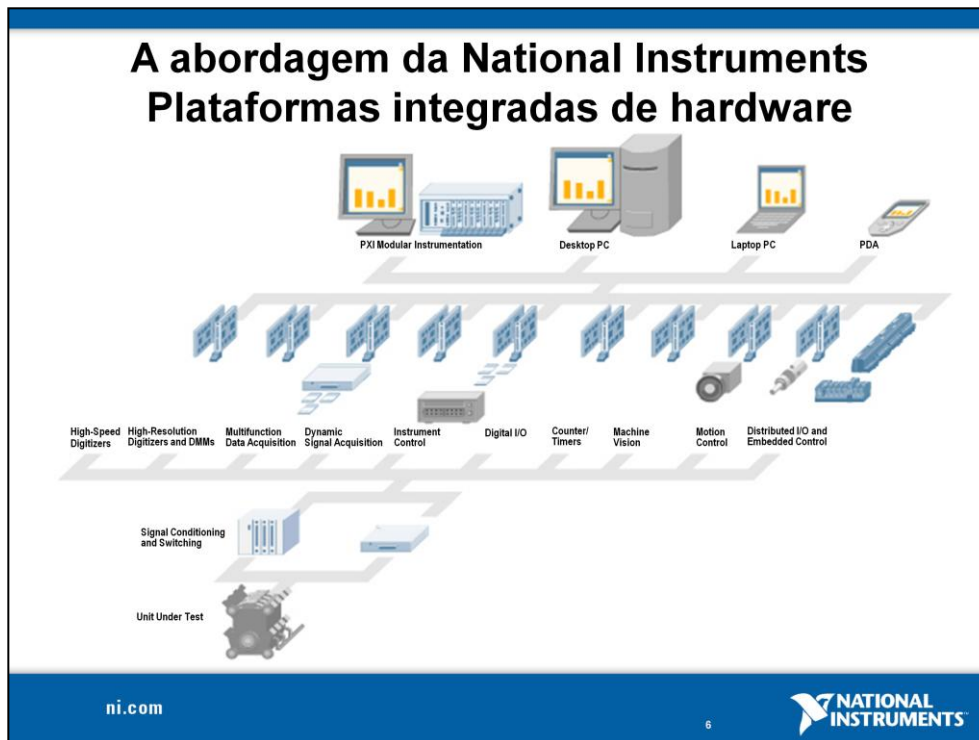
5



Aplicações para instrumentação virtual

A instrumentação virtual poder ser usada em diferentes tipos de aplicações, desde o projeto até a prototipagem e implementação. A plataforma LabVIEW fornece ferramentas e modelos específicos para atender aos desafios de sua aplicação, variando do projeto de um algoritmo de processamento de sinal até realizar medidas de tensão e pode ser executado em diferentes plataformas, de um desktop até dispositivos embarcados – com um paradigma gráfico intuitivo e poderoso.

Com a versão 2010, o LabVIEW pode realizar desde um projeto e desenvolvimento no PC até diferentes targets embarcados, desde as primeiras versões de prototipos do tamanho de uma torradeira até sistemas embarcados em chips. **O LabVIEW dinamiza o projeto de sistemas com uma única plataforma de desenvolvimento gráfico.** Ao fazer isso, ele abrange o melhor gerenciamento de sistemas distribuídos e em rede, porque na medida que os targets para LabVIEW aumentam em variedade e no modo que são embarcados, é preciso ser capaz de comunicar e distribuir com mais facilidade entre os diversos componentes com código LabVIEW em seus sistema.



Plataformas de hardware integradas

Um instrumento virtual é formado de um computador de padrão industrial ou estação de trabalho equipado com um poderoso software de aplicação, hardware com boa relação custo-benefício (como placas) e drivers, que em conjunto fazem a função de instrumentos tradicionais.

Instrumentos virtuais representam uma mudança fundamental da instrumentação tradicional centrada em hardware para sistemas centrados em software que exploram o potencial computacional, produtividade, apresentação e capacidade de conectividade dos computadores desktop e estações de trabalho.

Embora o PC e a tecnologia de circuitos integrados terem passado por avanços significativos nas últimas duas décadas, o software oferece a flexibilidade para desenvolver nesta poderosa base de hardware as ferramentas para criar instrumentos virtuais, fornecendo melhores maneiras para inovar e reduzir custos significativamente. Com instrumentos virtuais, engenheiros e cientistas criam sistemas de medidas e automação que satisfazem exatamente as necessidades definidas pelo usuário em vez de ser limitado por instrumentos tradicionais de função definida pelo fabricante.

Parte I – O ambiente LabVIEW

Adquirindo dados no computador

- Dispositivos de aquisição de dados
 - NI-DAQmx
 - Aquisição de dados simulada
 - Placa de som
- MAX

O ambiente LabVIEW

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

Componentes de uma aplicação do LabVIEW

- Criando um VI
- Execução por Fluxo de Dados

Ajuda adicional

- Help do LabVIEW
- Dicas para trabalhar com o LabVIEW

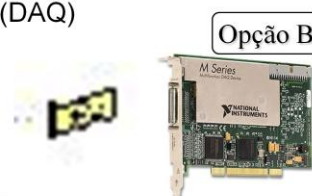
A. Configurando seu Hardware

- Dispositivo de aquisição de dados (DAQ)
 - Dispositivo USB, PCI ou PXI
 - Configuração no Measurement and Automation Explorer (MAX)



Opção A

- Dispositivo de aquisição de dados Simulado (DAQ)
 - Software simulado no driver
 - Configuração no MAX



Opção B

- Placa de som
 - Disponível na maioria dos computadores



Opção C

ni.com

8



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

Cada exercício é dividido em três opções (A, B e C):

Opção A: Desenvolvido para ser usado com hardware suportado pelo driver NI-DAQmx. Com isso, abrange a maioria dos dispositivos de aquisição com barramento USB, PCI e PXI com entrada analógica. Para o uso de um microfone com o dispositivo DAQ é necessário o condicionamento do sinal e sua excitação (alguns sensores, como microfones, requerem potência externa para funcionar, chamada de excitação).

Opção B: Desenvolvido para ser usado sem hardware. O hardware pode ser simulado a partir da versão 8.0 do driver NI-DAQmx. Este é feito usando a opção *NI-DAQmx Simulated Device* no menu *Create New* do MAX. O driver do dispositivo simulado é carregado e o programa que o utiliza é completamente funcional.

Opção C: Desenvolvido para ser usado com uma placa de som padrão e um microfone. O LabVIEW possui VIs para trabalhar com entradas e saídas analógicas usando a placa de som interna disponível em diversos computadores. Isto é muito conveniente para a maioria dos laptops porque possuem a placa de som e o microfone integrado.

Qual dispositivo devo usar?



	Placa de Som*	NI USB DAQ	NI PCI DAQ	Instrumentos*
Amostragem AI	8 a 44 kS/s	10 kS/s a 1.25 MS/s	20 kS/s a 10 MS/s	100 S/s a 2 GS/s
Precisão	12 a 16 bits	12 a 18 bits	12 a 18 bits	8 a 26 bits
Portabilidade	✓	✓	—	some
Canais AI	2	8 a 80	2 a 80	1 a 80
Canais AO	2	2 a 4	2 a 8	2 a 8
AC ou DC	AC	AC/DC	AC/DC	AC/DC
Triggering	—	✓	✓	✓
Calibração	—	✓	✓	✓

* A tabela acima pode não representar todas as variações de equipamentos existentes em cada categoria

ni.com

10



Qual dispositivo eu devo usar?

Há vários tipos de dispositivos de aquisição de dados e controle no mercado. Alguns desses foram apresentados acima. A escolha normalmente é feita pela taxa de amostragem (amostras/segundo), resolução (bits), número de canais e taxa de transferência (normalmente limitada pelo tipo de barramento: USB, PCI, PXI entre outras). Dispositivos DAQ com múltiplas funções (aquisição de dados) são ideais porque eles podem ser usados em uma ampla faixa de aplicações.

USB DAQ de baixo custo NI USB-6008 e USB-6009

O dispositivo USB-6009 da National Instruments fornece funcionalidade básica de aquisição de dados para aplicações como gravação de dados, medidas portáteis e experiências acadêmicas no laboratório. O dispositivo USB-6008 e USB-6009 são ideais para estudantes. Crie aplicações para medição programando o USB-6009 usando LabVIEW e o driver NI-DAQmx para Windows. Para usuários Mac OS X e Linux®, faça o download e use o driver DAQmx Base.

Especificações USB-6009:

- Oito entradas analógicas de 14 bits
- 12 linhas digitais (entrada e saída)
 - 2 saídas analógicas
 - 1 contador

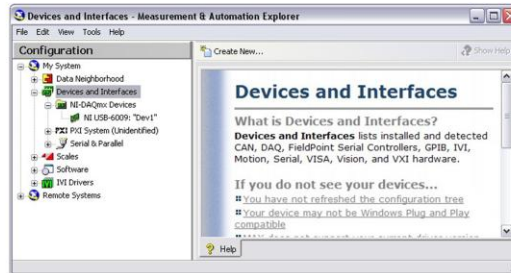


ni.com/daq

Linux® é uma marca registrada de Linus Torvalds nos E.U.A. e em outros países.

O que é MAX?

- Significa *Measurement & Automation Explorer*
- Configura e organiza todos os dispositivos da National Instruments Ex.: DAQ, PCI/PXI, GPIB, IMAQ, IVI, motion, VISA e VXI
- Testes dos dispositivos



ni.com

11



O próximo nível em software é chamado de Measurement & Automation Explorer (MAX). MAX é uma interface de software que fornece acesso a todos os dispositivos da National Instruments. São exemplos DAQ, GPIB, IMAQ, IVI, Motion, VISA e VXI. O atalho para o MAX aparece no área de trabalho do computador após a instalação.

A imagem do ícone é mostrada acima. O MAX é principalmente usado para configurar e testar as ferramentas de hardware da National Instruments, mas também oferece outras funcionalidades, tais como, verificar se a versão mais atual do driver NI-DAQmx está instalada. Quando uma aplicação é executada usando o NI-DAQmx, o software lê a configuração do MAX para determinar os dispositivos que foram configurados. Portanto, é preciso configurar antes o dispositivo DAQ com o MAX.

A funcionalidade do MAX divide-se em seis categorias:

- Data Neighborhood
- Devices and Interfaces
- IVI Instruments
- Scales
- Historical Data
- Software

Este curso irá focar em Data Neighborhood, Devices and Interfaces, Scales, Software e na funcionalidade que cada um oferece.



Exercício 1 – Testando o dispositivo (Opção A)

Neste exercício você usará o Measurement and Automation Explorer (MAX) para testar seu dispositivo DAQ NI USB-600X.

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 o hardware e o software da National Instruments que estão instalados em seu computador.
3. Expanda a seção **NI-DAQmx Devices** para ver o hardware compatível com o driver NI-DAQmx. O número do dispositivo aparece entre aspas seguido do nome. Os VIs de aquisição de dados usam este número para determinar qual dispositivo DAQ realiza a operação. Você verá seu hardware listado como NI USB-600X: “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. Isso testa os recursos do sistema atribuídos a esse dispositivo. O dispositivo deve passar no teste se estiver configurado.
5. Verifique 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 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 da janela. O painel de testes permite que você teste as funções de seu dispositivo, tais como 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 kHz. 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 a linha 0 e veja o LED acender. Clique “Close” para fechar o painel de testes.
9. Feche o MAX.



Exercício 1 – Configurando seu dispositivo (Opção B)

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

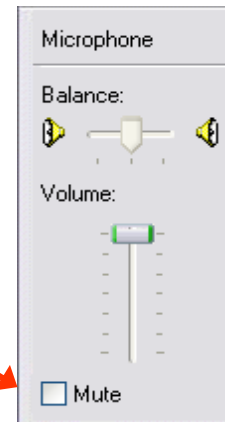
1. Abra o MAX clicando duas vezes no ícone no desktop ou selecionando **Start»Programs»National Instruments»Measurement & Automation**.
2. Expanda a seção **Devices and Interfaces** para ver os dispositivos da National Instruments que estão instalados. O MAX mostra o hardware e o software da National Instruments que estão instalados 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 qual DAQ realiza a operação.
3. Crie um dispositivo DAQ simulado para ser usado mais tarde neste curso. Dispositivos simulados são ferramentas potentes para o desenvolvimento, sem que seja necessário que o hardware esteja 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!
6. 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. Isso testa os recursos do sistema atribuídos a este dispositivo. O dispositivo deve passar no teste pois é um dispositivo simulado.
7. 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 da janela.
8. 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 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.
9. 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 tiver terminado.



Exercício 1 – Configurando seu dispositivo (Opção C)

Neste exercício, você irá usar o Windows utilities para verificar sua placa de som e usá-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. Caso não veja uma seção para o microfone vá para **Options»Properties»Recording** e selecione a opção **Microphone**. Isto mostrará o controle de volume do microfone. Clique “OK”.
3. Tire a seleção da caixa **Mute** se ela ainda estiver selecionada. Verifique que o volume esta em uma altura desejável.



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



Obs.: O método para controlar o volume do microfone para abrir o programa para gravação do som podem ser diferentes de acordo com o sistema operacional utilizado.

(Fim do exercício)

Iniciando o LabVIEW

Start » All Programs » National Instruments » LabVIEW 2010

Getting Started:

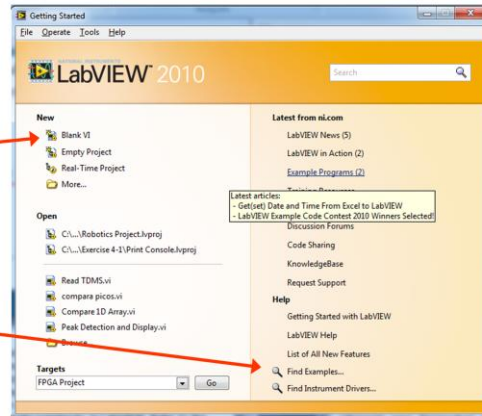
Comece com um VI em branco:

New » Blank VI

ou

Comece com um exemplo:

**Examples » Find
Examples...**



ni.com

17



LabVIEW

LabVIEW é uma linguagem de programação gráfica que utiliza ícones ao invés de linhas de textos para criar aplicações.

Comparando com linguagens de programação baseadas em textos, onde instruções determinam a execução do programa, o LabVIEW usa programação por fluxo de dados, onde o fluxo dos dados determina a ordem de execução.

Pode se comprar várias ferramentas de software que se integram ao LabVIEW (módulos e toolkits) para desenvolvimento de aplicações especializadas. Todas as ferramentas integram-se sem problemas ao LabVIEW. Veja o web site da National Instruments para mais informações sobre essas ferramentas de software.

O LabVIEW também inclui diversos assistentes para ajudar na rápida configuração dos dispositivos DAQ e dos instrumentos baseados em computadores para criar aplicações.

LabVIEW Example Finder

O LabVIEW tem como recurso centenas de VIs de exemplo que podem ser usadas e incorporadas aos VIs criados. Além dos VIs de exemplos integrados ao LabVIEW, pode-se acessar centenas de VIs de exemplos no NI Developer Zone (zone.ni.com). Pode-se modificar um VI de exemplo para este se adequar a sua aplicação ou pode-se copiar e colar de mais de um exemplo para um VI que foi criado.

Programas em LabVIEW são chamados de VIs (Virtual Instruments)

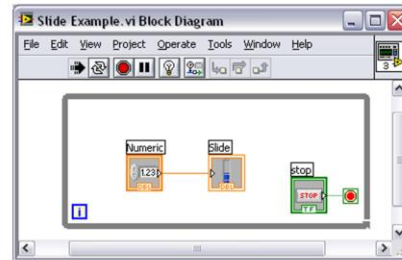
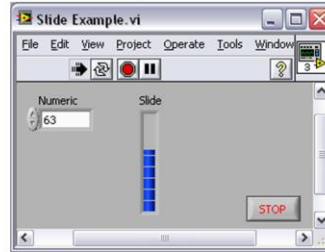
Cada VI tem duas janelas

Painel Frontal

- Interface de usuário (UI)
 - Controles = entradas
 - Indicadores = saídas

Diagrama de Blocos

- Código gráfico
 - Os dados passam pelos fios dos controles através das funções até os indicadores
 - Os blocos são executados por fluxo de dados



ni.com

18



Os programas criados no LabVIEW são chamados de instrumentos virtuais ou simplesmente VIs (virtual instruments).

Controles são entradas e indicadores são saídas.

Cada VI é composto de partes principais:

- **Painel frontal** – Onde o usuário interage com o VI
- **Diagrama de blocos** – O código que controla o programa
- **Ícone/connector pane** – Os modos de conectar o VI a outros VIs

No LabVIEW, constrói-se a interface de usuário usando um conjunto de ferramentas e objetos. A interface de usuário é conhecida como painel frontal. Então adiciona-se o código gráfico representando as funções para controlar os objetos do painel frontal. O diagrama de blocos contém este código. Em alguns aspectos, o diagrama de blocos assemelha-se a um fluxograma.

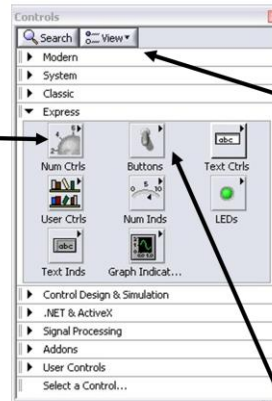
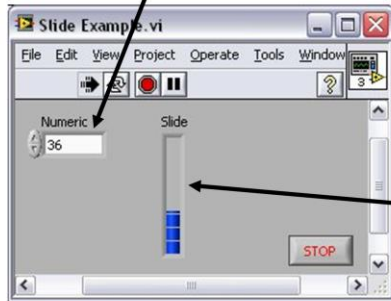
A interação quando o programa está em execução é realizada no painel frontal. Pode-se controlar o programa, mudar entradas e ver os dados atualizados. Controles são usados para entradas como um ajuste no controle slide para definir um valor de alarme, ligar ou desligar uma chave ou parando o programa. Indicadores são usados como saídas. Termômetros, LEDs e outros indicadores exibem os valores de saída do programa. Estes podem incluir dados, estado do programa e outras informações.

Todo controle ou indicador no painel frontal tem um terminal correspondente no diagrama de blocos. Quando um VI é executado, os valores dos controles fluem pelo diagrama de blocos, onde eles são usados em funções e os resultados são passados para outras funções ou indicadores através de fios.

Paleta de controles

(Insira ítems no Painel Frontal)

Controle:
Numérico



Personalize a
visualização
da paleta

Indicador:
Slide Numérico

ni.com

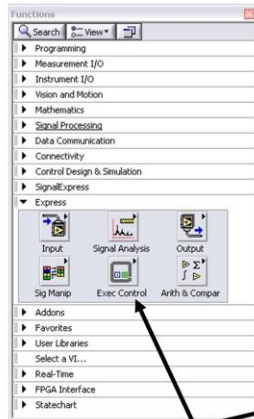
19



Use a paleta de **Controles** para inserir controles e indicadores no painel frontal. A paleta de **Controles** está disponível apenas no painel frontal. Para visualizar a paleta, selecione **View»Controls Palette**.

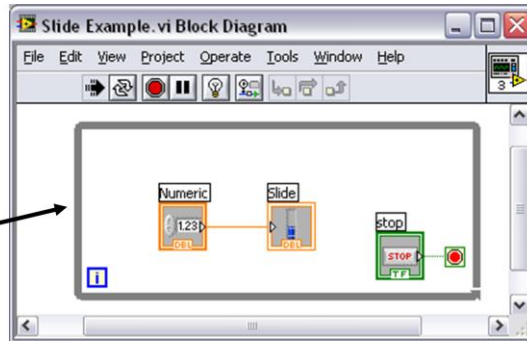
Pode-se também exibir a paleta de **Controles** clicando com o botão direito em uma área livre no painel frontal. Pode-se fixar a paleta de **Controles** clicando o alfinete no canto superior esquerdo da paleta.

Paleta de Funções (e Estruturas)



**Estrutura:
While Loop**

(Insira itens no diagrama de blocos)



ni.com

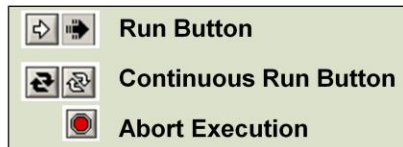
20



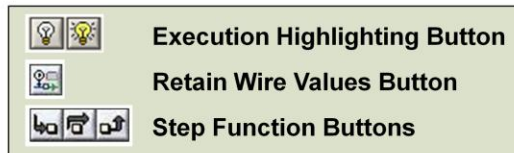
Use a paleta de **Funções** para construir o diagrama de blocos. A paleta de **Funções** está disponível apenas no diagrama de blocos. Para visualizar a paleta, selecione **View» Functions Palette**.

Pode-se também exibir a paleta de **Funções** clicando com o botão direito em uma área livre no diagrama de blocos. Pode-se fixar a paleta de **Funções** clicando no alfinete no canto superior esquerdo da paleta.

Barra de ferramentas



Botões adicionais na barra de ferramentas do Diagrama de blocos



ni.com

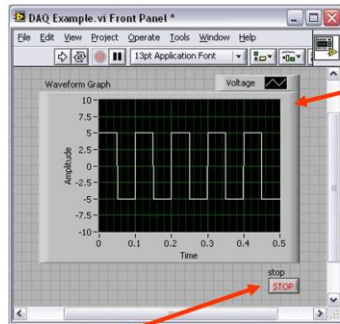
22



- Clique no botão **Run** para executar o VI. Enquanto o VI executa, o botão **Run** aparece com a seta preta se este VI for o VI principal, o que significa 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. Deixe o VI completar seu fluxo de dados ou desenvolva um método para parar o VI no programa. Assim, o VI estará sempre em um estado conhecido . Por exemplo, coloque 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**, o LabVIEW destaca a área do diagrama de blocos onde a execução foi paralisada. Clique novamente no botão **Pause** para continuar com a execução do VI.
- Selecione o menu **Text Settings** e mude as configurações da fonte do VI, incluindo tamanho, estilo e cor.
- Selecione o menu **Align Objects** para alinhar objetos ao longo de eixos, incluindo vertical, superior, esquerda, entre outros.
- Selecione o menu **Distribute Objects** para espaçar objetos uniformemente incluindo espaços, concentração, entre outros.
- Selecione o menu **Resize Objects** para alterar a largura e a altura dos objetos no painel frontal.

Demonstração 1: Criando um VI

Painel Frontal

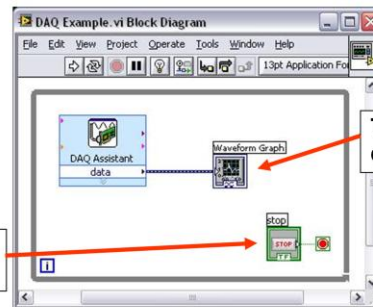


Indicador Gráfico

Controle Booleano

Terminal de entrada

Diagrama de Blocos



Terminal de saída

ni.com

24



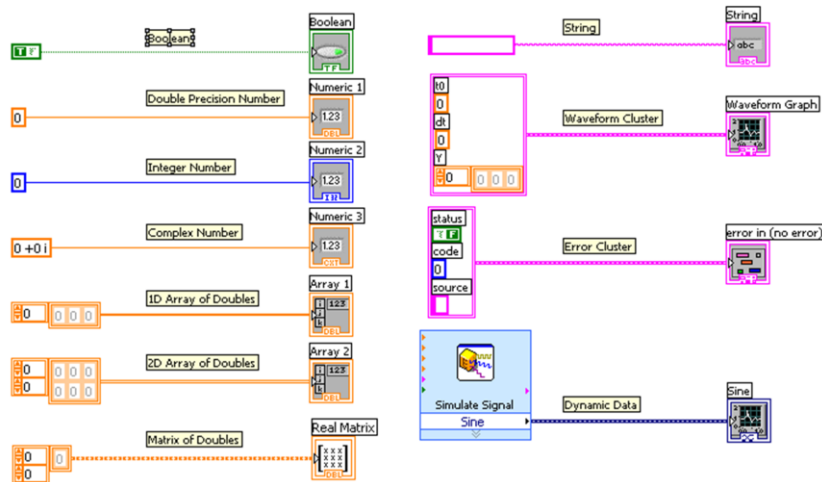
Quando é criado um objeto no painel frontal, um terminal é criado no diagrama de blocos. Estes terminais dão acesso aos objetos do painel frontal a partir do código do diagrama de blocos.

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

Em geral, os terminais devem ser ligados a terminais da mesma cor: verde com verde, azul com azul e assim por diante. Isto não é uma regra sempre válida; o LabVIEW permitirá que um usuário ligue um terminal azul marinho (dado dinâmico) a um terminal laranja (valor fracional), por exemplo. Na maioria dos casos, porém, procure ligar os terminais das mesmas cores.

Controles têm uma seta no lado direito e têm uma borda espessa. Indicadores têm uma seta no lado esquerdo e uma borda fina. Aplica-se regras de lógica nas conexões entre nós no LabVIEW: Cada fio tem de ter uma (e somente uma) fonte (ou controle) e cada fio pode ter múltiplos destinos (ou indicadores)

Tipos de dados do LabVIEW



ni.com

25



O LabVIEW usa muitos tipos de dados comuns, incluindo Booleanos, numéricos, arrays, strings e clusters.

A cor e o símbolo de cada terminal indica o tipo de dado do controle ou indicador.

Para verificar se o terminal corresponde a um controle ou indicador, pode-se ver pela borda deste (um controle possui a borda mais grossa que um indicador) e pelas setas que os terminais possuem (A seta aparece na direita se o terminal é um controle e à esquerda se o terminal é um indicador).

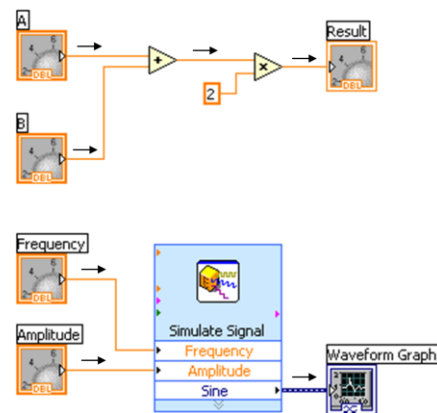
Definições

- **Array:** Arrays agrupam elementos de dados do mesmo tipo. Um array é formado de elementos e dimensões. Elementos são os dados que compõem o array e dimensões são o comprimento, a altura ou profundidade do array. Um array pode ter uma ou mais dimensões e o número de elementos por dimensão de $(2^{31}) - 1$, desde que haja memória.
- **Cluster:** Clusters agrupam elementos de dados de tipos variados, como um grupo de fios em um cabo telefônico, onde cada fio no cabo representa um elemento diferente no cluster.

Verifique **Help»Search the LabVIEW Help...** para mais informações. O *LabVIEW User Manual* no site ni.com fornece referências adicionais para os tipos de dados encontrados no LabVIEW.

Programação por fluxo de dados

- Execução do diagrama de blocos
 - Dependente do fluxo de dados
 - Diagrama de blocos NÃO executa da esquerda para a direita
- Os nós executam quando os dados estão disponíveis em TODOS os terminais de entrada
- Os nós fornecem uma saída para todos os terminais de saída quando finalizado



ni.com

26



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

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 Multiply não é válida enquanto a função Add não estiver finalizada e assim passar os dados para a função Multiply. Lembre que um nó executa somente quando há dados em todos os terminais de entrada e ele fornece dados para todos os seus terminais de saída somente no final da execução. Em um trecho do código, o VI Expresso Simulate Signal recebe dados de controles e passa estes resultados para um controle gráfico.

Pode-se considerar que os códigos discutidos co-existem no mesmo diagrama de blocos paralelamente. Isto significa que ambos começaram a execução ao mesmo tempo e funcionarão independentemente um do outro. Se o computador que estiver executando este código tiver múltiplos processadores, estes dois trechos são executados separadamente (cada um em um processador) sem a necessidade de programação adicional.

Técnicas de depuração

- **Encontrando erros**



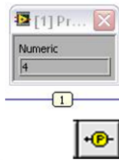
Clique no botão de **Execução (Run)** quebrado. Janela mostrando os erros aparecem.

- **Execução destacada**



Clique no botão **Execução destacada (Execution Highlighting)**. O fluxo de dados é animado e os valores são exibidos nos fios.

- **Probes**



Clique com o botão direito no fio para exibir o probe; Ele mostrará os dados assim que o dado flui pelo fio.

Pode-se também selecionar a ferramenta Probe na paleta de ferramentas e clicar sobre o fio.

ni.com

27



Quando o VI tem um erro, uma seta quebrada é exibida no botão **Run**.

- **Encontrando Erros:** Para listar os erros, clique na seta quebrada. Para localizar os elementos com erro, clique na mensagem.
- **Execução Destacada (Execution Highlighting):** O código é executado de maneira mais lenta e no diagrama é indicado o fluxo de dados nos fios de forma animada, permitindo que você veja valores intermediários. O símbolo na barra de ferramentas é uma lâmpada.
- **Probe:** Usado para ver os valores nos fios, inclusive em arrays e clusters. Clique nos fios com a ferramenta **Probe** ou clique com o botão direito no fio para inserir um probe.
- **Retain Wire Values:** Usado em conjunto com os probes para ver os valores da última iteração do programa. Caso não esteja selecionado, após a execução o valor não fica retido nos fios, sendo necessária nova execução.
- **Breakpoint:** Define pausas em diferentes pontos do diagrama de blocos. Clique nos fios ou nos objetos com a ferramenta **Breakpoint** para colocá-los. É possível utilizá-los no MathScript Node e Formula Node.



Exercício 2 – Adquirindo um sinal com um dispositivo DAQ (Opção A)

Complete os seguintes passos para criar uma VI que irá adquirir 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 Block Diagram**. Leia as instruções que estão escritas lá sobre como completar o programa.
5. Dê um clique duplo no DAQ Assistant para abrir o assistente de configuração.
6. Configure uma operação de entrada analógica.
 - a. Escolha **Acquire Signal»Analog Input»Voltage**.
 - b. Escolha **Dev1 (USB-600X)»ai0** para adquirir dados 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, na aba Setting, entre com **5 Volts** para o máximo e **-5 Volts** para o mínimo.No canto inferior da caixa de diálogo, em Time Settings, escolha **Continuous Samples** no combo box **Acquisition Mode** entre com o valor **10k** no controle Rate. Deixe as demais opções com seus valores padrões. Clique em “OK” para sair do assistente.
7. Coloque o VI Expresso Filter à direita de seus DAQ Assistant no diagrama de blocos. Da paleta de funções selecione: **Express»Signal Analysis»Filter** e o coloque no diagrama de blocos dentro de um loop While. Quando você abrir a paleta de funções, selecione o pequeno botão no canto esquerdo superior da paleta. Isso irá prender a paleta para que ela não desapareça. Este passo será omitido nos demais exercícios mas deve ser repetido. Na janela de configuração do VI Expresso, dentro de **Filtering Type**, selecione **Highpass**. Dentro de **Filter Specifications**, na opção **Cutoff Frequency (Hz)**, use o valor 300 Hz. Clique OK.



Exercício 2 – Adquirindo um sinal com a placa de som (Opção C)

Complete os seguintes passos para criar uma VI que irá adquirir dados com a placa de som.

1. Abra o LabVIEW.
2. Na janela **Getting Started**, clique em **Blank VI**.
3. Retorne ao Painel Frontal selecionando **Window»Show Front Panel** ou pressionando o atalho <Ctrl+E>.
4. Coloque o VI Acquire Sound Express no diagrama de blocos. Clique com o botão direito do mouse para abrir a paleta de funções e selecione **Express»Input»Acquire Sound**. Coloque o VI Expresso no diagrama de blocos.
5. Na janela de configurações em **#Channels**, selecione **1** da lista e clique “OK”.
6. Coloque o VI Expresso Filter à direita do VI Acquire Sound no diagrama de blocos. Da paleta de funções, selecione: **Express»Signal Analysis»Filter** e coloque-o no diagrama de blocos. Na janela de configuração do VI Expresso, dentro de **Filtering Type**, selecione **Highpass**. Dentro **Filter Specifications**, na opção **Cutoff Frequency (Hz)**, use o valor 300 Hz. Clique OK.
7. Faça as seguintes conexões no Diagrama de Blocos, colocando seu mouse sobre os terminais no diagrama de blocos, para que este se torne a ferramenta de fiação (wiring tool) e clique uma vez sobre cada terminal que irá ser conectado*:
 - a. Conecte o terminal de saída “Data” do VI Acquire Sound para o terminal de 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 “Filtered Signal” do VI Filter e escolha **Create»Graph Indicator**.
8. Retorne ao Painel Frontal selecionando **Window»Show Front Panel** ou pressionando o atalho <Ctrl+E>.
9. Execute seu programa clicando no botão “Run”. Faça um barulho ou bata no microfone para observar a variação de tensão no gráfico.
10. Clique no botão “Stop” uma vez que esteja finalizado.
11. Salve o VI como “Exercício 2 – Acquire.vi” na pasta Exercises e feche-o.

***Utilizada opção de escolha de Ferramenta Automática**

Nota: A solução deste exercício está no final deste manual

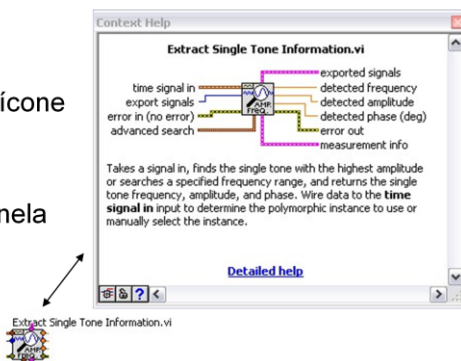
(Fim do exercício)

Janela Context Help

- **Help » Show Context Help** ou pressione as teclas <Ctrl+H>
- Passe o cursor sobre o objeto para atualizar a janela

Ajuda adicional

- Clique com o botão direito no ícone do VI e selecione **Help**
- ou
- Escolha “**Detailed help**” na janela context help



ni.com

33



A Janela **Context Help** mostra informações básicas sobre os objetos do LabVIEW quando o cursor passa sobre cada item. Os seguintes objetos possuem informações no **Context Help**: VIs, funções, constantes, estruturas, paletas, propriedades, métodos, eventos, e componentes de caixa de diálogo.

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

Conexões mostradas no Context Help:

Obrigatório – negrito

Recomendado – normal

Opcional – cinza

Ajuda Adicional

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

Dicas para trabalhar no LabVIEW

- Atalhos no teclado
 - <Ctrl+H> – Ativa/Desativa a janela Context Help
 - <Ctrl+B> – Remove fios quebrados do Diagrama de Blocos
 - <Ctrl+E> – Alterna entre o Painel Frontal e Diagrama de Blocos
 - <Ctrl+Z> – Undo (também no menu Edit)
- **Tools » Options...** – Define as preferências no LabVIEW
- **File » VI Properties** – Configura a aparência do VI, Documentação e etc

O LabVIEW possui diversos atalhos no teclado para facilitar seu trabalho. Os atalhos mais comuns estão listados acima.

Apesar da ferramenta de seleção automática ser ótima para escolher a ferramenta que será usada no LabVIEW, as vezes o controle manual da ferramenta pode ser útil. Uma vez que a ferramenta de seleção está selecionada para manual, você pode usar a tecla Tab para alternar entre as quatro ferramentas mais comuns (Operar Valores, Posição/Tamanho/Seleção, Editar Texto e Definir Cores no Painel Frontal ou as três primeiras ferramentas mais Conectar Fios no Diagrama de Blocos). Se quiser retornar a seleção da ferramenta para automática, pode-se fazer isso com o atalho <Shift+Tab>.

Na caixa de diálogo **Tools»Options...**, há diversas opções configuráveis para personalizar seu painel frontal, diagrama de blocos, documentação e muito mais.

Similar as opções do LabVIEW, pode-se configurar as propriedades específicas do VI no **File»VI Properties...** As opções incluem documentar o VI, mudar a aparência da janela do painel frontal e personalizar o VI de outras formas.

Parte II – Elementos dos programas

Loops

- While Loop
- For Loop


Funções e SubVIs

- Tipos de funções
- Procurando controles, VIs e funções

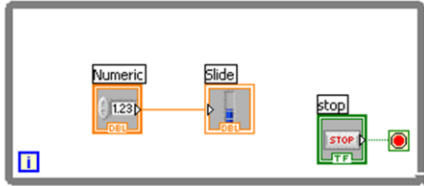
Tomada de decisão e gravação e leitura de arquivos

- Estrutura Case
- Select (declaração "If")
- Gravação e leitura de arquivos

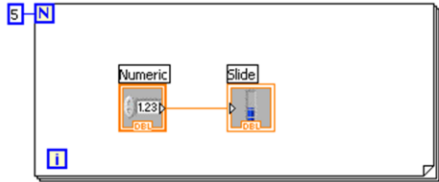
Loops


- While Loop
 - **I** Terminal conta o número de iterações
 - Executa pelo menos uma vez
 - Executa até que a condição de parada é atingida 
- For Loop
 - **I** Terminal conta o número de iterações
 - O número de execuções é determinado pela entrada **N**

While Loop



For Loop



ni.com
36


O loop for e while estão localizados na paleta de **Functions»Structures**.

O loop for executa um número determinado de vezes e tem nessa característica a principal diferença do loop while, pois o loop while para a execução no subdiagrama apenas se o valor no terminal condicional acontecer.

Loop while

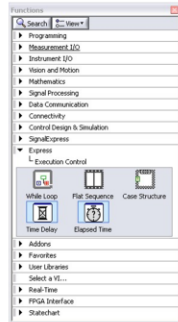
Similar a um *do loop* ou um *repeat-until loop* em linguagens de programação baseadas em texto, um loop while loop, mostrado na figura mais superior, executa o subdiagrama até que a condição seja atingida. Sendo assim, o loop while executa o subdiagrama até que o terminal condicional, um terminal de entrada, recebe um valor Booleano específico. O comportamento e aparência padrão do terminal condicional é **Stop If True**. Quando o terminal condicional é **Stop If True**, o loop while executa o subdiagrama até que terminal condicional receba um valor Verdadeiro (**True**). O terminal de iteração (um terminal de saída), contém o número de iterações completas. A contagem das iterações sempre começa em zero. Durante a primeira iteração, o terminal de iteração retorna 0.

Loop for

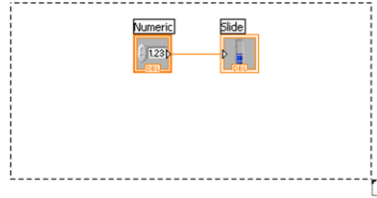
Um loop for, mostrado acima, executa um número determinado de vezes o subdiagrama. O valor no terminal de contagem (um terminal de entrada) representa **N**, indica quantas vezes o subdiagrama irá repetir. O terminal de iteração (um terminal de saída), contém o número de iterações completas. A contagem das iterações sempre começa em zero. Durante a primeira iteração, o terminal de iteração retorna 0.]

Criando um loop

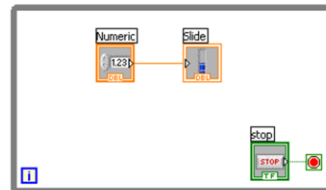
1. Selecione a estrutura



2. Inclua o código a ser repetido



3. Insira e retire nós e fios adicionais



ni.com

37

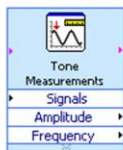


Insira loops no diagrama selecionando estes na paleta de **Funções**:

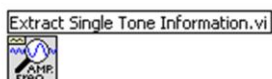
- Quando selecionado, o cursor do mouse se torna um ponteiro especial usado para selecionar a seção do código que será colocada dentro do Loop.
- Clique o botão do mouse para definir o canto superior e então, clique com o botão do mouse novamente no canto inferior direito. O limite do loop while é criado em volta do código selecionado.
- Clique e arraste nós adicionais para dentro do loop while se for necessário.

3 Tipos de funções (da Paleta de Funções)

VIs Expressos: VIs interativos, configurados através de caixas de diálogo para configurable dialog page (**borda azul**)



VIs Padrões: VIs modularizados configurados por fios (**personalizado**)



Funções: Elementos de operações fundamentais do LabVIEW; não possui painel frontal ou diagrama de blocos (**amarelo**)



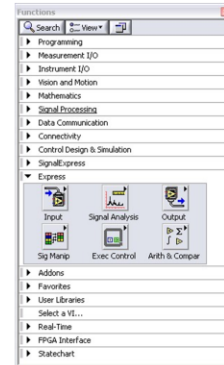
O LabVIEW 7.0 introduziu um novo tipo de subVI chamado de VIs expressas. Estes VIs interativos tem uma caixa de diálogo de configuração que auxilia o usuário a personalizar a funcionalidade da VI Expressa. O LabVIEW gera então um subVI baseado nestas configurações.

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

Funções são os blocos que formam todos os VIs. Funções não possuem painel frontal ou diagrama de blocos.

Quais os tipos de funções disponíveis?

- **Entrada e Saída**
 - Simulação de dados e sinais
 - Aquisição e geração de sinais reais com DAQ
 - Controle de instrumentos (Serial e GPIB)
 - ActiveX para comunicação com outros programas
- **Análise**
 - Processamento de sinal
 - Estatísticas
 - Matemática avançada e fórmulas
 - Atuador de tempo contínuo
- **Armazenamento**
 - Gravação e Leitura de Arquivos



Paleta de funções expressas

ni.com

39



O LabVIEW possui centenas de funções criadas para ajudar na aquisição, análise e apresentação dos dados. Estas serão normalmente usadas como descritas no slide acima.

Toolkits do LabVIEW

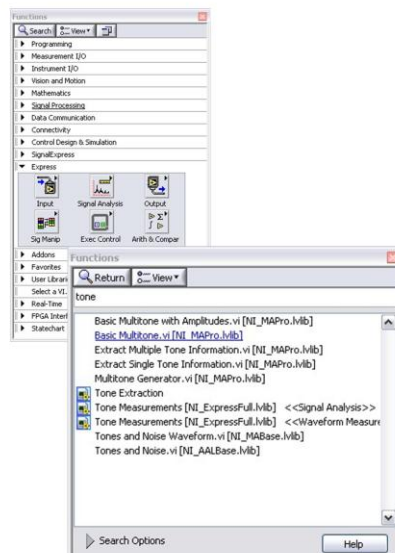
Toolkits adicionais estão disponíveis para acrescentar funções específicas ao LabVIEW. Estes toolkits incluem:

Módulos de implantação de aplicações e targets <ul style="list-style-type: none"> * LabVIEW Mobile Module * LabVIEW Real-Time Module * LabVIEW FPGA Module * NI Vision Development Module for LabVIEW Sistemas Embarcados <ul style="list-style-type: none"> * DSP Test Integration Toolkit * Embedded test integration toolkits * Digital Filter Design Toolkit * LabVIEW FPGA Module 	Processamento e análise de sinal <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 Engenharia de software e ferramentas de otimização <ul style="list-style-type: none"> * Real-Time Execution Trace Toolkit * Express VI Development Toolkit * State Diagram Toolkit * VI Analyzer Toolkit 	Projeto de controle e simulação <ul style="list-style-type: none"> * LabVIEW Control Design and Simulation Module * LabVIEW Real-Time Module * System Identification Toolkit * State Diagram Toolkit Processamento e aquisição de imagem <ul style="list-style-type: none"> * Vision Development Module for LabVIEW * NI Vision Builder for Automated Inspection * NI-IMAQ for IEEE 1394
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

ni.com/toolkits

Procurando por controles, VIs e funções

- Paletas são compostas com centenas de VIs
- Pressione o botão “Search” para classificar por índice a procura de todos os VIs
- Clique e arraste um item do índice de procura para diagrama de blocos
- Duplo clique no item para mostrar onde ele está localizado na paleta



ni.com

40



Use os botões no topo das paletas para navegar, pesquisar e editar as paletas.

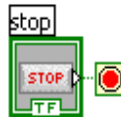
Você pode procurar controles, VIs e funções que contenham as palavras ou que comecem com essas palavras. Dando um duplo clique em um resultado de busca, é aberta a paleta que apresenta os resultados da busca. Você também pode clicar e arrastar o nome do controle, VI ou função diretamente para o painel frontal ou diagrama de blocos.



Exercício 3 – Análise - Sinal simulado (Opção A, B e C)

Crie um VI que produz uma onda senoidal com uma frequência específica e exiba os dados em um Waveform Chart até ser finalizado pelo usuário.

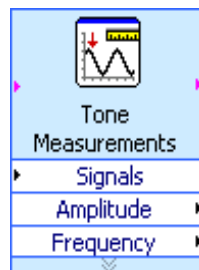
1. Abra um VI em branco a partir da tela **Getting Started**.
2. Coloque um gráfico no painel frontal. Clique com o botão direito do mouse para abrir a Paleta de Controles e selecione **Controls»Modern»Graph»Waveform Chart**.
3. Coloque um controle Dial no painel frontal. Da Paleta de Controles selecione **Controls»Modern»Numeric»Dial**. Quando um controle é colocado no Painel Frontal, o nome está destacado. Enquanto ele estiver destacado, digite “Frequency In” para dar um nome a este controle.
4. Mude para o diagrama de blocos (<Ctrl+E>) e coloque um Loop While. Clique com o botão direito do mouse para abrir a 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 o indicador Waveform Chart e o controle Dial e coloque dentro do Loop caso ainda não estejam. O botão Stop já está conectado ao terminal condicional do loop while.



5. Insira o VI Expresso Simulate Signal no diagrama de blocos. Na Paleta de Funções, selecione **Express»Signal Analysis»Simulate Signal** e coloque-o dentro do Loop While no diagrama de blocos. Na janela de configuração, no item Timing, escolha “Simulate acquisition timing”. Clique “OK.”



6. Insira o VI Expresso Tone Measurements no Diagrama de Blocos (**Express»Signal Analysis»Tone Measurements**). Na janela de configuração, selecione “Amplitude” e “Frequency” no item “Single Tone Measurements”. Clique “OK.”





Exercício 4 – Análise - Sinal (Opção A)

Crie uma VI que realize a medição da frequência e amplitude de sinal do seu dispositivo DAQ e exiba o sinal adquirido em um Waveform Chart. As instruções são as mesmas que no Exercício 3, mas o VI DAQ Assistant é usado no lugar do VI Simulate Signal. Tente fazer sem usar as instruções abaixo.

1. Abra um VI em branco.
2. Coloque um Waveform Chart no Painel Frontal. Clique com o botão direito do mouse para abrir a paleta de controles e selecione **Controls»Modern»Graph»Waveform Chart**.
3. Coloque um indicador numérico meter no painel frontal. O indicador meter é encontrado no caminho **Controls»Moden»Numeric»Meter**.
4. Clique com o botão direito no eixo y do Waveform Chart e retire a seleção da opção **“AutoScale Y.”**
5. Mude a escala do eixo y para -0.15 e 0.15 v clicando duas vezes sobre os valores de máximo e mínimo do eixo e escrevendo os novos valores. Mude o fundo de escala do meter de 100 para 2000.
6. Mude para o diagrama de blocos e coloque um Loop While em torno do chart e do meter (**Express»Execution Control»While Loop**).
7. Coloque o VI DAQ Assistant no diagrama de blocos (**Express»Input»DAQ Assistant**). Escolha a entrada analógica canal 0 de seu dispositivo (real ou simulado) e clique “Finish”. Em Time Settings, escolha Continuous Samples no combo box Acquisition Mode. Se estiver utilizando o USB-600X, mude o intervalo de entrada de -5 até 5 na aba Setting e o número de samples a serem lidos para 100.
8. Coloque o VI Filter Express à direita do DAQ Assistant no diagrama de blocos. Na 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 Filter Specifications, na opção “Cutoff Frequency,” use o valor de 300 Hz e clique “OK.”
9. Conecte o terminal de saída “Data” do DAQ Assistant à entrada “Signal” do VI Filter.
10. Conecte o terminal “Filtered Signal” no VI Filter ao Waveform Chart.
11. Coloque o VI Expresso Tone Measurements Express VI no diagrama de blocos (**Express»Signal Analysis»Tone**). Na janela de configuração, escolha as medidas “Amplitude e Frequency” no ítem Single Tone Measurements.
12. Conecte a saída do VI Filter à entrada “Signals” do VI Expresso Tone Measurements. Conecte também a saída “Frequency” ao meter.
13. Retorne ao Painel Frontal e execute o VI. Observe seu sinal adquirido e sua frequência e amplitude. Faça um barulho ou bata em seu microfone se você tiver um USB-600X e observe a amplitude e frequência que você está produzindo.
14. Salve o VI como “Exercício 4 - Data.vi”.
15. Feche o VI.

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

(Fim do exercício)



Exercício 4 – Análise - Sinal (Opção B)

Crie uma VI que realize a medição da frequência e amplitude de sinal do seu dispositivo DAQ simulado e mostre o sinal adquirido em um Waveform Chart. As instruções são as mesmas que no Exercício 3, mas o VI DAQ Assistant é usado no lugar do VI Simulate Signal. Tente fazer sem usar as instruções abaixo.

1. Abra um VI em branco.
2. Coloque um Waveform Chart no Painel Frontal. Clique com o botão direito do mouse para abrir a paleta de controles e selecione **Controls»Modern»Graph»Waveform Chart**.
3. Coloque um indicador numérico meter no painel frontal. O indicador meter é encontrado no caminho **Controls»Moden»Numeric»Meter**.
4. Mude para a diagrama de blocos e coloque um Loop While (**Express»Execution Control»While Loop**).
5. Coloque o VI DAQ Assistant no diagrama de blocos (**Express»Input»DAQ Assistant**). Escolha a entrada analógica canal 0 de seu dispositivo (real ou simulado) e clique "Finish". Em Time Settings, escolha Continuous Samples no combo box Acquisition Mode. Mantenha as outras configurações.
6. Coloque o VI Filter Express à direita do DAQ Assistant no diagrama de blocos. Na 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 Filter Specifications, na opção "Cutoff Frequency", use o valor de 300 Hz e clique "OK."
7. Conecte o terminal de saída "Data" do DAQ Assitante à entrada "Signal" do VI Filter.
8. Conecte o terminal "Filtered Signal" no VI Filter ao Waveform Chart.
9. Coloque o VI Expresso Tone Measurements Express VI no diagrama de blocos (**Express»Signal Analysis»Tone**). Na janela de configuração, escolha as medidas "Amplitude e Frequency" no item Single Tone Measurments.
10. Crie indicadores para as medidas de amplitude e frequência clicando com o botão direito do mouse em cada terminal do VI Expresso Tone Measurment e selecione **Create»Numeric Indicator**.
11. Conecte a saída do Filtro à entrada "Signals" do VI Expresso Tone Measurements.
12. Retorne ao Painel Frontal e execute o VI. Observe seu sinal adquirido e sua frequência e amplitude. Faça um barulho ou bata em seu microfone se você tiver um USB-600X e observe a amplitude e frequência que você está produzindo.
13. Salve o VI como "Exercício 4 - Data.vi".
14. Feche o VI.

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

Apesar da placa ser simulada, sua utilização no LabVIEW é similar ao dispositivo real

(Fim do exercício)



Exercício 4 – Análise - Sinal real (Opção C)

Crie uma VI que realize a medição da frequência e amplitude do sinal da sua placa de som e exiba o sinal adquirido em um Waveform Chart. As instruções são as mesmas que no Exercício 3, mas o VI Sound Signal é usado no lugar do VI Simulate Signal. Tente fazer sem usar as instruções abaixo.

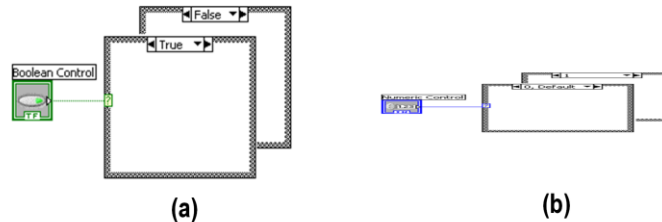
1. Abra um VI em branco.
2. Coloque um Waveform Chart no Painel Frontal. Clique com o botão direito do mouse para abrir a paleta de controles e selecione **Controls»Modern»Graph»Waveform Chart**.
3. Coloque um indicador numérico meter no painel frontal. O indicador meter é encontrado no caminho **Controls»Modern»Numeric»Meter**.
4. Clique com o botão direito no eixo y do Waveform Chart e retire a seleção da opção **“AutoScale Y.”**
5. Mude a escala do eixo y para -1 e 1 v clicando duas vezes sobre os valores de máximo e mínimo do eixo e escrevendo os novos valores. Mude o fundo de escala do meter de 100 para 2000.
6. Mude para o diagrama de blocos e coloque um While Loop (**Express»Execution Control»While Loop**).
7. Coloque o VI Expresso Acquire Sound no diagrama de blocos (**Express»Input»Acquire Sound**).
8. Mude a taxa de amostragem para 44100 e selecione “OK.”
9. Coloque um VI Expresso no Diagrama de Blocos. Na janela de configuração escolha a opção Highpass em “Filtering Type” e use o valor de 300 Hz na opção “Cutoff Frequency (Hz)” em Filter Specifications.
10. Coloque um VI Expresso Tone Measurements no Diagrama de Blocos (**Express»Signal Analysis»Tone**). Na janela de configuração, selecione as medidas “Amplitude” e “Frequency” na seção Single Tone Measurements.
11. Conecte o meter na saída “Frequency” do VI Tone Measurements.
12. Conecte o terminal “Data” do VI Acquire Sound à entrada “Signal” do VI Filter.
13. Conecte o terminal “Filtered Signal” do VI Filter à entrada “Signals” do VI Tone Measurements. Conecte também essa saída ao waveform chart.
14. Retorne ao Painel Frontal e execute o VI. Observe seu sinal da sua placa de som e sua frequência e amplitude. Faça um barulho ou bata em seu microfone e observe a amplitude e frequência que você está produzindo.
15. Salve o VI como “Exercício 4 - Data.vi”. Feche o VI.

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

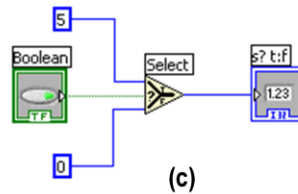
(Fim do exercício)

Como tomar uma decisão no LabVIEW?

1. Estruturas Case



2. Select



ni.com

46



Estrutura Case

A estrutura Case tem um ou mais subdiagramas, ou casos, em que somente um será executado, quando a estrutura for executada, de acordo com o valor do terminal seletor. O terminal seletor pode ser do tipo Booleano, string, inteiro, ou enum. Clique com o botão direito do mouse na borda da estrutura para adicionar ou deletar casos. Use a ferramenta de texto para inserir valores no label do seletor de casos e configure os valores administrados por cada caso. Essa estrutura é encontrada em **Functions»Programming»Structures»Case Structure**.

- **Exemplo a:** Entrada Booleana no terminal seletor: Simple Case se-então (if-then). Se a entrada Booleana é TRUE, este caso será executado; Caso contrário, o caso FALSE será executado.
- **Exemplo b:** Entrada Numérica no terminal seletor: O valor de entrada determina que caso será executado. Caso o valor no terminal seletor não tenha sido definido em nenhum dos casos programados, o LabVIEW executa o caso padrão (default).

Select

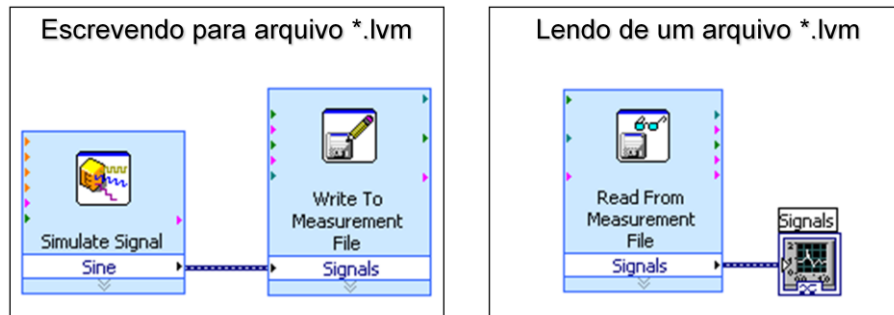
Retorna os valores conectados aos terminais de entradas **t** ou **f**, dependendo do valor do terminal de entrada **s**. Se no terminal **s** o valor for TRUE, a função retorna o valor ligado ao terminal **t**. Caso contrário, a função retorna o valor ligado ao terminal **f**. Esta é uma função polimórfica que pode ter em sua saída diversos tipos de dados (dependendo dos tipos de dados conectados aos terminais de entrada **t** e **f**). Essa função encontra-se em **Functions»Programming»Comparison»Select**.

Exemplo c: Quando o valor Booleano na entrada **s** da função Select é TRUE, o valor 5 é passado para o indicador. Quando o valor Booleano na entrada **s** da função Select é FALSE o valor 0 é passado para o indicador.

Gravação e Leitura de Arquivo

Gravação e Leitura de Arquivo – Utilizar dados em arquivos

- Arquivos podem ser binários, texto ou planilha
- Write/Read LabVIEW Measurements file (*.lvm)



ni.com

47



Use o tipo de arquivo LabVIEW measurement para salvar os dados gerados pelo VI Expresso Write Measurement File. O arquivo de dados LabVIEW measurement é um arquivo de texto data delimitado por tab que pode ser aberto com aplicações do tipo planilha ou editor de texto. Além dos dados, o VI Expresso gera o arquivo .lvm que inclui informações sobre os dados, tais como o tempo e a data que os dados foram gerados.

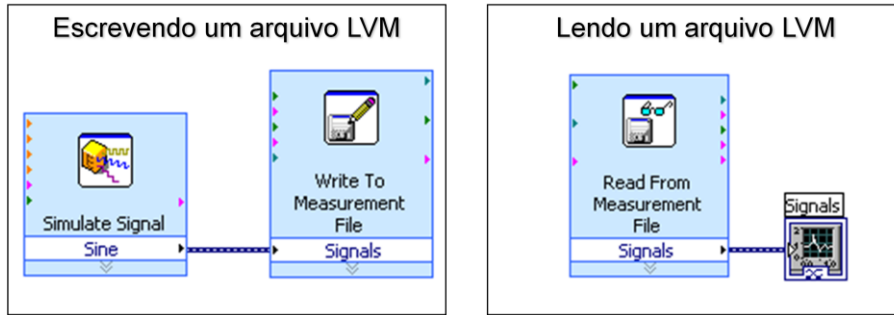
As operações de gravação e leitura de arquivo passam dados da memória para arquivos ou de arquivos para a memória. No LabVIEW você pode usá-las para:

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

No próximo exercício, verifique como escrever ou ler de um arquivo do tipo LabVIEW measurement files (arquivos *.lvm).

Funções de alto nível para Gravação/Leitura de arquivos

- Fáceis de usar
- Alto nível de abstração



ni.com

48



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

- **Write to a Spreadsheet File** – Converte um array de uma ou duas dimensões do tipo numérico ponto flutuante (double precision) em uma string e escreve esta em um novo arquivo ASCII ou acrescenta a string em um arquivo existente.
- **Read from a Spreadsheet File** – Lê um número especificado de linhas ou colunas de um arquivo de texto numérico começando em um caracter especificado e converte os dados em um array de duas dimensões do tipo numérico ponto flutuante (double-precision). Deve-se selecionar manualmente a instância polimórfica desejada. O VI abre o arquivo antes de iniciar a leitura e fecha este após a realização
- **Write to Measurement File** – VI Expresso que escreve dados em um arquivo de medidas no formato texto (.lvm) ou em um arquivo de medidas no formato binário (.tdm/ .tdms).
- **Read from Measurement File** – VI Expresso que lê dados de um arquivo de medidas no formato texto (.lvm) ou em um arquivo de medidas de formato binário (.tdm/ .tdms). Pode-se especificar o nome, formato e tamanho do segmento do arquivo.

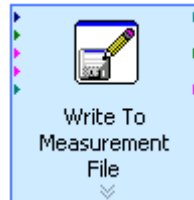
Estas funções são fáceis de usar e são excelentes para aplicações simples. No caso da escrita e/ou leitura desse arquivo ter de ser feita em alta velocidade (Streaming) de forma contínua, algumas dessas funções podem ter um excesso de cabeçalho e assim torná-la lenta ou corromper o arquivo.



Exercício 5 – Tomando decisão e salvando dados (Opção A, B e C)

Crie um VI que permite que você salve seus dados em um arquivo se o valor da frequência dos dados estiver abaixo de um limite definido pelo usuário.

1. Abra o Exercício 4 – Data.vi.
2. Abra **File»Save As...** e salve como “Exercício 5 – 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. Para o caso “True”, acrescente um VI Expresso Write to Measurement File (**Functions»Programming»File I/O»Write to Measurement File**).

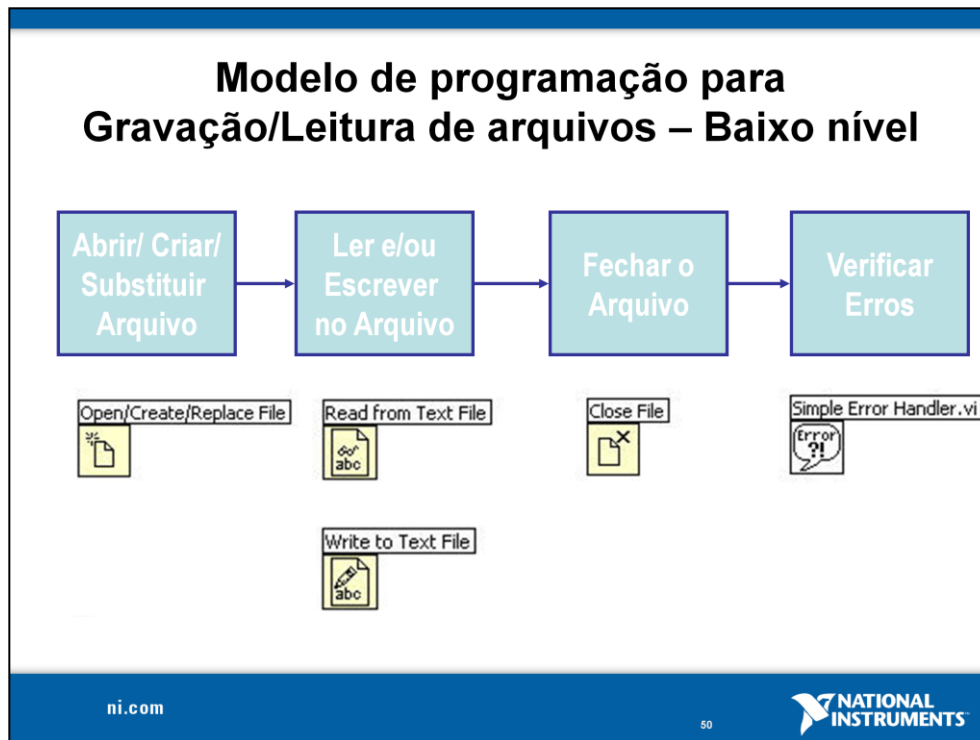


- a. Na janela de configurações, escolha selecione a opção “Save to series of files (multiple files)”. Verifique qual é o local default para salvar seu arquivo e mude se quiser.
 - b. Clique em “Settings...” e escolha “Use next available file name” em **Existing Files**.
 - c. Em **File Termination** escolha para começar um novo arquivo a cada 10 segmentos. (opção “After n segments”). Clique “OK” duas vezes.
5. Acrescente o código para que a frequência computada pelo VI Expresso Tone Measurements seja salva caso este fique abaixo do limite. Dica: Vá para **Functions»Programming»Comparison»Less?**
 6. Lembre-se de conectar os dados do DAQ Assistant ou do Acquire Sound à entrada “Signals” do Measurement File VI. Se você precisar de ajuda, veja a solução deste exercício.
 7. Mude ao Painel Frontal e execute seu VI. Varie o limite de frequência e pare o VI.
 8. Navegue para a pasta **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.

Notas: A solução deste exercício está no final deste manual

Se estiver utilizando um dispositivo DAQ simulado, não será possível modificar a frequência. Pense em uma implementação diferente, como usar um botão para gravar o dado ou utilizar como função de entrada um VI Simulate Signal, como feito no exercício 3.

(Fim do exercício)



Modelo de Programação para os VIs de Arquivos Intermediários

Este mesmo modelo de programação se aplica para aquisição de dados, controle de instrumentos, gravação e leitura de arquivos e a maioria das formas de comunicação. Na maioria das instâncias, é aberto o arquivo ou o canal de comunicação, é possível então ler e escrever várias vezes e ao final a comunicação é fechada ou encerrada. É também uma boa prática de programação verificar se houve erros no final. Lembre-se deste modelo de programação quando estudar programação mais avançada ou estudar outros temas, como DAQ, comunicação entre redes e protocolos.

VIs e Funções para gravação/leitura arquivos

Use as funções e VIs para gravação/leitura de arquivos para realizar diversas tarefas com os arquivos, tais como abrir ou fechar os arquivos; ler e escrever dados nos arquivos; criar diretórios e arquivos com os caminhos definidos; obter informações sobre os diretórios e escrever strings, números, arrays e clusters em arquivos.

Use os VIs para gravação/leitura de arquivos de alto nível localizados na parte superior da paleta para realizar operações comuns, como escrever ou ler vários tipos de dados. Dependendo do tipo de dados utilizado é possível incluir caracteres ou linhas em um arquivo de texto, arrays de uma ou duas dimensões contendo numéricos ponto flutuantes em arquivos de planilhas de texto, arrays de uma ou duas dimensões contendo numéricos ponto flutuantes em arquivos binários ou inteiros sinalizados de 16 bits em arquivos binários.

Use VIs e funções de baixo nível para gravação/leitura de arquivos localizadas no meio da paleta e a subpaleta Advanced File Functions para controlar cada operação de gravação/leitura de arquivos individualmente. As funções de baixo nível podem realizar as tarefas básicas (criar ou abrir e fechar arquivos, escrever e ler dados) como podem criar diretórios: mudar, copiar ou deletar arquivos; listar conteúdo de diretórios; mudar características de arquivos ou manipular caminhos.

Consulte o NI Developer Zone para mais informações sobre como escolher um formato de arquivo.

Parte III – Apresentando os resultados

Apresentando dados no painel frontal

- Controles e indicadores
- Graphs e charts
- Temporizando um loop

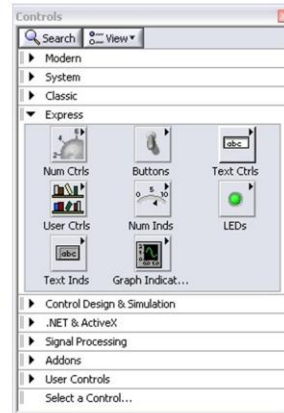
Processamento de sinal

- Arrays
- LabVIEW MathScript

Quais os tipos de controles e indicadores disponíveis?

- **Dados numéricos**
 - Number Input e Display
 - Analog Sliders, Dials e Gauges
- **Dados Booleanos**
 - Botões e LEDs
- **Dados em array e matrix**
 - Numeric Display
 - Chart
 - Graph
 - XY Graph
 - Intensity Graph
 - 3D Graph: Point, Surface, and Model
- **Decorações**
 - Tab Control
 - Arrows
- **Outros**
 - Strings e Text Boxes
 - Picture/Image Display
 - ActiveX Controls

Paleta de controles expressos



ni.com

52



Controles e Indicadores são itens do painel frontal que permitem ao usuário interagir com o programa, fornecendo informações de entrada e obtendo resultados. Pode-se acessar a **paleta de controles** (contendo controles e indicadores) clicando com o botão no painel frontal.

Além disso, controles e indicadores adicionais são incluídos quando são instalados módulos e toolkits.

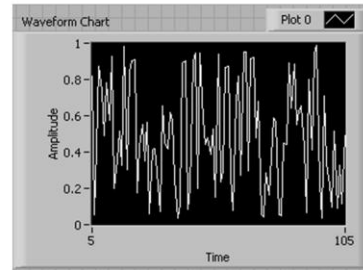
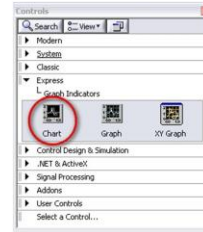
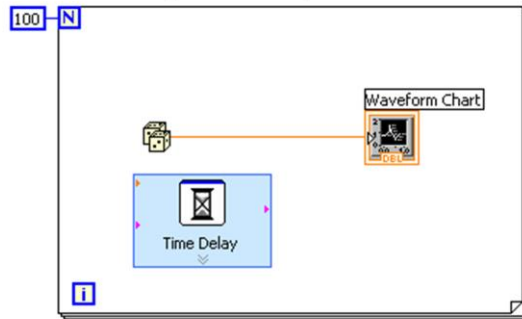
Por exemplo, quando são instaladas as ferramentas de Control Design, são incluídos gráficos especializados como o de Bode e Nyquist por definição.

Charts – Adciona um ponto de dado por vez com histórico

Waveform chart – Indicador numérico especial que pode exibir o histórico dos valores

- Chart atualiza a cada ponto que recebe

Controls » Express » Graph Indicators » Chart



ni.com

53



O Waveform Chart é um indicador numérico especial que mostra uma ou mais linhas em um gráfico e está localizado no Paleta de Controle no caminho **Controls»Modern»Graph**.

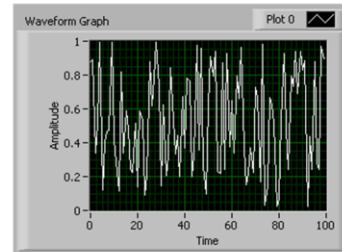
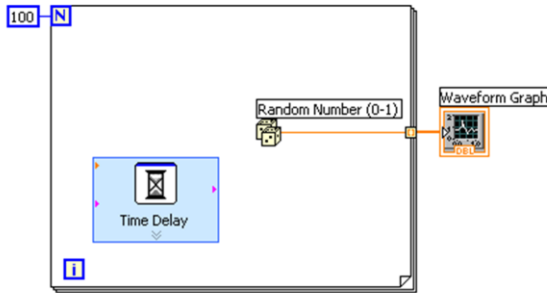
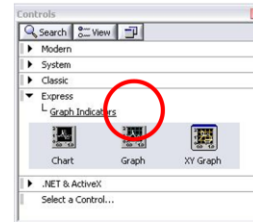
Pode-se mudar o valor de máximo e mínimo dos eixos x e y clicando duas vezes no valor com a ferramenta de edição de texto e digitando novos valores. Da mesma forma, pode-se mudar a legenda dos eixos. Por fim, pode-se também clicar na legenda do gráfico e mudar o estilo, o formato e a cor das linhas exibidas no gráfico.

Graphs – Exibe vários pontos de dados de uma única vez

Waveform graph – Indicador numérico especial que exibe um array de dados

- O Graph atualiza após todos os pontos coletados
- Pode ser usado em um loop se o VI coletar buffers de dados

Controls » Express » Graph Indicators » Graph



ni.com

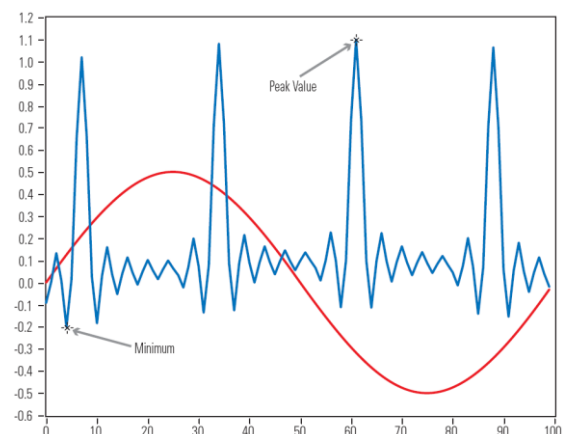
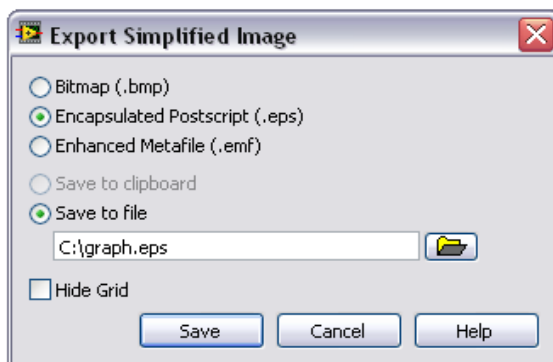
54



Graphs são indicadores muito poderosos do LabVIEW. Pode-se usar as ferramentas que permitem muitos níveis de personalização para exibir uma grande quantidade de informação de maneira concisa.

A página de propriedades do objeto graph exibe as possíveis configurações para os tipos de linhas a serem traçadas, opções de escala e cursor e outros recursos. Para abrir a página de propriedades, clique com o botão direito sobre o graph no painel frontal e escolha **Properties**.

Pode-se também criar material de documentação técnica usando graphs utilizando a função “export simplified image”. Para isso, clique com o botão direito no graph, selecione **Data Operations»Export Simplified Image...**



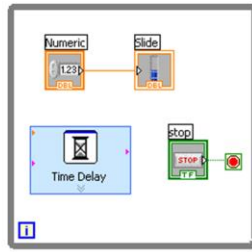
Como eu temporizo um loop?

1. Loop Time Delay

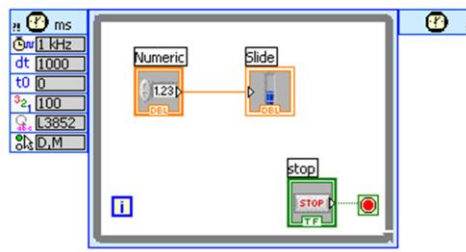
- Configura o VI expresso Time Delay, o intervalo de espera em segundos para cada iteração do loop (funciona em for ou while loops).

2. Timed Loops

- Configura um loop while especial para o dt desejado.



Time Delay



Timed Loop

ni.com

55



A principal idéia de temporizar um loop com as funções ou estruturas abaixo é gerar tempo livre para o processador poder executar outras tarefas.

Time Delay

O VI Expresso Time Delay atrasa a execução pelo tempo especificado de segundos. Seguindo as regras da programação por fluxo de dados, o loop While não irá passar os dados para a função seguinte até que todas as tarefas dentro do loop tenham sido executadas e assim atrasando cada iteração do loop.

Timed Loop

Executa cada iteração do loop no período especificado. Use o Timed Loop quando for desenvolver VIs que necessitem de múltiplas taxas de temporização, precisão na temporização, feedback sobre a execução no loop, características de temporização que mudam dinamicamente ou vários níveis de prioridades de execução.

Dê um duplo clique ou clique com o botão direito no Input Node e selecione

Configure Input Node do menu de atalho para visualizar a caixa de diálogo

Configure Timed Loop, onde poderá configurá-lo. Os valores inseridos na caixa de diálogo **Loop Configuration** aparecem como opções no Input Node.



Wait Until Next ms Multiple

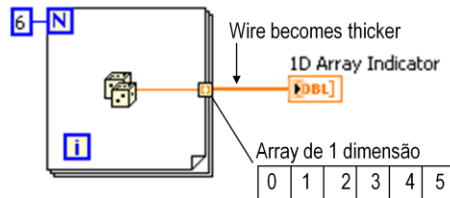
Esta função espera até o valor do tempo no microprocessador em milisegundos seja um múltiplo especificado do valor da entrada **millisecond multiple**, o que auxilia no sincronismo das atividades. Pode-se utilizar função em um loop para controlar a taxa de execução deste. No entanto, é possível que o período da primeira execução do loop seja menor. Esta função faz chamadas assíncronas, mas os nós funcionam de forma síncrona. **Portanto, ele não** completa a execução até que o tempo especificado tenha terminado. Esta função pode ser encontrada no caminho

Functions»Programming»Timing» Wait Until Next ms Multiple

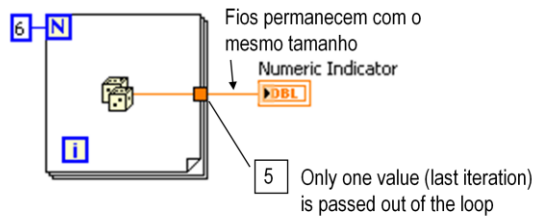
Construindo arrays com loops (Auto-indexação)

- Loops podem acumular arrays em suas bordas com auto-indexação
- For loops tem a auto-indexação habilitada por definição
- Por definição, a saída do while loop apresenta apenas o valor final
- Clique com o botão direito sob o túnel para habilitar/desabilitar a auto-indexação

Auto-indexação habilitada



Auto-indexação desabilitada



Loop for e while podem ser indexados e acumular arrays em suas bordas. Isto é conhecido como auto-indexação (Auto-Indexing)

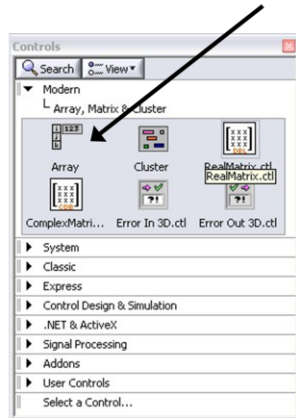
- O ponto de indexação na borda é conhecido como túnel.
- O loop for tem auto-indexação habilitada por padrão.
- O loop while tem auto-indexação desabilitado por padrão.

Exemplos:

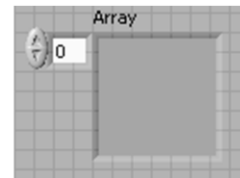
- Habilite a auto-indexação para coletar valores dentro do loop para construir um array. Todos os valores são colocados em um array quando o loop for finalizado.
- Desabilite a auto-indexação se estiver interessado somente no último resultado das iterações.

Criando um array (Passo 1 de 2)

Da subpaleta **Controls » Modern » Array, Matrix, and Cluster** subpalette, selecione o ícone **Array**.



Drop it on the front panel.



Para criar um array de controles ou indicadores como mostrado acima, selecione uma estrutura array na paleta de controles, no caminho **Controls»Modern»Array, Matrix, and Cluster** e coloque-o no painel frontal. Depois, selecione um controle ou indicador válido e arraste para dentro da estrutura array. Se você tentar arrastar um controle ou indicador inválido como um gráfico XY, para dentro da estrutura, o LabVIEW não permitirá essa ação.

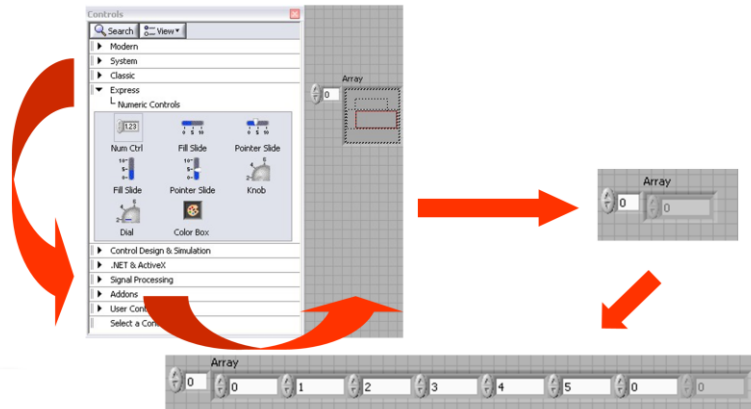
Você deve inserir um objeto dentro da estrutura do array antes de usar o array no diagrama de blocos. Caso contrário, o terminal do array ficará preto e os colchetes estarão vazios.

Arrays devem ter somente controles ou indicadores de um único tipo de dado. Não é possível combinar por exemplo um controle e um indicador numérico em um mesmo array.

Um array pode ter APENAS um tipo de dado em uma única estrutura. Não é possível combinar tipos de dados em uma única estrutura array. (para isso, use a estrutura de dado cluster).

Criando um array (Passo 2 de 2)

1. Coloque uma estrutura array.
2. Insira o tipo de dado dentro da estrutura
(Por exemplo: controle numérico).



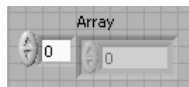
ni.com

58

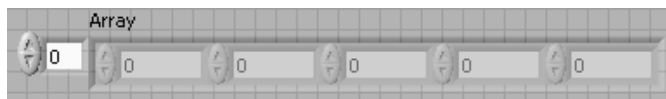


Para acrescentar dimensões a um array, clique com o botão direito no index display e selecione **Add Dimension** do menu de atalho. Pode-se redimensionar (ferramenta de posicionamento) ou selecionar a posição desejada (ferramenta de operação) do array e assim mostrar os valores desejados. O valor do index display sempre mostra o valor da posição mais acima e mais à esquerda.

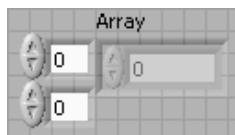
Array de uma dimensão - um único elemento visualizado:



Array de uma dimensão - múltiplos elementos visualizados:



Array de duas dimensões - um único elemento visualizado:

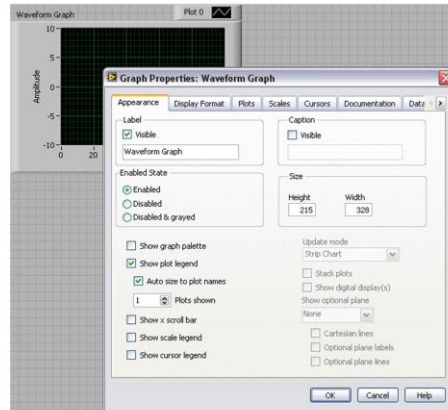


Array de duas dimensões - múltiplos elementos visualizados



Propriedades de controles e indicadores

- Propriedades são características ou qualidades de um objeto
- Propriedades podem ser encontradas clicando com o botão direito no controle ou indicador
 - Propriedades:
 - Tamanho
 - Cor
 - Estilo do plot
 - Cor do plot
 - Recursos:
 - Cursores
 - Escalas



ni.com

59



Propriedades são todas as qualidades dos objetos do painel frontal. Através das propriedades, é possível configurar ou ler características como cores, formatação de dados e precisão, visibilidade, texto descritivo, tamanho e lugar no painel frontal e etc.

Para abrir as propriedades do objeto, clique com o botão direito sobre ele. Para uma visualização através de caixas de diálogos, clique sobre “Properties”



Exercício 6 – Análise Manual (Opção A, B, & C)

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

1. Abra Exercício 3 – Simulated.vi.
2. Salve o VI como “Exercício 6 – 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 o Waveform Chart por um Waveform Graph. Clique com o botão direito do mouse na tabela e selecione **Replace»Modern»Graph»Waveform Graph**.
6. Torne a legenda do cursor visível 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. Dê um duplo clique 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 no Waveform Graph. Se você não conseguir visualizar o sinal, você pode precisar desligar a escala 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 sejam vistos alguns períodos 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-os pelo gráfico e suas coordenadas serão mostradas na legenda.

Cursors:	X	Y
Cursor 0	0.0315	0.090909
Cursor 1	0.06165	-0.36969

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

- Integra scripts existentes com LabVIEW para um rápido desenvolvimento
- Use um ambiente interativo, fácil de usar e de aprendizado prático
- Desenvolva algoritmos, explore conceitos matemáticos e analise resultados usando um único ambiente
- Liberdade para escolher a sintaxe mais efetiva, tanto gráfica quanto textual em um único VI

ni.com

MATLAB® é uma marca registrada da The MathWorks, Inc.

61

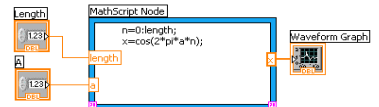


Visão Geral

Com o LabVIEW 2010 da National Instruments, é possível ter a liberdade de escolher a sintaxe mais eficaz para computação técnica, se estiver desenvolvendo algoritmos, explorando conceitos DSP ou analisando resultados. Pode-se instrumentar seus scripts e desenvolver algoritmos no diagrama de blocos interagindo com ferramentas matemáticas de terceiros como o software MathWorks Inc. MATLAB, Wolfram Mathematica, Maplesoft Maple, Mathsoft Mathcad, ITT IDL e Xmath. O uso destas ferramentas com o LabVIEW pode ser feito de várias formas, dependendo do programa:

Nó matemático textual nativo LabVIEW:

LabVIEW MathScript* node e Formula node



Comunicação com outros software de terceiros através de nós do LabVIEW:

Xmath node, MATLAB script node, Maple** node e IDL** node

Comunicação com software de terceiros através do VI Server:

Vis Mathematica** e Vis Mathcad**

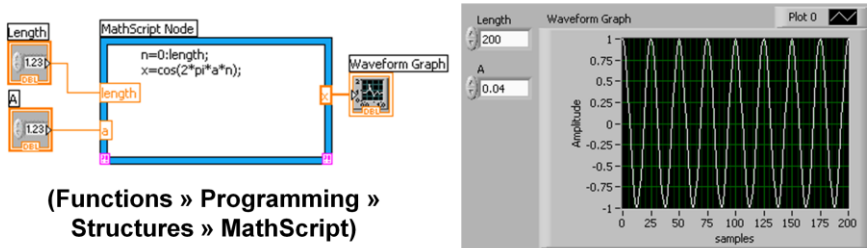
No LabVIEW, pode-se combinar a programação gráfica e intuitiva do LabVIEW baseada em fluxo de dados com o LabVIEW MathScript, uma linguagem matemática textual normalmente é compatível com a linguagem de arquivos .m

*O MathScript Node e o Formula Node fazem parte do Módulo MathScript RT.

**Deve ser instalado um toolkit específico no LabVIEW para ferramentas matemáticas

Matemática com o LabVIEW MathScript Node

- Implementa equações e algoritmos textualmente
- Variáveis de entrada e saída criadas na borda
- Geralmente compatível com a linguagem de arquivo .m
- Termine as declarações com um ponto e vírgula para desabilitar a saída imediata



Prototipe suas equações utilizando a interativa LabVIEW **MathScript Window**.

ni.com

62



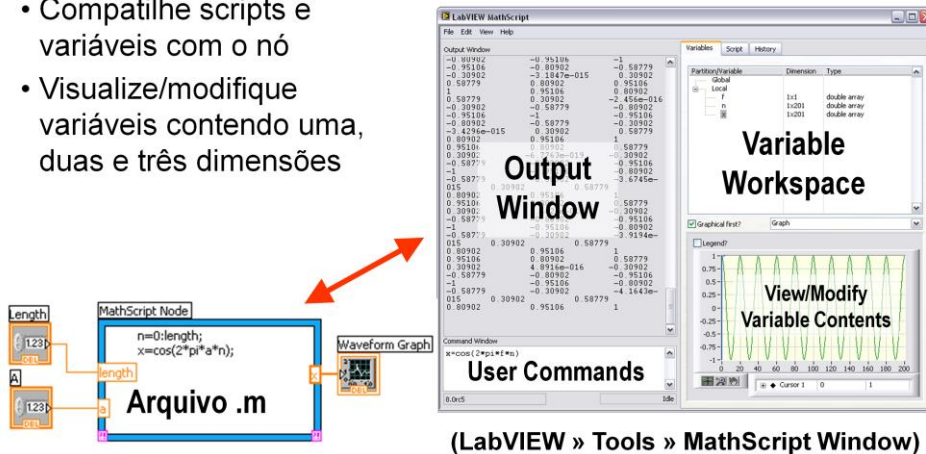
O LabVIEW MathScript Node aumenta o potencial do LabVIEW ao acrescentar uma linguagem baseada em texto para implementação de algoritmos matemáticos no ambiente de programação gráfica do LabVIEW. Pode-se abrir e usar arquivos com scripts .m que foram escritos e salvos na LabVIEW MathScript window no LabVIEW MathScript Node. Os arquivos com scripts .m criados em outros programas matemáticos normalmente podem também ser executados bem. Com o LabVIEW MathScript é possível selecionar a sintaxe que mais se adequa para resolver problemas. Equações podem ser instrumentadas com o LabVIEW MathScript Node para serem parâmetros, simulação e implementação em uma aplicação final.

O LabVIEW MathScript Node

- Localizado na subpaleta **Programming»Structures**.
- Caixa redimensionável para inserir textos computacionais diretamente no diagrama de blocos.
- Para adicionar variáveis, clique com o botão direito e selecione **Add Input** ou **Add Output**.
- Nomeie variáveis como elas são usadas na fórmula. (Nomes são diferenciáveis por letras maiúsculas e minúsculas).
- Pode-se mudar o tipo de dado da saída clicando com o botão direito no nó de entrada ou saída.
- Termine as declarações com ponto e vírgula para evitar a execução imediata.
- Clique com o botão direito no nó para importar e exportar arquivos .

A interativa LabVIEW MathScript Window

- Desenvolva e teste algoritmos rapidamente
- Compartilhe scripts e variáveis com o nó
- Visualize/modifique variáveis contendo uma, duas e três dimensões



(LabVIEW » Tools » MathScript Window)

ni.com

63



O LabVIEW MathScript window fornece um ambiente interativo onde pode-se prototipar equações e realizar cálculos. O MathScript window e o MathScript Node compartilham uma sintaxe comum e variáveis globais, tornando fácil a migração do ambiente de prototipagem para a implementação. A janela data preview fornece uma forma conveniente para visualizar os dados das variáveis como números, gráficos ou áudio (com suporte à placa de som).

Ajuda para o LabVIEW MathScript

Pode-se acessar a ajuda para o ambiente usando o ambiente interativo do LabVIEW MathScript window. Digite **Help** na janela de comandos para uma introdução à ajuda do MathScript. Digitando **Help** seguido de uma **função** exibirá a ajuda específica para essa função.

Recursos do MathScript window:

- Prototipe equações e fórmulas através da janela Command.
- Acesse arquivos de ajuda para as funções digitando **Help <function>** na janela Command.
- Selecione uma variável para exibir seus dados na janela Preview Pane e ouvir os resultados.
- Escreva, salve, armazene, e execute arquivos .m usando o tab Script.
- Compartilhe dados entre o MathScript Node e o MathScript window usando variáveis globais.
- Aproveite os recursos gráficos e de exportação de imagens.



Exercício 7 – Usando o MathScript (Tracks A, B, and 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 6 – 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. 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.
7. Ligue a saída “Sine” do Simulate Signal VI à entrada “Dynamic Data” do Convert from Dynamic Data VI.
8. Ligue a saída “Array” do Convert from Dynamic Data VI ao nó “In” do MathScript Node.
9. 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;”.
10. Retorne ao Painel Frontal e aumente a frequência para um valor entre 50 e 100. Execute o VI.
11. Abra a Janela Interativa do MathScript(**Tools»MathScript Window...**).
12. Na Janela Interativa do MathScript, a janela de comandos pode ser usada para digitar os comandos desejados. Digite “global In” e aperte “Enter”. Isso permitirá que você veja os dados que passam pela variável “In” no MathScript Node.

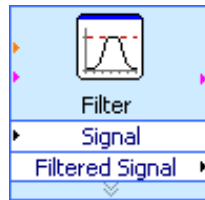


Exercício 8 – Aplique o que foi aprendido (Opção A, B, & C)

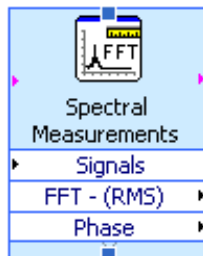
Neste exercício, crie um VI com o que aprendeu até aqui.

Desenvolva um VI que faz o seguinte:

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



3. Faça uma transformada rápida de Fourier (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 principal 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, acenda um LED. Se você estiver usando o USB-600X, ascenda o LED de seu hardware usando o DAQ Assistant. Você precisará inverter a linha digital para acender 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. Caso tenha dificuldades, abra a solução ou veja a mesma no final deste manual.

(Fim do exercício)

Parte IV – Tópicos avançados de fluxo de dados (Opcional)

Tipos de dados adicionais

- Cluster
- Waveform

Construções de fluxo de dados

- Property e Invoke nodes
- Modularidade e subVIs
- Máquinas de estados
- Variáveis locais
- Produtor/ Consumidor

Introdução a Clusters

- Estrutura de dados que agrupam dados
- Dados podem ser de tipos diferentes
- Análogo a *struct* em ANSI C
- Elementos devem ser todos controles ou todos indicadores
- Analogia com fios empacotados em um cabo
- **Ordem é importante**



ni.com

69



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

Os componentes do cluster podem ser de tipos diferentes de dados

Exemplos:

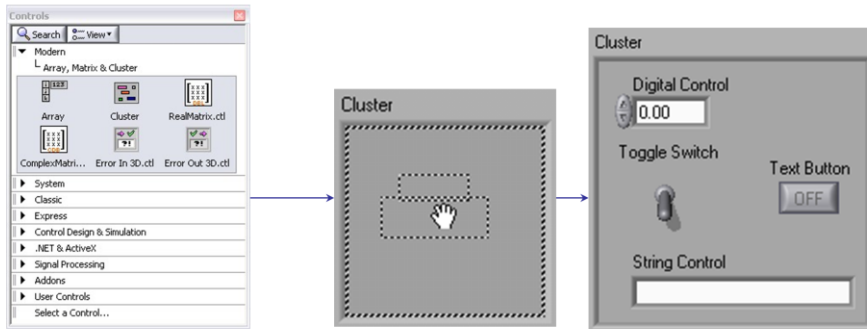
- Informação de erros — Um tipo de dado onde é agrupado uma informação booleana de erro, um código numérico de erro e uma string indicando a fonte do erro para especificar o erro.
- Informação de usuário — Um tipo de dado onde é agrupado uma string com o nome do usuário e um número de identificação definindo seu código de segurança.

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

Criando um Cluster

1. Selecione uma estrutura **Cluster**.
2. Insira objetos dentro da estrutura.

Controls » Modern » Array, Matrix & Cluster



ni.com

70



Crie um objeto cluster no painel frontal selecionado **Cluster** da paleta **Controls»Modern»Array, Matrix & Cluster**.

- Esta opção traz a estrutura do cluster (parecida com a estrutura do array).
- Você pode dimensionar a estrutura do cluster ao colocá-la no painel frontal.
- Abra a paleta de controles e acrescente objetos na estrutura.

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 também criar um cluster de constantes no diagrama de blocos usando o **Cluster Constant** localizado na Paleta de Funções, no caminho **Functions»Programming»Cluster, Class, & Variant**.

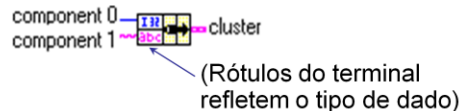
- Isto traz uma estrutura de cluster vazia.
- Você pode dimensionar o cluster ao colocá-lo no diagrama de blocos.
- Coloque as constantes dentro da estrutura.

Nota: Você não pode colocar terminais do painel frontal em um cluster de constantes e nem colocar constantes “especiais”, como constantes de Tab ou empty strings dentro do cluster de constantes.

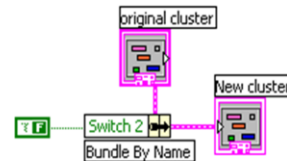
Funções de Cluster

- Na subpaleta **Cluster & Variant** da paleta **Programming**
- Podem também ser acessados clicando com o botão direito no terminal do cluster

Bundle



Bundle By Name



Os termos Bundle e Cluster são intimamente 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 objetos dados na ordem específica.

Função Bundle by Name - Atualiza os valores dos objetos selecionado no cluster (o objeto deve ter seu próprio label).

Função Unbundle – Separa um cluster em cada um dos elementos de forma individual pelo tipo de dado.

Unbundle by Name - Retorna os valores dos elementos do cluster selecionado.

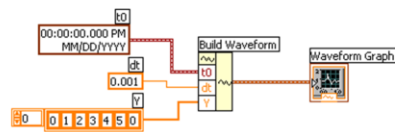
Nota: Você precisa ter um cluster existente ligado no terminal input cluster da função Bundle by Name.

Usando Arrays e Clusters com Graphs

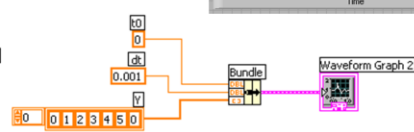
O tipo de dado waveform é composto de três partes de dados:

- t_0 = Tempo de início
- Δt = Tempo entre as amostras
- Y = Array de Y magnitudes

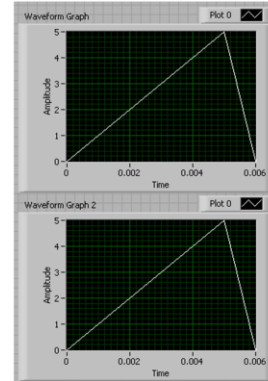
Pode-se criar um cluster de waveform de duas formas:



Build Waveform (tempo absoluto)



Cluster (tempo relativo)



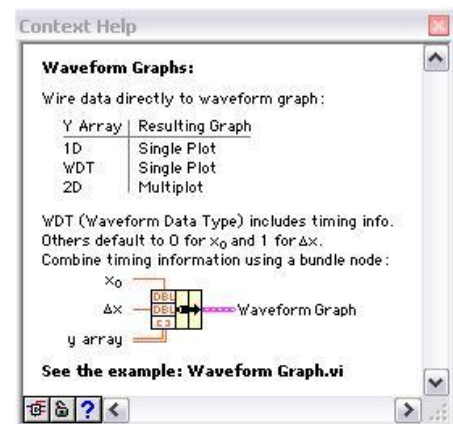
O tipo de dado waveform traz os dados, o tempo de início e o Δt da forma de onda. Pode-se criar formatos de onda usando a função Build Waveform. Vários VIs e funções usados para adquirir ou analisar formas de onda aceitam e retornam este tipo de dado quando trabalham de forma padrão. Quando você conecta um tipo de dado waveform a um waveform graph ou chart, as ondas são plotadas automaticamente baseando-se nos dados, tempo inicial e Δx da forma de onda. Quando conecta um array contendo o tipo de dado waveform a um gráfico ou tabela.

Build Waveform

Cria ou modifica o tipo de dado waveform com o tempo inicial representado com um tempo absoluto em timestamps. Timestamps contém o horário e a data real e são muito úteis para gravação de dados.

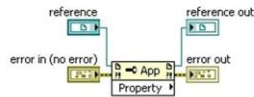
Bundle

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

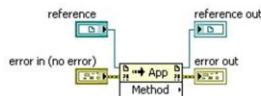


Property e Invoke Nodes

- Crie Property e Invoke nodes para um elemento no VI clicando com o botão direito nele e selecionando **Create » Property Node**.
- Use o Property Node para obter e definir a propriedade do objeto



- Chame métodos do objeto com o Invoke Node.



Waveform Chart



Waveform Chart Property Node



Value

Waveform Chart Invoke Node



FitControlToPane

O Property node e o Invoke node fornecem uma interface para controlar objetos do LabVIEW. O objeto pode ser um objeto ActiveX ou uma referência ao VI através do VI Server.

Pode-se criar Property e Invoke usando dois métodos. O primeiro é clicar com o botão direito no objeto e selecionar **Create»Property Node**. Este é bom para definir propriedades e acessar método dos elementos do VI dentro do VI. O segundo método é colocar um Property Node ou Invoke Node do menu **Programming» Application Control**. Este é melhor para acessar as propriedades e métodos de aplicações e VIs externos porque esses nós podem ter uma referência de entrada. É possível expandir esses nós para ler ou escrever múltiplas propriedades de uma única vez e utilizar vários nós com diferentes métodos.

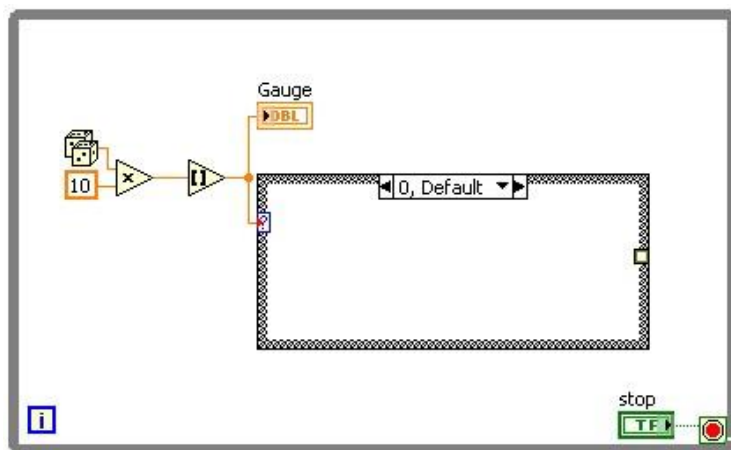
Pode-se usar o Property Node para obter ou definir propriedades dos objetos. Por exemplo, se for necessário reiniciar no programa um controle no painel frontal, pode-se criar um property node para o controle que acessa a propriedade “Value” (clique com o botão direito e selecione **Create»Property Node»Value**). Com isso, será possível ler o valor deste controle ou mudar esta propriedade modificando o Property Node para o modo Write.

Use o Invoke Node para realizar métodos (ações) do objeto. Estes métodos podem incluir informações sobre uma aplicação ou controle ActiveX, tais como, número da versão ou ações que aquele objeto pode realizar.



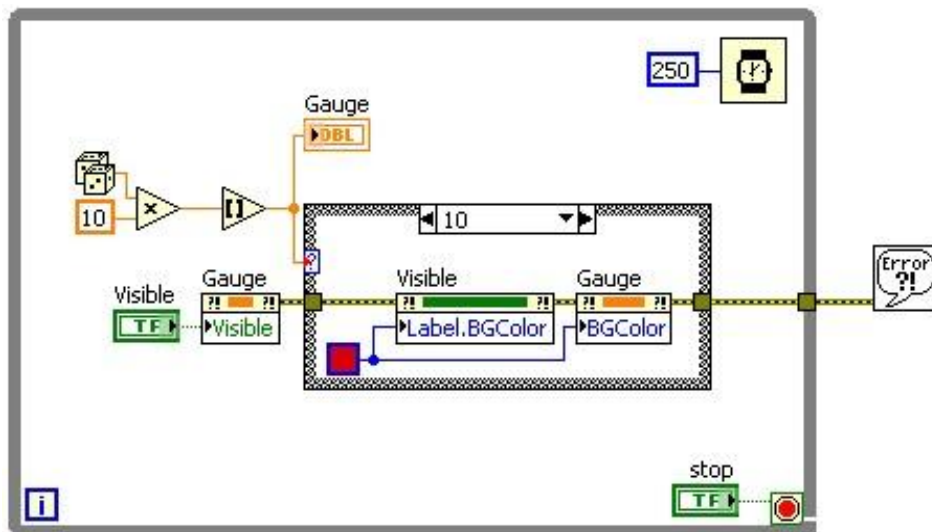
Exercício 9 – Entendendo Property Nodes usando cores (Opção A, B e C)

1. Abra um novo VI em branco.
2. Adicione um controle a gauge ao painel frontal. Clique com o botão direito para abrir a paleta de controles e então navegue para **Modern»Numeric»Gauge**.
3. Mude para o diagrama de blocos e insira um loop while ao redor do terminal Gauge selecionando **Programming»Structures»While Loop**.
4. Clique com o botão direito no terminal condicional do loop while e selecione **Create»Control**. Isto cria o botão stop.
5. Dentro do loop while, insira a função *random number (0-1)*, para gerar um número aleatório. O caminho é **Programming»Numeric»Random Number (0-1)**.
6. À direita da função *random number (0-1)*, insira uma função *Multiply* pelo caminho **Programming»Numeric»Multiply**. Conecte a função *random number* à função *Multiply*.
7. Clique com o botão direito no outro terminal da função *Multiply* e selecione **Create»Constant**. Insira o valor 10 para ela.
8. À direita da função *Multiply*, insira a função *Round To Nearest* no caminho **Programming»Numeric»Round To Nearest**.
9. Conecte a saída da função *Multiply* à entrada da função *Round To Nearest*.
10. Conecte a saída da função *Round To Nearest* ao indicador Gauge.
11. Insira uma estrutura case abaixo do indicador Gauge e conecte o fio da saída da função *Round To Nearest* ao terminal seletor da estrutura case. O diagrama de blocos deve ser similar com a imagem abaixo.



12. Mude para Case 1 na estrutura case. Clique com o botão direito em sua borda e selecione **Delete This Case**.
13. Crie um property node com a propriedade Visible. Clique com o botão direito no indicador Gauge e selecione **Create»Property Node»Visible**. Insira este property node à esquerda da estrutura case.

26. Após finalizar, o diagrama de blocos deverá ser similar à imagem abaixo.

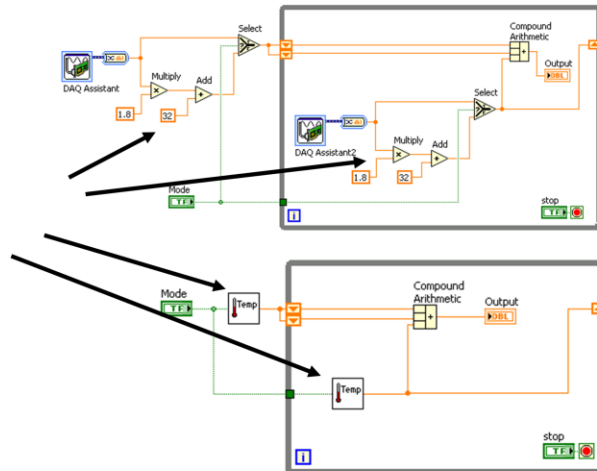


27. Salve o VI e o execute. Note que quando aperta o botão Visible, o Gauge aparece e desaparece. Note também como a cor do Gauge muda de acordo com o número aleatório criado.

(Fim do exercício)

Modularidade no LabVIEW – SubVIs

Converte repetidas
funções e VIs em um
único VI



ni.com

77

NATIONAL
INSTRUMENTS

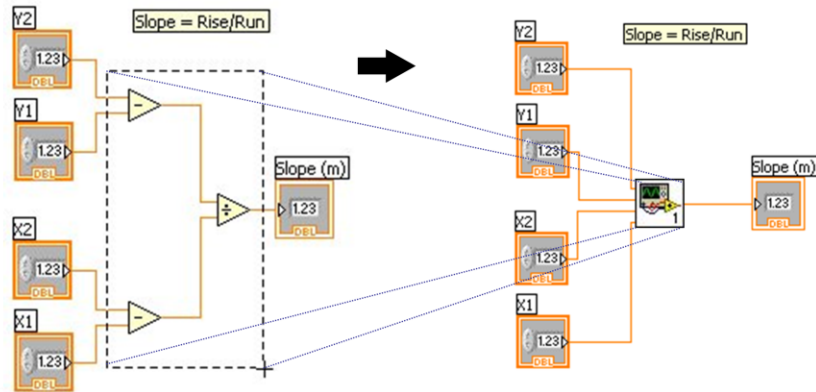
Modularidade define o grau em que o VI é composto por componentes discretos tal que uma mudança em um componente tenha impacto mínimo nos demais componentes. No LabVIEW estes componentes separados são chamados de subVIs. Criar um subVI em seu código aumenta a legibilidade e a capacidade de reutilização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

Na imagem superior, o código repetido permite que o usuário escolha entre escalas de temperatura. Como esta parte do código é idêntica para ambos os casos, foi criado um subVI deste. Isso tornará o código mais legível por ser menos poluído e permitirá o reaproveitamento do código facilmente. Como pode ser visto, o código ficou mais organizado e tem-se a mesma finalidade. Pode-se reutilizar facilmente esta parte do código, de seleção de escalas de temperatura, em outras aplicações utilizando-o como um subVI.

Pode-se transformar qualquer código LabVIEW ou parte deste em um subVI que pode ser usado em outros códigos do LabVIEW.

Crie SubVI

- Selecione a área a ser convertida em um subVI.
- Selecione **Edit » Create SubVI** do menu Edit.



ni.com

78



Criando um SubVI

Um nó de subVI corresponde a uma chamada de subrotina em linguagens de programação baseadas em texto. 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 ou texto **Select a VI** na paleta de funções, navegue e clique duas vezes em um VI e coloque este no diagrama de blocos para criar uma chamada de subVI naquele VI.

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

Criando SubVIs a partir de seções de um VI

Transforme uma seção de seu VI em um subVI usando a ferramenta de posição (**Positioning tool**) para selecionar a seção de seu diagrama de blocos que deseja reutilizar 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, configurando de forma automática o connector pane baseado no número de controles e indicadores selecionados e conecta o subVI aos fios existentes.

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

Funções e SubVIs no LabVIEW operam como funções em outras linguagens

Pseudo código da função

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

Chamando o programa com o pseudo código

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

Diagrama de blocos - SubVI

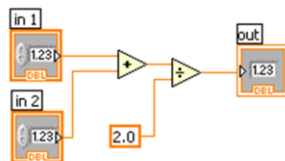
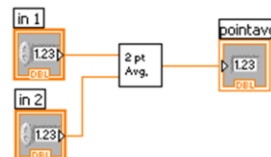


Diagrama de blocos – VI Chamador



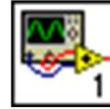
Um nó do subVI corresponde uma chamada de subrotina em uma linguagem de programação baseada em texto. O nó não é o subVI propriamente dito, assim como uma chamada de subrotina em um programa baseado em texto não é a subrotina propriamente dita. O diagrama de blocos que contém vários nós idênticos de subVIs chama o mesmo subVI diversas vezes.

A abordagem modular torna mais fácil depurar e manter as aplicações.

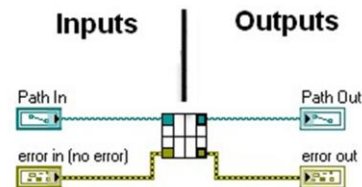
A funcionalidade do subVI não importa para este exemplo. O ponto importante é mostrar que há duas entradas numéricas e uma única saída numérica.

Connector Pane e Ícone

- Use este connector pane layout como um padrão



- Terminais superiores são normalmente reservados para caminhos de arquivos e referências, como uma referência de arquivo
- Terminais inferiores são normalmente reservados para clusters de erros



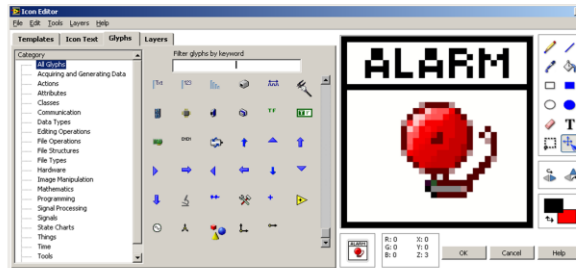
Com o connector pane e o Icon Editor, pode-se definir os dados que serão transferidos para dentro e para fora do subVI e sua aparência no código principal do LabVIEW. Cada VI exibe um ícone no canto superior direito do painel frontal e diagrama de blocos. Depois de criar seu VI, crie um ícone e um connector pane para melhorar o uso deste VI como subVI.

O ícone e o connector pane corresponde ao protótipo da função em linguagens de programação baseadas em texto. Há diversas opções para o connector pane, mas alguns padrões estão exemplificados acima. Em geral, sempre reserve os terminais superiores para referências e os inferiores para usar os clusters de erro.

Para definir um connector pane, clique com o botão direito no ícone (canto superior direito do painel frontal) e selecione **Show Connector** no menu de atalho. Cada retângulo no connector pane representa um terminal. Use os terminais para atribuir entradas e saídas. Selecione padrões diferentes clicando com o botão direito no connector pane e selecionando **Patterns** no menu de atalhos.

Icon Editor – Crie um ícone

- Crie ícones personalizados clicando com o botão direito no ícone (canto superior direito da tela do painel frontal ou diagrama de blocos) e selecionando **Edit Icon** ou clicando duas vezes no ícone
- Pode-se também arrastar imagens de qualquer lugar de seu sistema de arquivos e utilizá-lo como ícone



ni.com

81



Um ícone é uma representação gráfica do VI. Se você usar um VI como subVI, o ícone identifica o subVI no diagrama de blocos do VI principal. O Icon Editor é uma funcionalidade integrada ao LabVIEW que permite aos usuários personalizar totalmente a aparência de seus subVIs. Isso permitirá que seus programadores diferenciem visualmente seus subVIs e aumentem a usabilidade do subVI em grandes códigos.

Depois de ter definido o connector pane e de personalizar seu ícone, o subVI está pronto para ser usado em outros códigos do LabVIEW. Há duas formas de fazer isso:

Para colocar um subVI no Diagrama de Blocos

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

Para colocar um VI aberto no diagrama de blocos de outro VI

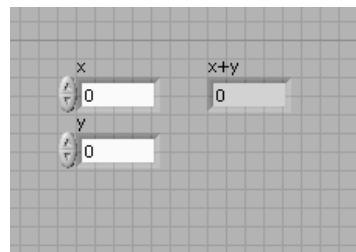
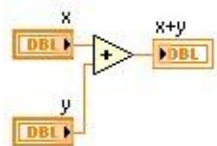
1. Use a ferramenta de posição (**Positioning tool**) para clicar no ícone do VI que você deseja usar como subVI;
2. Arraste o ícone para o diagrama de blocos do outro VI.



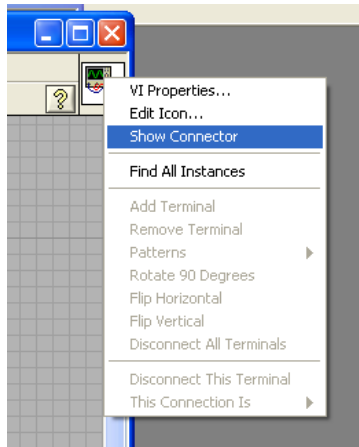
Exercício 10 – Criando um SubVI

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 ficar parecidos com as imagens abaixo.



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

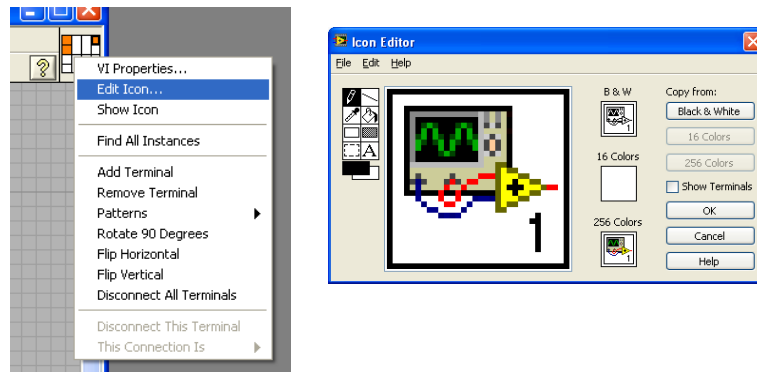


5. Atribua terminais aos dois controles e ao indicador clicando primeiro no terminal e depois no controle/indicador desejado

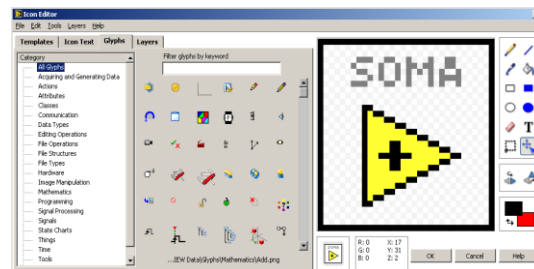
Nota: Por convenção, são usados os controles como dados de entrada no lado esquerdo e indicadores como saída no lado direito do ícone.



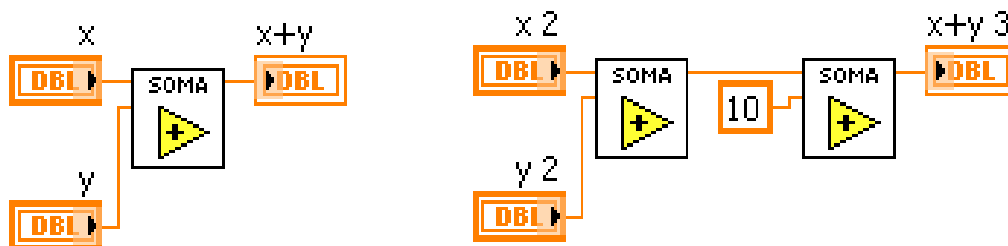
6. Clique com o botão direito do mouse no painel de conexões e selecione **Edit Ícone....** Isto abrirá o editor.



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



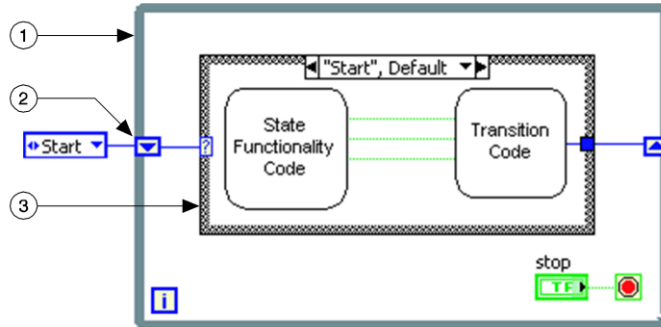
8. Salve o SubVI. Ele agora pode ser usado como qualquer outro VI para realizar qualquer função, neste caso a soma de dois números.



(Fim do exercício)

Máquina de estados

- While Loop
- Estrutura Case
- Shift Register



ni.com

84

NATIONAL
INSTRUMENTS

Pode-se usar o padrão de projeto máquina 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 funções de transições que direcionam para o próximo estado.

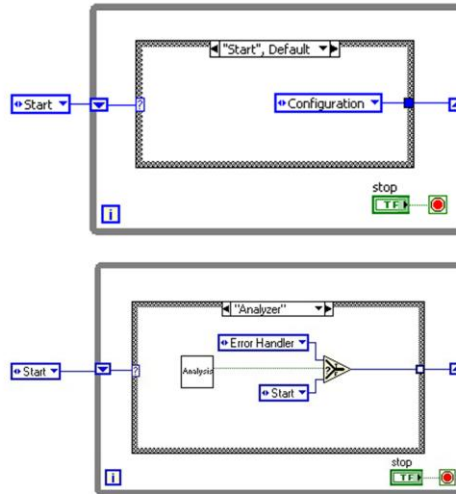
Cada estado pode levar a um ou mais estados ou encerra o fluxo do processo.

Uma aplicação comum para máquinas de estados é para criar interfaces de usuário. Em uma interface de usuário, diferentes ações do usuário encaminham o programa para diferentes segmentos do processo. Cada segmento do processo atua como um estado diferente.

Outra aplicação comum do padrão de projeto máquina de estados são processos de teste. Para um processo de teste, um estado representa um segmento do processo. Dependendo do resultado de cada teste, um estado diferente pode ser chamado.

Transições na máquina de estados

- Existem diversas técnicas de programação para realizar transições entre estados no LabVIEW usando máquinas de estados
- Transição padrão significa que após um estado ser realizado, o próximo estado sempre será constante
- Transições entre dois potenciais estados podem ser gerenciadas por uma função select

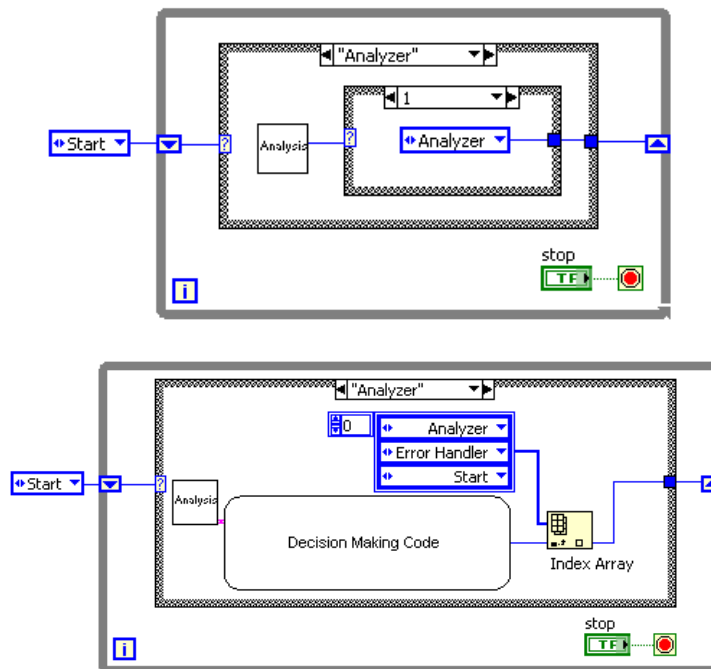


ni.com

85

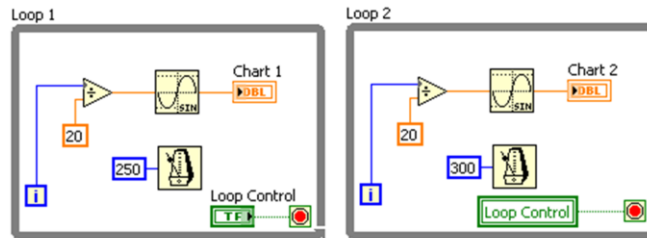
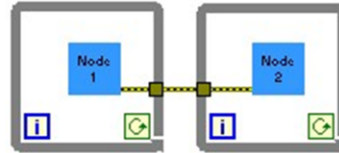
NATIONAL INSTRUMENTS

Se um estado pode ter a transição para vários estados potenciais, pode-se usar uma estrutura case. Outra abordagem é usar um array de estados futuros 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 não é possível
- A execução do loop à direita inicia-se somente após o loop à esquerda ter sido finalizado.
- Variáveis são necessárias quando a comunicação com fios não apresenta o comportamento desejado



ni.com

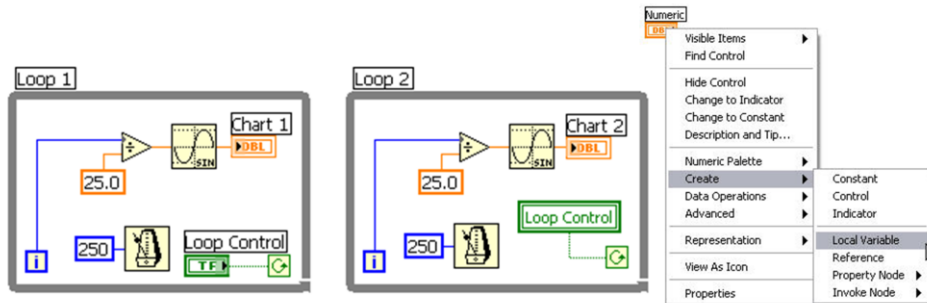
86



É impossível comunicar entre loops paralelos usando fluxo de dados. Os dados não podem entrar ou sair de uma estrutura enquanto ela ainda estiver em execução. Variáveis são elementos do diagrama de blocos que permitem o acesso ou armazenamento dos dados em outra localização. Variáveis locais guardam os dados em controles e indicadores do painel frontal. Variáveis permitem que você “quebre” o fluxo de dados passando dados de um lugar para outro sem ter de usar um fio.

Variáveis locais

- Variáveis locais permitem que o dado seja passado entre loops paralelos.
- Um único controle ou indicador pode ser lido ou escrito em mais de um ponto do programa
 - Variáveis locais quebram o paradigma do fluxo de dados e devem ser usadas com moderação



ni.com

87



Variáveis locais são encontradas na subpaleta **Programming » Structures** da paleta **Functions**.

Quando uma variável local é inserida no diagrama de blocos, ela não é alocada a nenhum objeto do painel frontal e permanece com o sinal de ? até que seja clicada, o que permite selecionar o objeto em uma lista que contém todos os objetos que podem ser alocados para esta variável local.

Pode-se também clicar com o botão direito sobre qualquer controle ou indicador e selecionar **Create » Local Variable**. Isto cria uma variável local alocada diretamente ao controle ou indicador de onde foi criada

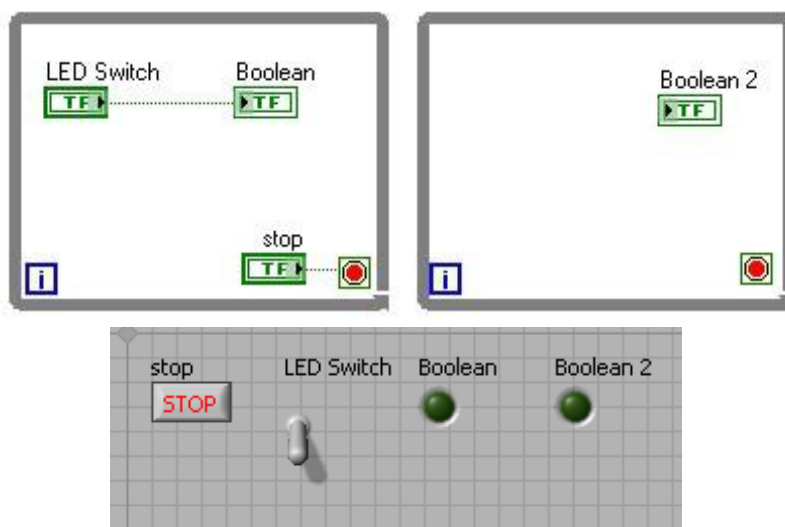
Em seguida, deve-se definir o comportamento do objeto (escrita ou leitura). Clique com o botão direito na variável local e selecione **Change To Read** ou **Change to Write** no menu de atalho.



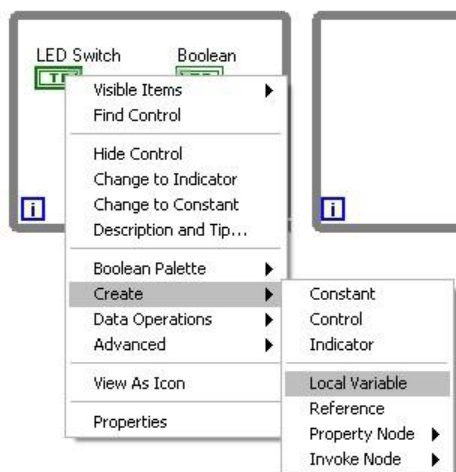
Exercício 11 – Criando variáveis locais

Crie um VI que comunica entre dois loops paralelos usando uma variável local.

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. Organize código da seguinte maneira:



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

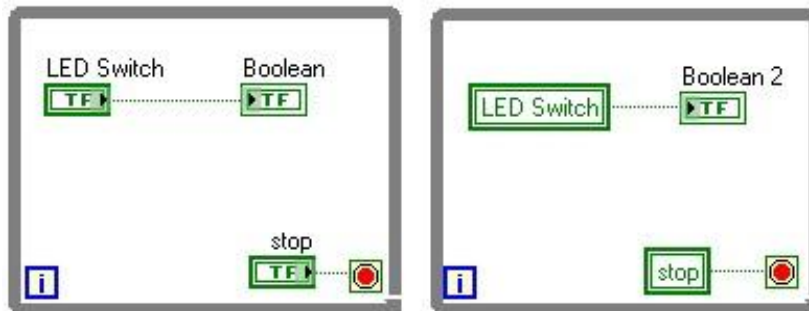


6. Coloque a variável

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, serão escritos nela.



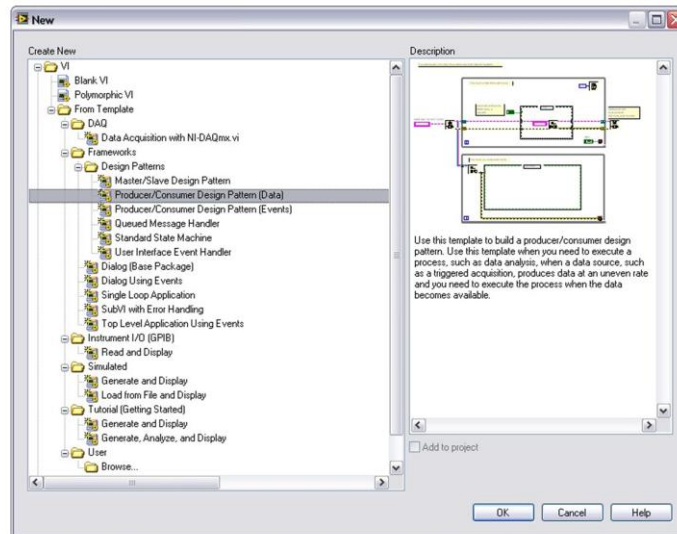
8. Repita o processo com o botão Stop.
9. Clique com o botão direito do mouse no botão Stop no painel frontal e mude a ação mecânica para Switch When Released. Variáveis locais não guardam dados booleanos com a configuração latch. O código completo deve parecer com o seguinte:



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

(Fim do exercício)

Padrão de projeto Produtor/Consumidor



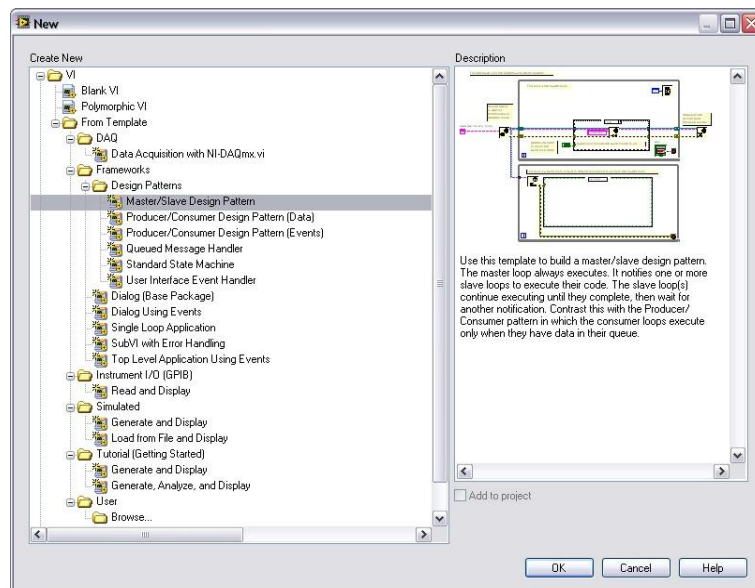
ni.com

90



Além das variáveis, há outros métodos para transferência de dados entre loops paralelos. Isto é possível usando as funções Notifier (Notificação) e Queue (Fila). Pode-se usar a função Notifier para implementar o padrão de projeto Mestre/Escravo e Queues para implementar o padrão de projeto Produtor/Consumidor. As duas funções permitem aos programadores de LabVIEW compartilhar dados entre loops.

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



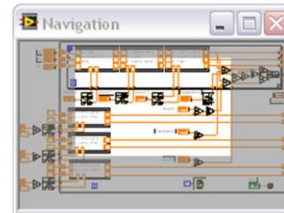
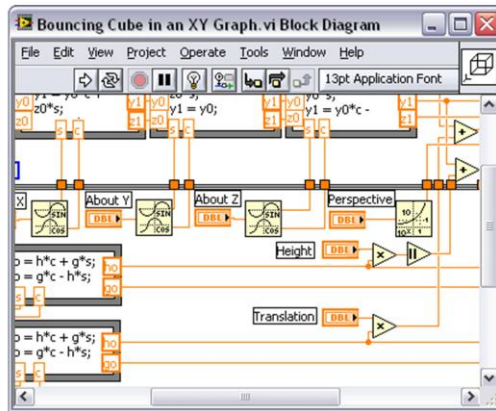
V. Desenvolvimento de grandes programas

Navigation Window

LabVIEW Project

Variáveis compartilhadas

LabVIEW Navigation Window



- Exibe a região atual de visualização comparada a todo painel frontal ou diagrama de blocos
- Funciona bem para grandes programas

Organize e reduza o tamanho do programa com subVIs.

ni.com

92



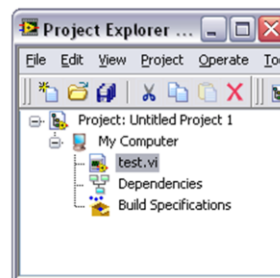
Selecione **View»Show Navigation Window** ou o atalho **Ctrl+Shift+N** para exibir esta janela.

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

Obs.: Não é recomendado que o Painel Frontal e o Diagrama de Blocos sejam muito grandes. Por isso, deve se usar as subVI's. Caso seja necessário, é recomendado que esse deslocamento seja apenas em um eixo.

LabVIEW Project

- Agrupa e organiza VIs
- Gerencia hardware e E/S
- Gerencia grandes aplicações em LabVIEW
- Gerencia VIs para múltiplos targets
- Construa bibliotecas e executáveis
- Habilita gerenciamente e monitoramento da versão



(LabVIEW » Project » New)

LabVIEW Project

Use os projetos para agrupar arquivos (tanto do LabVIEW quanto de outros programas), criar aplicações autônomas (usando o Build Specifications) e implementar ou publicar arquivos para os targets. Um target é um dispositivo ou máquina onde um VI pode ser executado. Quando um projeto é salvo, o LabVIEW cria um arquivo de projeto (.lvproj), que inclui informação de configuração e implementação, especificação da aplicação autônoma e referências aos arquivos no projeto.

Para construir uma aplicação autônoma e bibliotecas compartilhadas deve-se usar um project. Para trabalhar com targets (RT, FPGA entre outros) também deve-se usar um project. Consulte a documentação para mais informações sobre como usar projetos com os módulos LabVIEW Real-Time e LabVIEW FPGA.

Project Explorer Window

Use o **Project Explorer Window** para criar e editar projetos. Selecione **File»New Project** para exibir esta janela. Pode-se selecionar também, **Project»New Project** ou selecionar **File»New** e selecionar **Empty Project** na nova caixa de diálogo para ver a janela do **Project Explorer Window**.

Variáveis compartilhadas

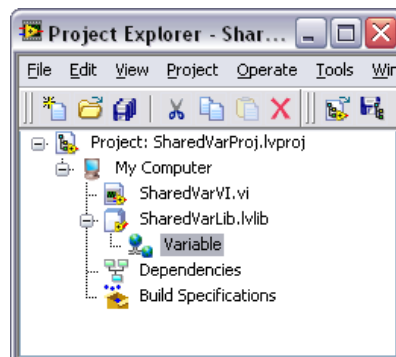
- As variáveis compartilhadas (shared variables) do LabVIEW são usadas para enviar dados entre VIs
- Tipos de variáveis:
 - Single process: Compartilha os dados entre VIs no computador local
 - Network-published: Comunica entre VIs, computadores remotos e ferramentas de hardware através do LabVIEW shared variable engine
- As variáveis compartilhadas devem existir dentro de uma biblioteca no projeto
- As variáveis compartilhadas devem ser publicadas para ficar disponível para outros projetos e computadores remotos



Shared variables são usadas para compartilhar dados entre VIs ou entre lugares em uma aplicação que não podem ser conectados 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.

Estas shared variables devem estar dentro da biblioteca de projetos. Se você criar uma shared variable de um target ou pasta que não estiver dentro da biblioteca, o LabVIEW cria uma nova biblioteca e coloca a shared variable dentro da mesma. Você deve publicar a shared variable para que ela esteja disponível em outros projetos e computadores remotos. É possível fazer isso executando o VI que contém a shared variable. Pode-se também clicar com o botão direito do mouse na biblioteca do projeto que contém as variáveis compartilhadas e selecionar **Deploy**

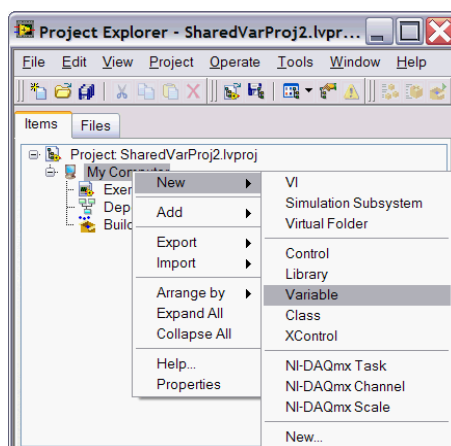




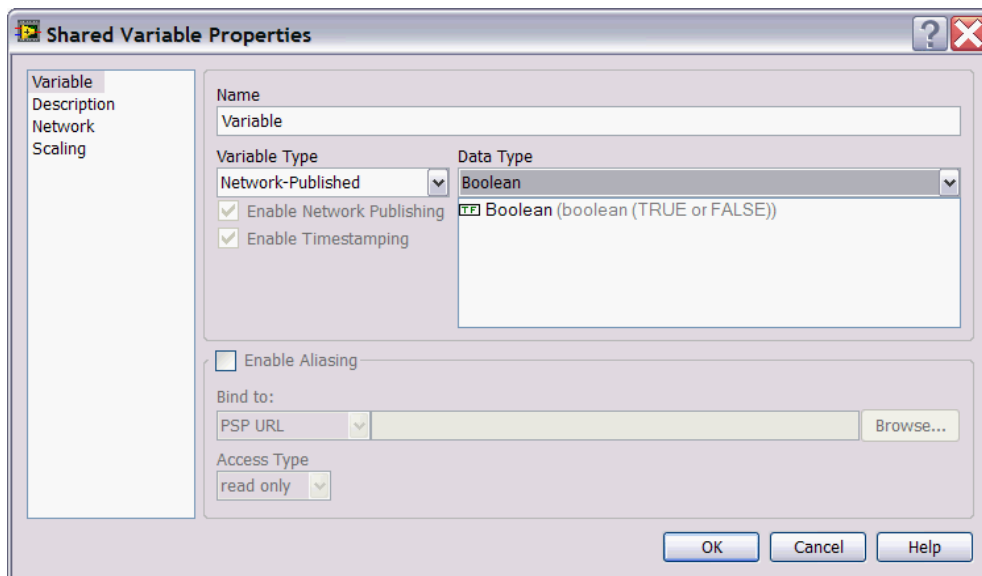
Exercício 12 – Shared Variable

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

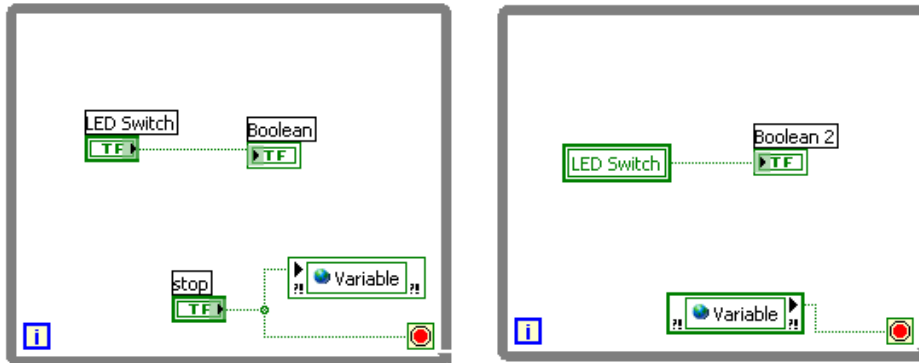
1. Abra o VI de variável local criado no Exercício 11.
2. Renomeie o VI selecionando **File » Save As... e Rename**.
3. Selecione **Project » New Project** na Barra de Menu. Isso irá criar um novo projeto quando pedido, selecione **Add** para acrescentar o VI aberto ao projeto.
4. Salve o projeto selecionando **Project » Save Project** na janela Project Explorer.
5. Crie uma Shared Variable clicando com o botão direito do mouse em My Computer e selecionando **New » Variable**.



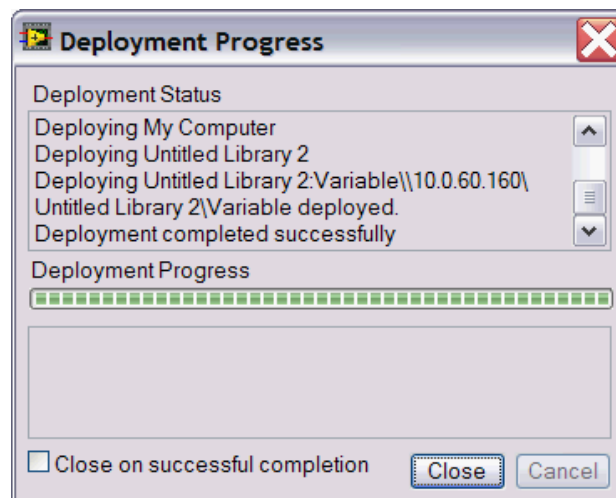
6. Na janela de configuração, nomeie a variável e selecione “Boolean” no Data Type. Deixe as demais opções com seus valores default e clique em **OK**.



7. Como as 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**.
8. Shared Variables podem ser usadas facilmente clicando e arrastando-as do Project Explorer ao VI. Clique e arraste a Shared Variable que você criou para o diagrama de blocos do Local Variable VI.
9. Delete a variável local que controla o botão stop no segundo loop.
10. Coloque a variável no segundo loop e ligue-a ao terminal de saída.
11. Coloque outra Shared Variable no primeiro loop. Esta Shared Variable irá escrever a informação que é lida no segundo loop.
12. 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 Shared Variable. O código completo deve parecer com o seguinte:



13. Execute o VI. Quando visualizar a janela abaixo, selecione **Close**. Veja que quando clicamos no botão Stop ambos os loops e o VI param.



(Fim do exercício)

Parte VI – Controle de Instrumentos

- A. Visão geral de controle de instrumentos
- B. GPIB
- C. Serial
- D. Instrument I/O Assistant
- E. VISA
- F. Instrument Drivers e IDNet

Quais os tipos de instrumentos podem ser controlados?

- GPIB
- Serial
- Instrumentos modulares
- Aquisição de imagem
- Controle de movimento
- USB
- Ethernet
- Porta Paralela
- CAN

Quando configurar um sistema de teste, normalmente é necessário usar diferentes tipos de instrumentos. Estes instrumentos podem incluir barramentos de diferentes protocolos, como GPIB, serial, USB, Ethernet, porta paralela e CAN, instrumentos modulares, aquisição de imagem e controle de movimento. Quando usar um PC para comunicar com qualquer tipo de instrumento, deve-se conhecer as propriedades do instrumento, como o protocolo de comunicação.

GPIB

- GPIB (General Purpose Interface Bus)
- Normalmente usado em instrumentos de bancada autônomos para controlar medidas e comunicar dados
- Interface de comunicação paralela digital característica de 8 bits
- Definido pelos padrões IEEE 488.1 e 488.2

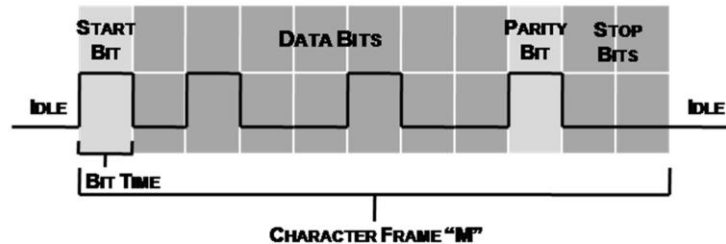
GPIB (General Purpose Interface Bus) é definido pelos padrões ANSI/IEEE 488.1-1987 e 488.2-1992 e descreve uma interface padrão para comunicação entre instrumentos e controladores de vários fabricantes. Normalmente usado em instrumentos de bancada autônomos para controlar medidas e comunicar dados. A comunicação GPIB é realizada através de uma interface digital, paralela e de 8 bits com handshaking de três fios e podem atingir taxas de transferência de dados superiores a 1MB/s.

Consulte o site de suporte a GPIB da National Instruments GPIB em ni.com/gpib para informação adicional sobre GPIB.



Serial

- Comunicação serial transmite um bit por vez na linha de transmissão
- Normalmente não exige hardware externo
- Quatro parâmetros: taxa de transmissão (baud rate), bits de dados, bit de paridade e bits de parada.



Comunicação serial transmite dados entre um computador e um dispositivo periférico. O protocolo de comunicação serial usa um transmissor para enviar um único bit de dado por vez em uma linha de comunicação para o receptor. Este método tem seu melhor desempenho quando as taxas de transferência são baixas ou quando o dado deve ser transmitido por longas distâncias. A maioria dos computadores possuem ao menos uma porta serial, então, não é preciso nenhum hardware adicional.

Deve-se especificar quatro parâmetros para comunicação serial: taxa de transmissão (baud rate), bits de dados, bit de paridade e bits de parada. Um quadro característico transmite cada caractere com um bit de start, seguido pelo bit de dado, como mostrado acima para o caractere M.

Existem padrões para portas seriais; no entanto, os mais comuns são:

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

RS 422 (padrão AIA RS-422A)

RS 485 (padrão EIA-485)

Instrument I/O Assistant

- VI Expressa do LabVIEW usada para comunicar-se com instrumentos baseados em mensagens
- Comunica –se com instrumentos que utilizam interface serial, Ethernet ou GPIB
- Use o Instrument I/O Assistant quando o driver de instrumento não está disponível



O VI Expresso Instrument I/O Assistant pode ser usado para comunicar com instrumentos baseados em mensagens e converte a resposta do dado bruto para uma representação ASCII. Quando um driver de instrumento não estiver disponível, pode-se usar o Instrument I/O para comunicar-se com um instrumento que usa a interface serial, Ethernet ou GPIB.

O Instrument I/O Assistant organiza a comunicação com o instrumento em passos sequenciais. Para usá-lo, coloque os passos de maneira ordenada. Conforme os passos são adicionados à sequência, eles aparecem na janela **Step Sequence**. O LabVIEW adiciona terminais de entradas e saídas ao VI Expresso Instrument I/O Assistant no diagrama de blocos que correspondem ao dado recebido do instrumento.

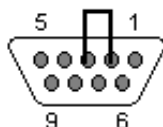


Exercício 13 – *Loopback* com APIs VISA

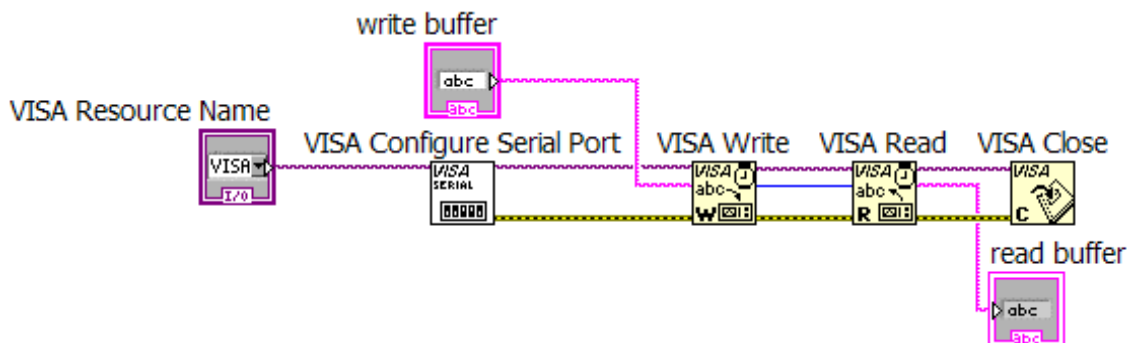
Nota: Este exercício usa a porta serial e exige um cabo serial e um fio. Alguns PCs tem a porta serial integrada. Pode-se usar facilmente o Instrument I/O Assistant para comunicação com um dispositivo GPIB, mas será necessário a porta ou o hardware GPIB ao invés da porta serial.

Complete os passos abaixo para criar um VI que realiza um teste de *loopback* usando a porta serial.

1. Conecte o cabo serial na porta COM do computador.
2. Conecte as linhas de transmissão e recepção do cabo serial ligando os pinos 2 e 3, como mostrado abaixo.



3. Abra um VI em branco da tela *Getting Started*.
4. Abra o diagrama de blocos e insira o VI VISA Configure Serial Port VI clicando com o botão direito no diagrama de blocos para abrir a paleta de **Funções** e selecione **Instrument I/O » Serial » VISA Configure Serial Port**.
5. Insira as funções VISA Write, VISA Read e VISA Close no diagrama de blocos abrindo a paleta de **Funções** e selecionando **Instrument I/O » Serial**.
6. Crie um controle **VISA Resource Name** clicando com o botão direito no terminal **VISA Resource Name** do VI VISA Configure Serial Port VI e selecione **Create » Control**.
7. Repita o passo anterior para criar controles para o terminal **Write Buffer** no VI VISA Write e no terminal **Read Buffer** do VI VISA.
8. Conecte os fios no diagrama de blocos como mostrado abaixo:



9. Mude para o painel frontal e entre um texto qualquer no **controle String**. Execute o VI e observe que o texto digitado no **controle String** aparece no **indicador String**.

(Fim do exercício)

VISA

- VISA (Virtual Instrument Software Architecture)
- API de alto nível que chama drivers de baixo nível
- Pode controlar instrumentos baseados em computadores, VXI, GPIB e serial
- Realiza a chamada do driver apropriado dependendo do instrumento usado

Virtual Instrument Software Architecture (VISA) é a base para o LabVIEW instrument driver. VISA não fornece de forma direta a capacidade de programar instrumentação mas serve como uma API de alto nível que realiza chamadas de drives de nível mais baixo. VISA pode controlar VXI, GPIB, serial ou outros instrumentos baseados em computador e realiza a chamada correta do driver dependendo do tipo de instrumento.

No LabVIEW, VISA é uma simples biblioteca de funções que adapta-se a instrumentos diferentes, tornando desnecessário utilizar paletas de entrada e saída de dados separadas. A terminologia abaixo é usada para programação VISA:

Resource: Qualquer instrumento no sistema, incluindo portas seriais e paralelas.

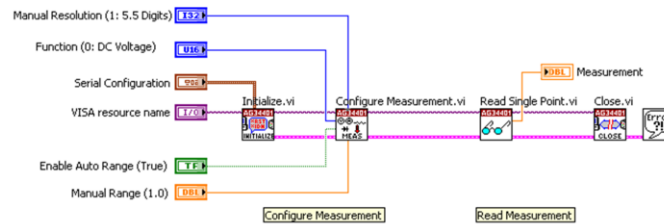
Session: Canal de comunicação que é usado pelo VISA para identificar uma referencia específica para aquele instrumento.

Instrument Descriptor: Nome exato do instrumento (exemplos

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]

Drivers de instrumentos

- Drivers pronto para uso no LabVIEW
- Consiste em um conjunto de VIs que controla um instrumento programável
- VIs correspondem à operação do instrumento: configuração, triggering e leitura de medidas
- Ideal para começar, porque o protocolo de programação para cada instrumento é conhecido



ni.com

104



Um driver de instrumento no LabVIEW é um conjunto de VIs que controlam um instrumento utilizando programação. Cada VI no driver corresponde a uma operação específica do instrumento como configuração, triggering e leitura de medidas. Isto reduz de forma significativa o tempo de desenvolvimento porque pode-se iniciar o uso do instrumento no LabVIEW sem um conhecimento profundo do protocolo de comunicação.

A imagem acima é um exemplo de um driver de instrumento para o multímetro digital (DMM) da 34401 da Agilent, que inicializa, configura, lê a medida, fecha a sessão com o instrumento e verifica se houve erros.

IDNet

- IDNet (Instrument Driver Network)
- Instrument Driver Finder integrado ao LabVIEW

Tools » Instrumentation » Find Instrument Drivers
Help » Find Instrument Drivers

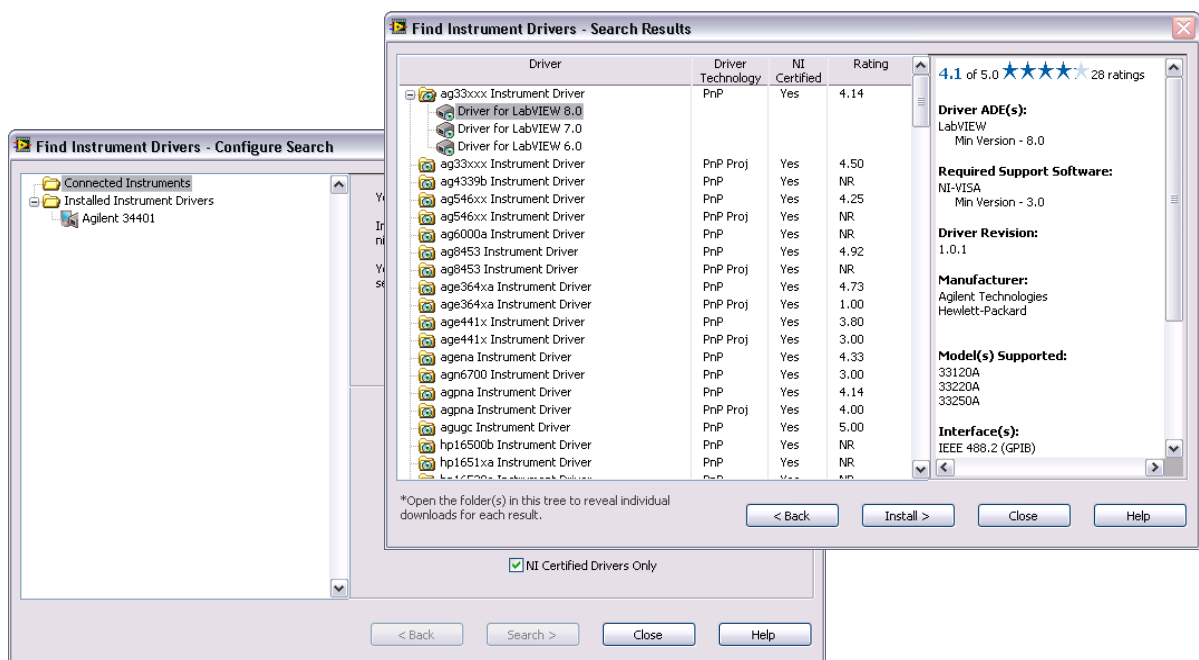
- Pode ser encontrado na rede em ni.com/idnet

ni.com

105



Pode-se encontrar a maioria dos drivers de instrumento usando o Instrument Driver Finder integrado ao LabVIEW, que pode ser acessado selecionando pelos caminhos **Tools » Instrumentation » Find Instrument Drivers** ou **Help » Find Instrument Drivers**. O Instrument Driver Finder conecta ao ni.com para encontrar os drivers de instrumento. Utilizando o finder, pode-se visualizar os instrumentos conectados e com os drivers atualmente instalados, bem como, é possível procurar os drivers pelo fabricante e por palavra chave



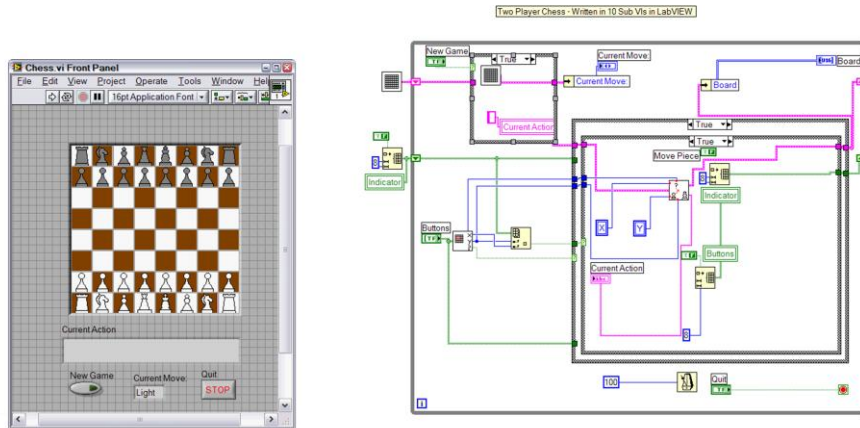
Estudos de caso LabVIEW

O que pode ser realizado no LabVIEW? Jogos!

- Xadrez para dois jogadores
- Indicador da posição do mouse

Xadrez para dois jogadores

Escrito em LabVIEW usando subVIs e controles personalizados

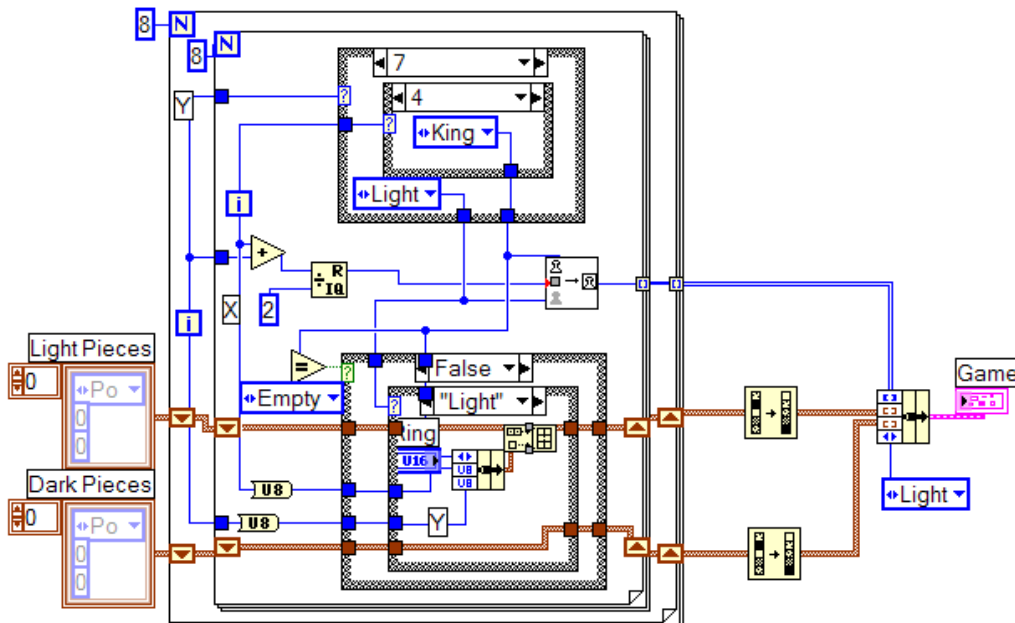


ni.com

107

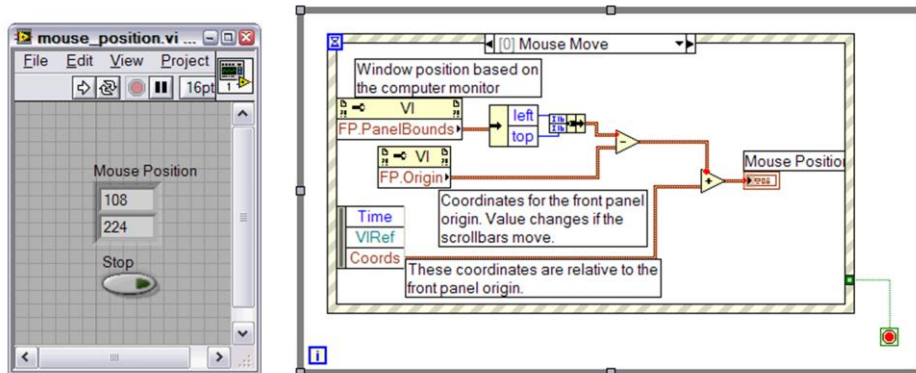
NATIONAL
INSTRUMENTS

Este VI do LabVIEW permite que duas pessoas joguem xadrez. Usando alguns controles personalizados e várias construções do LabVIEW que você conheceu nesse curso, o VI determina se o movimento é válido e exibe o jogo ocorrendo. O diagrama de blocos acima gerencia o VI de alto nível que controla os subVIs e aciona a interface de usuário. O diagrama de blocos abaixo é usado para reiniciar o tabuleiro.

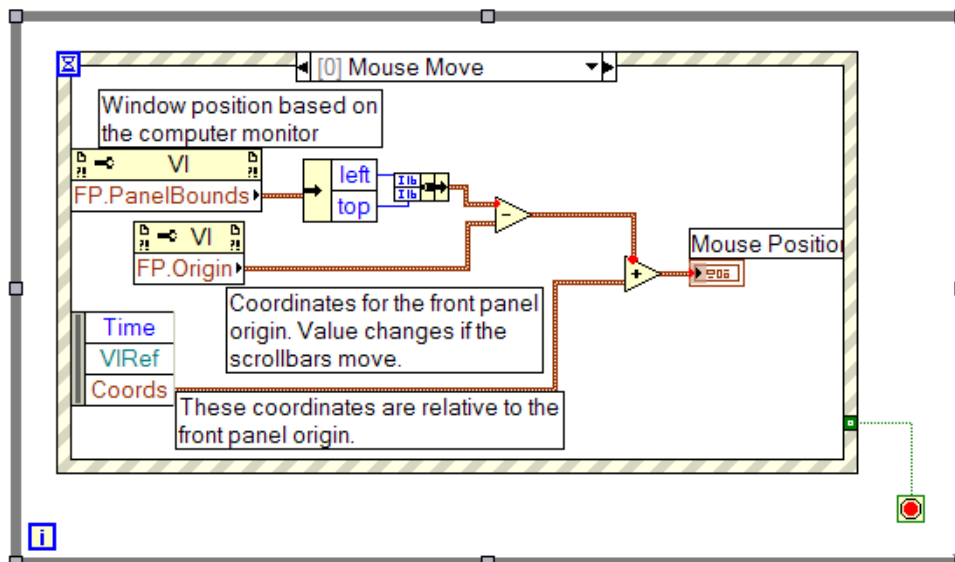


Indicador da posição do mouse

Calcula a posição do cursor do mouse no monitor usando o LabVIEW e estruturas de evento.



Este VI utiliza funções mais avançadas do LabVIEW para calcular a posição do cursor do mouse no monitor. O VI usa dois elementos principais. O primeiro, estruturas event, trabalha como estruturas case, mas elas são disparadas por eventos gerados pelos elementos no painel frontal. Estes eventos podem ser o movimento ou o clicar com o mouse, usar o teclado entre outros. O outro elemento que este VI utiliza são os property nodes. Com property nodes, pode-se ler ou definir propriedade de vários elementos do VI e elementos do sistema. Estas são apenas algumas das muitas ferramentas do LabVIEW que podem ser “descobertas”.



O que há de novo no LabVIEW 2010?

- Desempenho Melhorado
- Melhorias no Ambiente de programação do LabVIEW
- Desenvolvimento de Grandes Aplicações
- Melhorias na Transferência de Dados e na Configuração de HW Distribuídos



ni.com

109



Recursos adicionais

- NI Academic Web e Student Corner
 - ni.com/academic
 - ni.com/textbooks
 - Obtenha sua própria cópia do LabVIEW Student Edition
- NI KnowledgeBase
 - ni.com/kb
- NI Developer Zone
 - ni.com/devzone
- Certificação LabVIEW
 - CLAD (Certified LabVIEW Associate Developer Exam)
 - Certificação reconhecida pela indústria)
 - <http://sine.ni.com/nips/cds/view/p/lang/en/nid/14438>



ni.com

110



NI ACADEMIC

National Instruments gives educators the tools to make it possible for students to do engineering.

The NI Academic Program

- Educators**: Learn how to incorporate NI LabVIEW in your classroom and laboratory for teaching. [Go to section](#)
- Researchers**: Explore how to use LabVIEW in scientific computing and experimental research applications. [Go to section](#)
- Students**: Discover how to use LabVIEW for your next class assignment and student design project. [Go to section](#)

Shop Online
View Products with Academic Discounts
See If You Qualify for Discounts

Browse Featured Products

- Academic Software Licensing
- LabVIEW
- NI Multisim
- NI ELVIS
- Download Academic Brochure
- Academic Product Bundles
- NI myDAQ

View Application Areas

- Measurements and Instrumentation
- Circuit Design
- Control Design and Simulation
- Embedded Systems
- RF and Communications
- Signal and Image Processing

What's New

Enter the LabVIEW Student Design Competition and Win Cash Prizes
Students who use LabVIEW in their design project will have the opportunity to win cash prizes and trips to NIWeek in Austin, Texas.
[Competition details](#)

LabVIEW Robotics
Teach robotic design principles with new LabVIEW Robotics
[Learn more](#)

Addressing the Engineering Grand Challenges
Discover how universities are using LabVIEW in the class and lab to address global challenges
[Watch the webcast](#)

My Profile | RSS | Privacy | Legal | Contact NI | © 2011 National Instruments Corporation. All rights reserved. | [Email this Page](#)

O programa de certificação em LabVIEW

Architect

- Domínio de LabVIEW
- Perito em desenvolvimento de grandes aplicações
- Qualificado para liderar equipes de projeto

Certified
LabVIEW
Architect

Developer

- Conhecimento avançado em LabVIEW e experiência em desenvolvimento de aplicações
- Capacidade de gerenciar projetos

Certified LabVIEW
Developer

Associate Developer

- Proficiência em trabalhar com o ambiente LabVIEW
- Alguma experiência em desenvolvimento de aplicações

Certified LabVIEW Associate
Developer

Fundamentals Exam

- Teste de pré-certificação

Testes gratuitos na rede

ni.com

111



Hoje, cada vez mais companhias estão procurando experiência em LabVIEW em suas entrevistas. O programa de certificação foi construído em cima de uma série de exames profissionais. A certificação LabVIEW é usada para validar experiência e habilidades com LabVIEW para oportunidades de emprego.

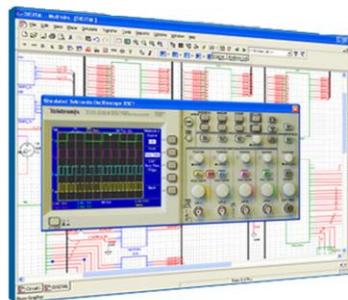
O Certified LabVIEW Associate Developer é o primeiro passo de certificação em LabVIEW e demonstra um bom conhecimento para usar o LabVIEW e seu ambiente. Como um aluno, o certificado de Associate Developer diferencia suas habilidades com o LabVIEW para oportunidades de emprego e também traz reconhecimento de sua experiência com o LabVIEW. O CLAD é um exame de múltipla escolha que dura uma hora e é aplicado em diversos lugares do país. O exame cobre vários tópicos do ambiente LabVIEW incluindo conceitos de fluxo de dados, estruturas de programação, técnicas avançadas de leitura e gravação de arquivos, prática de programação modular, propriedades de VIs e referências de controle.

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

Os Certified LabVIEW Developer e Architect (CLD e CLA, respectivamente) são certificados profissionais que validam um conhecimento avançado de LabVIEW e experiência em desenvolvimento de aplicações. Adicionalmente, o certificado CLA demonstra habilidades em liderar times de projetos e experiência desenvolvendo grandes aplicações. Estes exames, são provas práticas que duram quatro horas e são conduzidos pela National Instruments.

NI Multisim e Ultiboard

- Software para aprendizado de eletrônica mais popular no mundo
- Mais de 180,000 usuários na industriais e acadêmicos
- Versões para estudantes disponíveis
- ni.com/multisim



ni.com

112



O software Multisim integra a simulação SPICE e a captura de esquemáticos em um intuitivo laboratório de eletrônica em seu PC. Como única companhia a desenvolver produtos especificamente para o mercado educacional, nosso software está se tornando a ferramenta principal de aprendizagem e ensino para milhares de educadores.

Multisim - Simulation and Capture

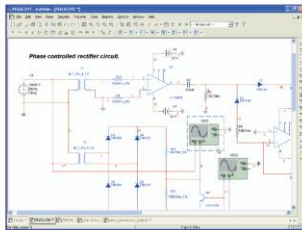
O Multisim é um programa de simulação e captura intuitivo, que permite aos educadores e alunos criarem circuitos completos rapidamente contendo tanto componentes analógicos quanto digitais. O Multisim também adiciona a capacidade de co-simulação para microcontroladores, o que permite incluir um MCU, programado na linguagem Assembly, em conjunto com o circuito modelado em SPICE.

Ultiboard - PCB Layout

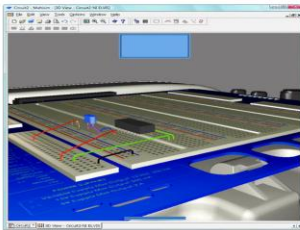
Com o Ultiboard, estudantes podem trabalhar em implementações físicas e de construção de circuitos em PCBs. O esquemático do Multisim pode ser importado para o Ultiboard com um único clique.

Multisim integrado com LabVIEW

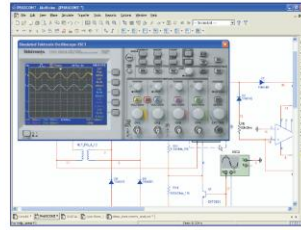
1. Crie o esquemático



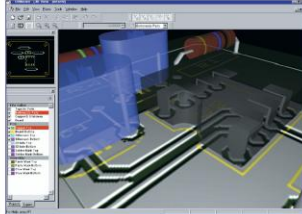
2. Protoboard virtual



3. Simule



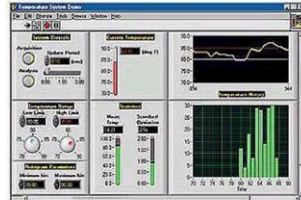
4. Layout PCB



5. Teste



6. Compare



ni.com

113



1. Multisim - Esquemáticos

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

2. Multisim – Placa de Testes Virtual

- Técnicas de Placas de Testes
- Sincronizado com o esquemático
- Relatório de fiação para o Elvis (Passo 5)

3. Multisim – Simulação

- biblioteca de 13.000 componentes
- 20 instrumentos virtuais
- Mudanças durante a execução
- 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 embarcado

5. NI ELVIS II – Teste

- Instrumentação
- Aquisição de dados
- Prototipagem

6. LabVIEW – Comparação

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

O próximo passo

Realize o *Quiz LabVIEW* gratuitamente
Disponível na página de Treinamento e Certificação
ni.com/brasil

O primeiro passo para ser certificado em LabVIEW!



ni.com

114



Continue seu aprendizado

Treinamentos com instrutores:

- LabVIEW Core I e II: Aprofunde seu conhecimento do LabVIEW como ferramenta e aprenda a alterar programaticamente a Interface de Usuário e a criar Aplicações Autônomas
- LabVIEW Core III e Conectividade: Aprenda sobre o desenvolvimento de grandes projetos, programação com eventos e comunicação do LabVIEW com diversos software
- Cursos de hardware como o de aquisição de dados, LabVIEW Real-Time e LabVIEW FPGA

©2010 National Instruments. Todos os direitos reservados. LabVIEW, Multisim, National Instruments, NI e ni.com são marcas registradas da National Instruments. Linux® é uma marca registrada de Linus Torvalds nos E.U.A. e em outros países. Outros produtos e companhias listadas são marcas registradas de suas respectivas empresas.

Copyright

© 2005-2009 National Instruments Corporation. All rights reserved.

Under the copyright laws, this publication may not be reproduced or transmitted in any form, electronic or mechanical, including photocopying, recording, storing in an information retrieval system, or translating, in whole or in part, without the prior written consent of National Instruments Corporation.

National Instruments respects the intellectual property of others, and we ask our users to do the same. NI software is protect by copyright and other intellectual property laws.

Trademarks

National Instruments, NI, ni.com and LabVIEW are trademarks of National Instruments Corporation. Refer to ni.com/trademarks for more information about National Instruments trademarks.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries. Product and company names mentioned herein are trademarks or trade names of their respective companies.

Patents

For patents covering National Instruments products/technology, refer to the appropriate location:

Help»Patents in your software, the patents.txt file on your media, or the *National Instruments Patent Notice* at ni.com/patents.