

Engenharia

Sistemas Computacionais
- Automação -

UNISAL

Centro Universitário Salesiano de Campinas

Prof. Gale

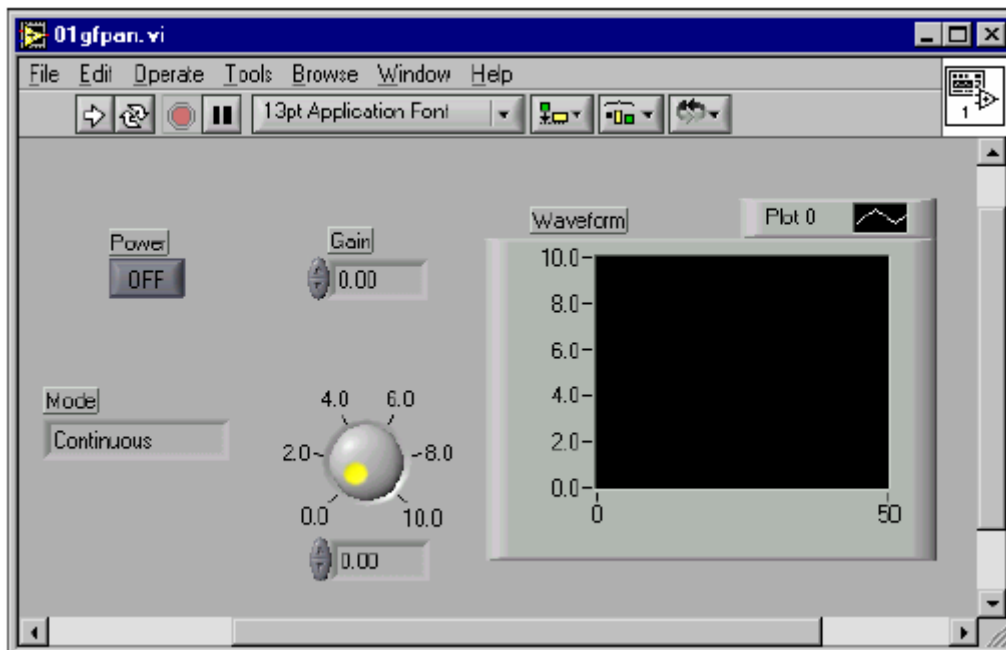
ÍNDICE

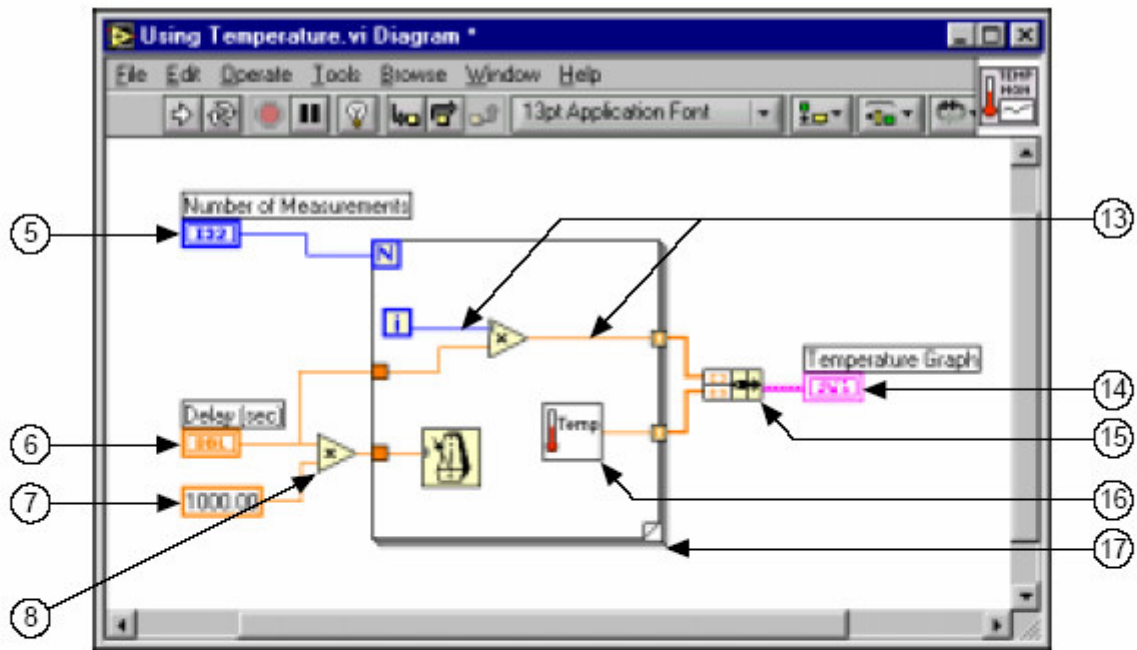
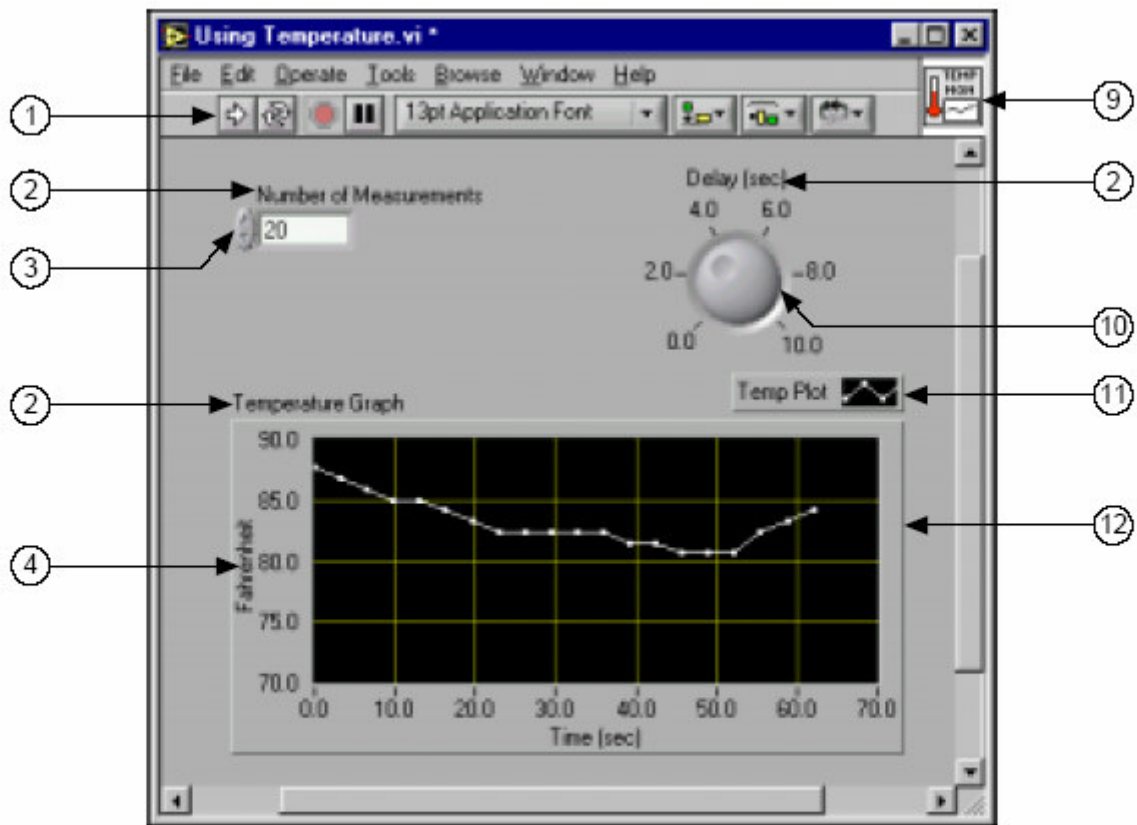
Assunto	Pág.
1ª. Aula	03
Atividade no.1	06
2ª. Aula	07
Atividade no. 2	09
3ª. Aula	10
Atividade no. 3	11
4ª. Aula	12
Atividade no. 4	13
5ª. Aula	14
Atividade no. 5	15
6ª. Aula	16
7ª. Aula	17
8ª. Aula	19
9ª. Aula	21
10ª. Aula	25
Guia de Referência	27
Documentação National	28

1ª AULA LABVIEW – PROF. GALE



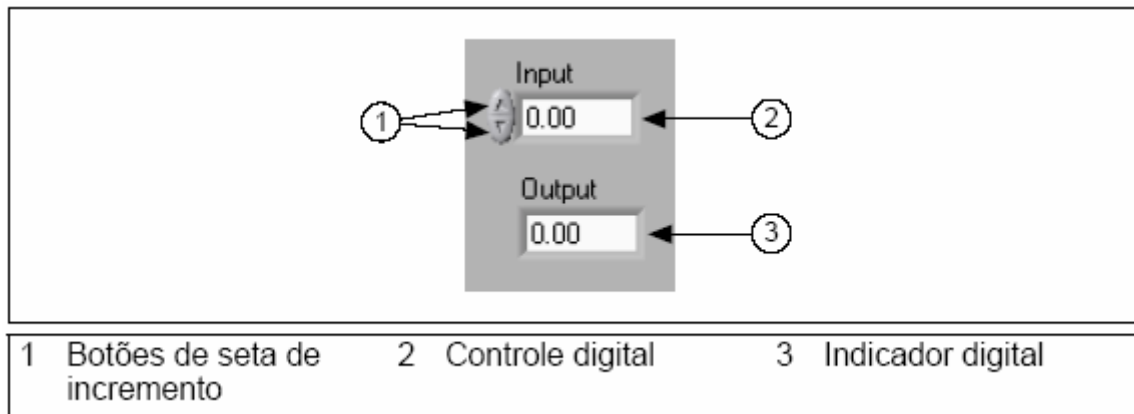
Painel Frontal





- | | | | |
|-----------------------------|---|---------------------------------|---------------------------|
| 1 Barra de ferramentas | 5 Terminal de controle numérico digital | 9 Ícone | 14 Terminal do gráfico XY |
| 2 Título do objeto | 6 Terminal de controle numérico | 10 Controle numérico deslizante | 15 Função de agrupamento |
| 3 Controle numérico digital | 7 Constante numérica | 11 Legenda de plotagem | 16 SubVI |
| 4 Legenda livre | 8 Função de multiplicação | 12 Gráfico XY | 17 Estrutura de For Loop |
| | | 13 Via de ligação de dados | |

Atividade 01



Controle (entrada) → Indicador (Saída)

- Montar um programa que realize a soma de duas entradas e exiba na saída
- Montar um programa que realize as 4 operações básicas entre dois valores fornecidos pelo usuário
- Montar um programa que leia um valor qualquer digitado pelo usuário e converta este valor de metros para quilômetros, hectômetros, decâmetros, decímetros, centímetros e milímetros.
- Montar um programa que leia um número inteiro qualquer e exiba o seu sucessor e o seu antecessor
- Montar um programa que realize uma conversão de temperaturas. O usuário fornece o valor em Celsius e a saída é em Fahrenheit e kelvin.
- Montar um programa que entre com um número qualquer e exiba a raiz quadrada do número, o quadrado, o seno e o cosseno do número.
- Montar um programa que realiza a soma de um valor qualquer definido pelo usuário e um valor randômico definido pelo computador.
- Montar um programa que entre com os catetos de um triângulo retângulo e calcule a hipotenusa.

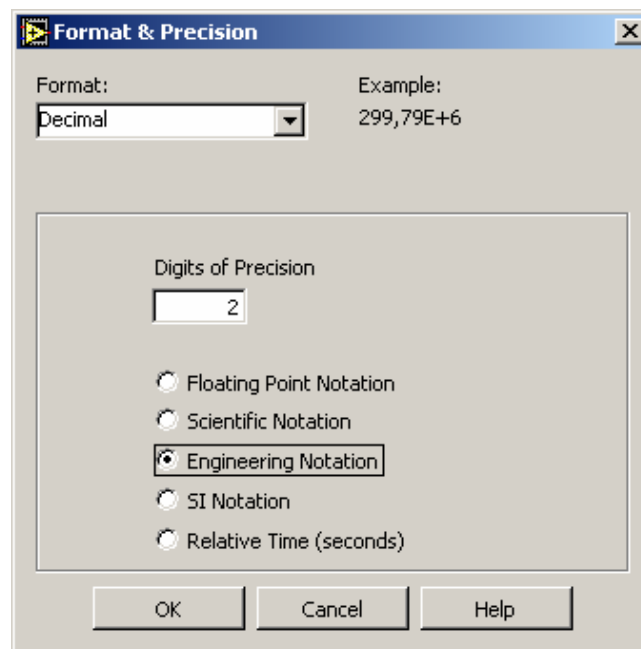
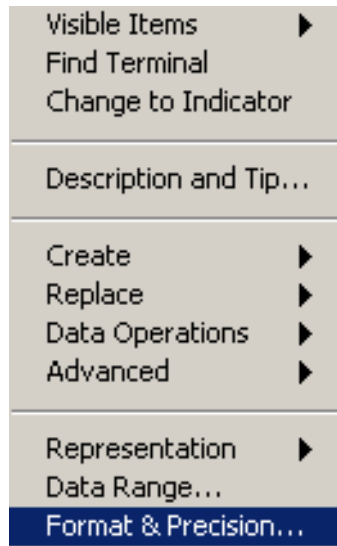
Desafio: Montar um programa que permita o usuário digitar um número qualquer e exiba o número da unidade, dezena e centena separados.

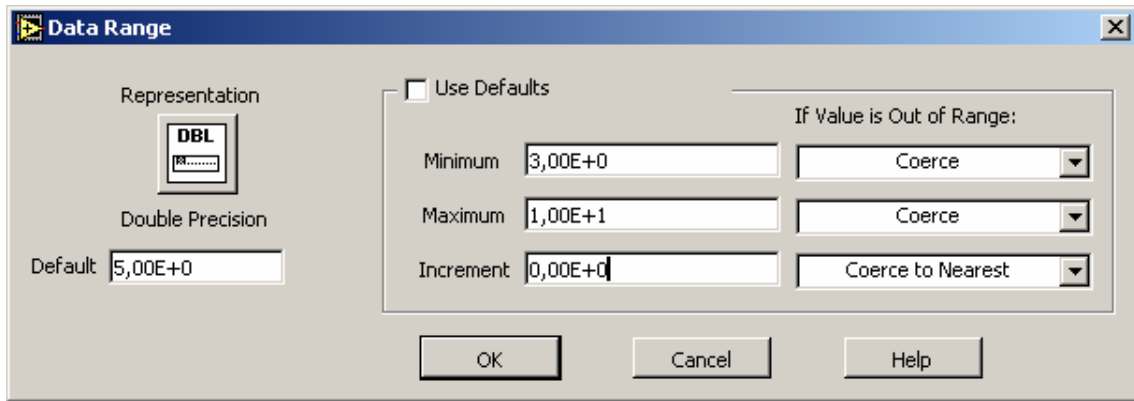
Laboratory Virtual Instrument Engineering Workbench
Prof. Gale

2ª AULA LABVIEW – PROF. GALE

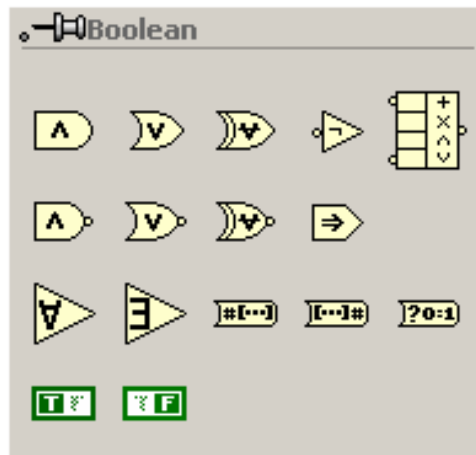
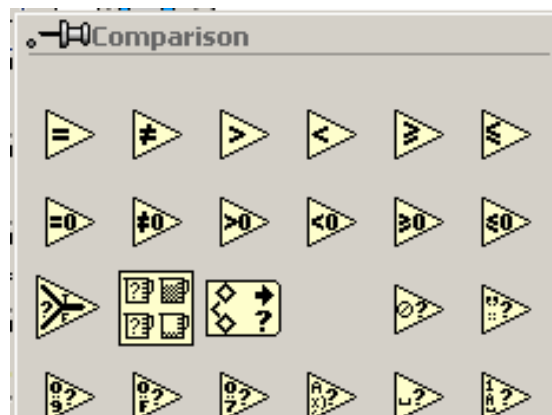
Formatando Dados

(Utilizando o botão direito do mouse)

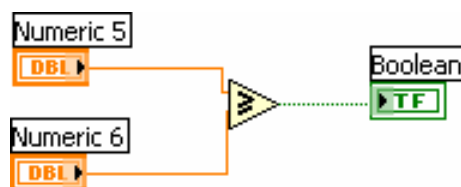




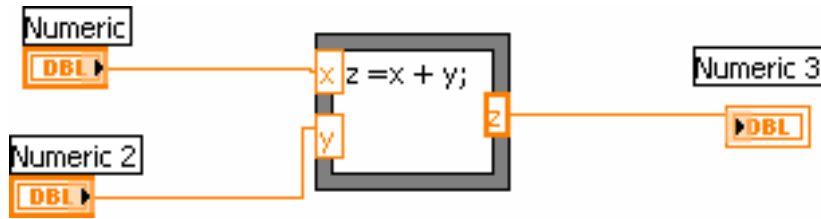
Operadores de Comparação



Exemplo de operadores de Comparação



Exemplo de Fórmula (Structure – Formula Node (= e ;))



Atividade 02

Fórmulas:

- 1.) Hip = $\text{raiz}(a^2 + b^2)$
- 2.) $\Delta = b^2 - 4ac$
 $X1 = \frac{-b + \text{raiz}(\Delta)}{2a}$
 $X2 = \frac{-b - \text{raiz}(\Delta)}{2a}$
- 3.) Área do Cilindro: $\frac{PI * \text{raio}^2}{2} \times \text{altura}$
- 4.) Digite um valor em km/h e exiba em m/s
- 5.) monte a sua fórmula

Comparações:

Utilize os objetos de comparações para efetuar comparações entre:

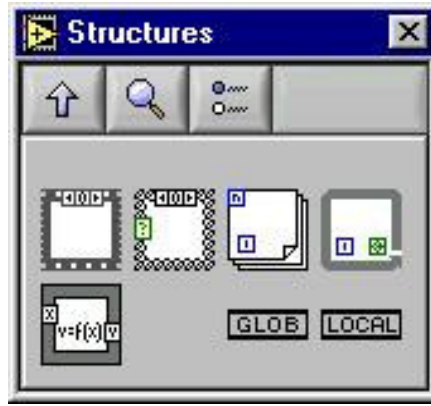
- 6.) dois números reais
- 7.) um número real e uma constante

Operadores Binários (portas lógicas)

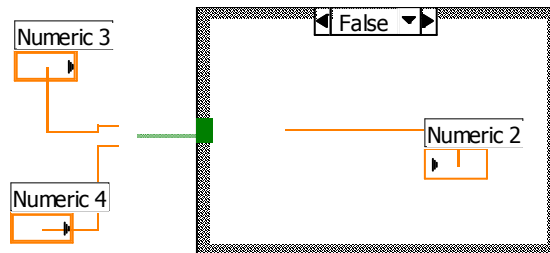
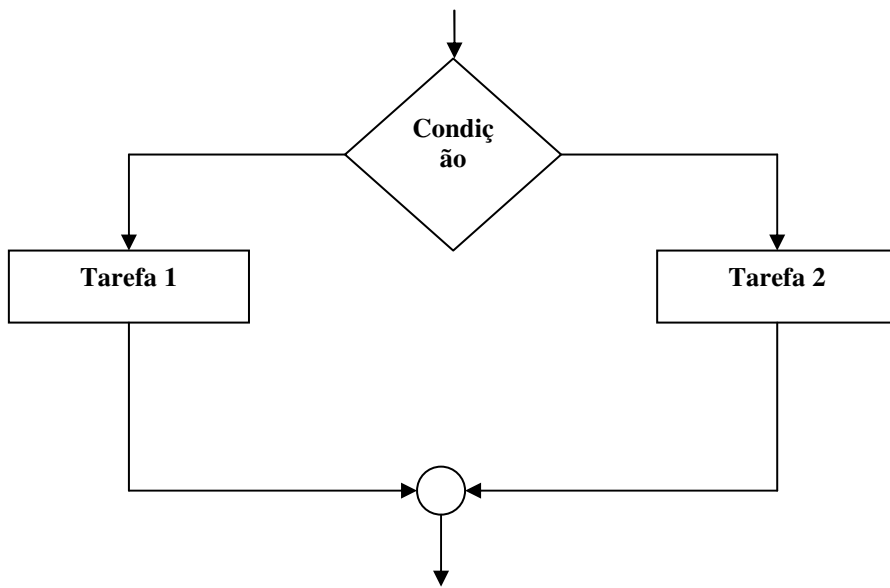
- 8.) A and B
- 9.) A or B
- 10.) A and B or C
- 11.) A and (B or C)
- 12.) não (A and B)
- 13.) A and C or A and B
- 14.) A or B and B or C
- 15.) B and (não A)
- 16.) não (A and B and C)
- 17.) A and (B or (A and C))

3ª AULA LABVIEW – PROF. GALE

Structures



Case Structure:



Atividade 03

Fórmulas:

- 1.) Faça um programa que permita o usuário selecionar dois números e exiba o maior.
- 2.) Faça um programa que permita o usuário selecionar um número qualquer. Se o número for maior que 0 exiba a raiz quadrada do mesmo; caso contrário eleve o número ao quadrado
- 3.) Faça um programa que permita o usuário escolher um número qualquer e exiba o módulo do número.
- 4.) Faça um programa que permita o usuário escolher um número qualquer e exiba se o número é par ou ímpar.
- 5.) Faça um programa que calcule uma equação do 2º. Grau. Lembre-se: se Delta for negativo não existem raízes reais, caso contrário calcule as duas raízes.
- 6.) Uma empresa resolveu dar um aumento de salário aos funcionários de acordo com a seguinte tabela:

0 → 500,00 aumento de 25%

500,00 → 1500,00 aumento de 10%

1500 → aumento de 5%

Faça um programa que permita a entrada do salário, mostre o aumento e o novo salário.

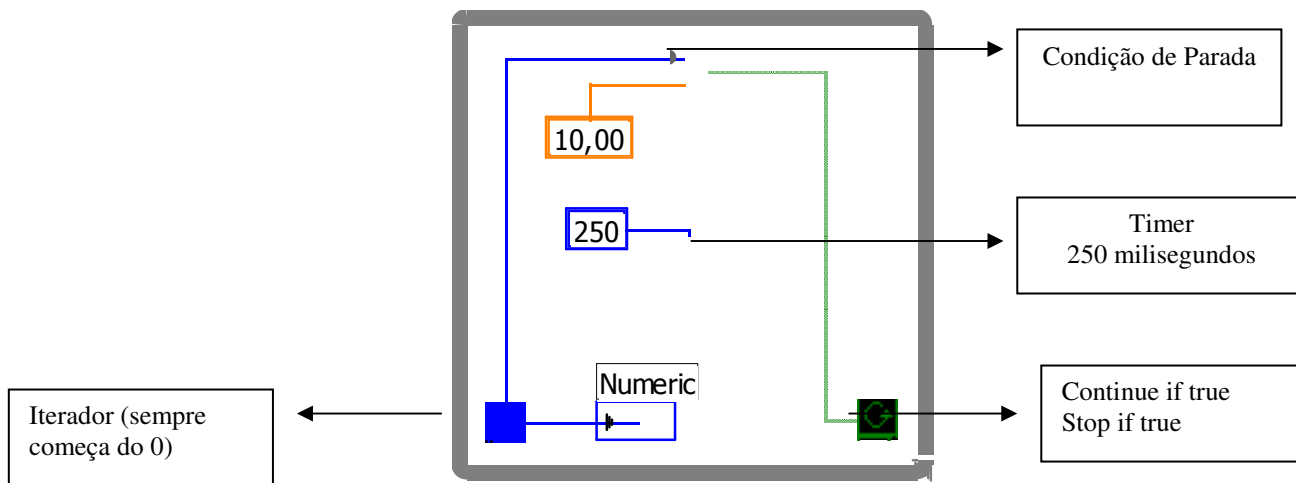
4ª AULA
LABVIEW – PROF. GALE

Estruturas de Repetição – While Loop



Semelhante a um Do Loop ou um Repeat-Until Loop em linguagens de programação baseadas em texto, um While Loop, como mostrado acima, executa um subdiagrama até que uma condição seja alcançada. A estrutura de While Loop está localizada na paleta **Functions»Structures**.

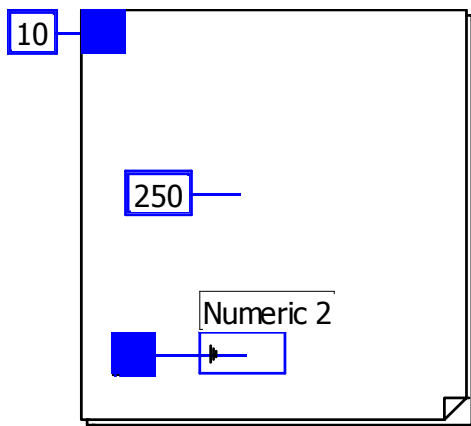
Vamos ao exemplo:



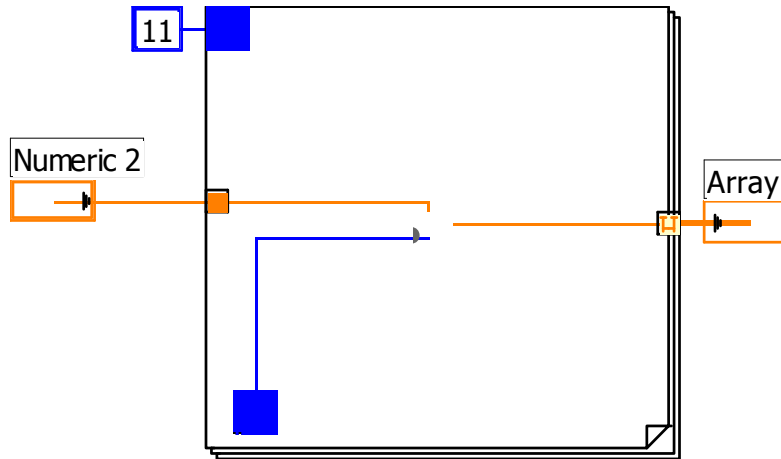
For Loop



Nesta outra estrutura você define um valor para N que será o valor final do seu loop decrementado de 1, pois a estrutura começa na posição 0.



Exemplo do cálculo de uma tabuada definida pelo usuário:



Atividade 04

- 1.) Faça um programa que mostre uma seqüência de 1 a 20 de 2 em 2.
- 2.) Faça um programa que monte uma tabela de 1 a 20 contendo o número, o quadrado e a raiz quadrada do número.
- 3.) Faça um programa que exiba quantos alunos em uma sala de 10 alunos, possuem idade superior ou igual a 18 anos e quantos possuem idade inferior a 18 anos.
- 4.) Faça um programa que permita o usuário digitar 10 números e no final exiba quantos números são pares, quantos ímpares, quantos múltiplos de 5 e quantos múltiplos de 7
- 5.) Faça um programa que exiba os pares de 0 até um número definido pelo usuário.
- 6.) Faça um programa que permita o usuário digitar 10 valores e exiba a somatória desses valores no final
- 7.) Faça um programa que permita o usuário digitar um número qualquer e exiba o fatorial deste número:

Ex: $5! = 5 \times 4 \times 3 \times 2 \times 1 = 120$

Desafio:

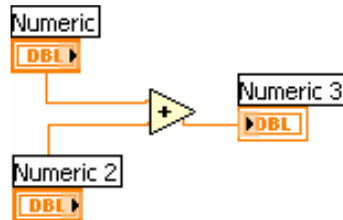
Calcular:

$$C = \frac{N!}{(N - P)! * P!}$$

5ª AULA LABVIEW – PROF. GALE

Criação de Sub-Vi

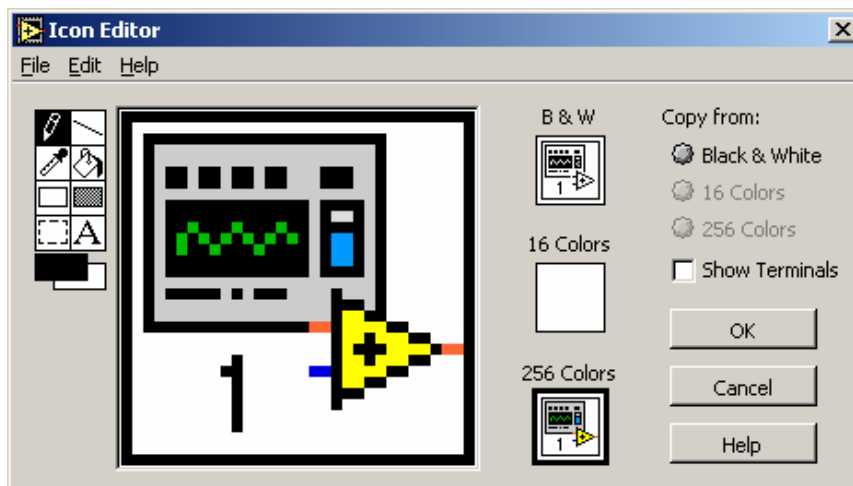
1º) Passo – Montamos a estrutura que se deseja encapsular.



2º) Passo – Selecionamos a caixa situada no canto superior direito do Painel Frontal com o botão direito do mouse



3º) Passo – Selecionamos Edit Icon



4º) Passo – Desenhamos o ícone e teclamos OK

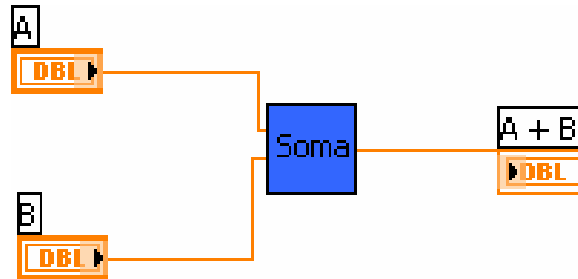


5º) Passo – Selecionamos o ícone gerado e pressionamos Show Connector. Ligamos os terminais com o mouse.

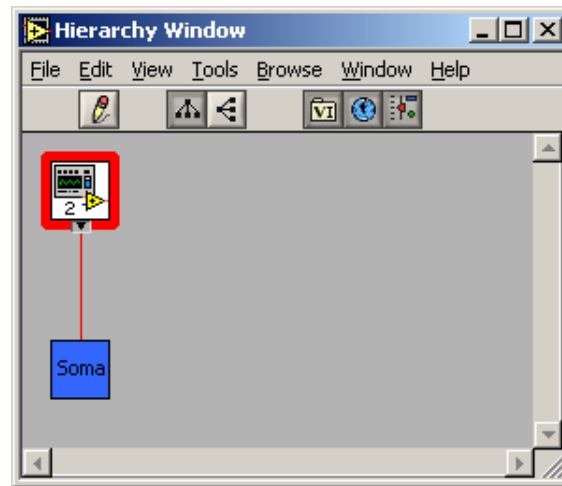
6º) Passo – Feito isso salve a sua Sub-Vi no mesmo diretório que irá montar o seu programa.

7º) Passo – Comece um novo programa e utilize a sua nova Sub-VI que se encontra no Diagrama de Blocos selecionando as funções o último ícone denominado “Select a Vi...”

Por fim, o seu novo programa, ficaria:



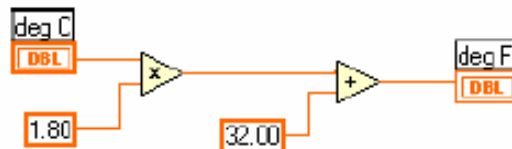
Podemos ainda verificar a hierarquia das funções: (BROWSE – Show VI Hierarchy)



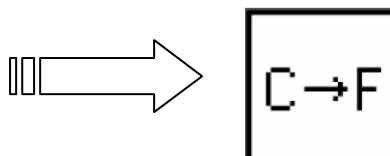
Atividade 05

8.) Desenvolva as seguintes Sub – Vi's:

a.) Transforma Celsius em Farenheit









- b.) Transforma Farenheit em Celsius
- c.) Transforma Celsius em Kelvin
- d.) Transforma Farenheit em Kelvin
- e.) Transforma Kelvin em Celsius
- f.) Transforma Kelvin em Farenheit



6ª AULA

LABVIEW – PROF. GALE

Vetores

Functions – Array →      

11

Posição Desejada

Numeric
0,00

OK Button
OK

size(s)
0

Vetor

0,00
0,00
0,00
0,00
0,00

Valor na posição do vetor
0,00

<p>Vetor de Entrada</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>10,00</td></tr> <tr><td>5,00</td></tr> <tr><td>0,00</td></tr> <tr><td>6,00</td></tr> <tr><td>7,00</td></tr> <tr><td>0,00</td></tr> <tr><td>8,00</td></tr> <tr><td>9,00</td></tr> <tr><td>11,00</td></tr> <tr><td>5,00</td></tr> </table>	10,00	5,00	0,00	6,00	7,00	0,00	8,00	9,00	11,00	5,00	<p>Inversão do Vetor</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>5,00</td></tr> <tr><td>11,00</td></tr> <tr><td>9,00</td></tr> <tr><td>8,00</td></tr> <tr><td>0,00</td></tr> <tr><td>7,00</td></tr> <tr><td>6,00</td></tr> <tr><td>0,00</td></tr> <tr><td>5,00</td></tr> <tr><td>10,00</td></tr> </table>	5,00	11,00	9,00	8,00	0,00	7,00	6,00	0,00	5,00	10,00	<p>Ordenar um Vetor</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>0,00</td></tr> <tr><td>0,00</td></tr> <tr><td>5,00</td></tr> <tr><td>5,00</td></tr> <tr><td>6,00</td></tr> <tr><td>7,00</td></tr> <tr><td>8,00</td></tr> <tr><td>9,00</td></tr> <tr><td>10,00</td></tr> <tr><td>11,00</td></tr> </table>	0,00	0,00	5,00	5,00	6,00	7,00	8,00	9,00	10,00	11,00
10,00																																
5,00																																
0,00																																
6,00																																
7,00																																
0,00																																
8,00																																
9,00																																
11,00																																
5,00																																
5,00																																
11,00																																
9,00																																
8,00																																
0,00																																
7,00																																
6,00																																
0,00																																
5,00																																
10,00																																
0,00																																
0,00																																
5,00																																
5,00																																
6,00																																
7,00																																
8,00																																
9,00																																
10,00																																
11,00																																

Valor Máximo
11,00

Valor Mínimo
0,00

Índice do Valor Máximo
8

Índice do Valor Mínimo
2

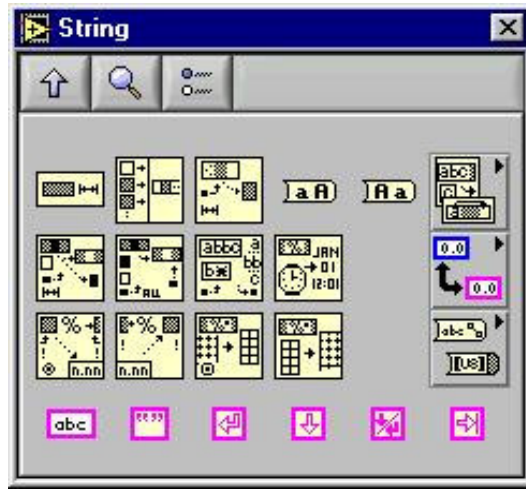
<p>Posições Ímpares</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>10,00</td></tr> <tr><td>0,00</td></tr> <tr><td>7,00</td></tr> <tr><td>8,00</td></tr> <tr><td>11,00</td></tr> <tr><td>0,00</td></tr> <tr><td>0,00</td></tr> </table>	10,00	0,00	7,00	8,00	11,00	0,00	0,00	<p>Posições Pares</p> <table border="1" style="border-collapse: collapse; text-align: center;"> <tr><td>5,00</td></tr> <tr><td>6,00</td></tr> <tr><td>0,00</td></tr> <tr><td>9,00</td></tr> <tr><td>5,00</td></tr> <tr><td>0,00</td></tr> <tr><td>0,00</td></tr> </table>	5,00	6,00	0,00	9,00	5,00	0,00	0,00
10,00															
0,00															
7,00															
8,00															
11,00															
0,00															
0,00															
5,00															
6,00															
0,00															
9,00															
5,00															
0,00															
0,00															

7ª AULA LABVIEW – PROF. GALE

Strings

Uma string é uma seqüência de caracteres ASCII exibíveis ou não exibíveis. As strings proporcionam um formato independente de plataforma para informações e dados. Algumas das aplicações mais comuns de strings são:

- Criar mensagens de texto simples.
- Transferir dados numéricos como strings de caracteres para instrumentos e, em seguida, converter as strings em dados numéricos.
- Armazenar dados numéricos em disco. Para armazenar dados numéricos em um arquivo ASCII, primeiro é necessário converter os dados numéricos em strings, antes de os gravar em um arquivo de disco.
- Instruir o usuário com caixas de diálogo ou enviar mensagens a ele através delas.



Tipo de Exibição

Tipo de exibição	Descrição	Mensagem
Exibição normal	Exibe caracteres imprimíveis, utilizando a fonte do controle. Os caracteres não imprimíveis são exibidos como caixas.	There are four display types. \ is a backslash.
Exibição de códigos "\	Exibe códigos de barras invertidas para todos os caracteres não imprimíveis.	There\sare\sfour\sdisplay\sty pes.\n\\\sis\sa\sbackslash.

Tipo de exibição	Descrição	Mensagem
Exibição de senha	Exibe um asterisco (*) para cada caractere, incluindo espaços.	***** *****
Exibição hexadecimal	Exibe o valor ASCII de cada caractere em formato hexadecimal, em vez do próprio caractere.	5468 6572 6520 6172 6520 666F 7572 2064 6973 706C 6179 2074 7970 6573 2E0A 5C20 6973 2061 2062 6163 6E73 6C61 7368 2E

Funções de Strings

String Length: retorna o número de caracteres presentes em uma string, incluindo espaços.

Concatenate Strings: concatena strings de entrada e arranjos 1D de strings em uma única string de saída.

String Subset: retorna a substring que inicia em **offset** e contém um número de caracteres de **tamanho**. O **offset** do primeiro caractere na **string** é 0.

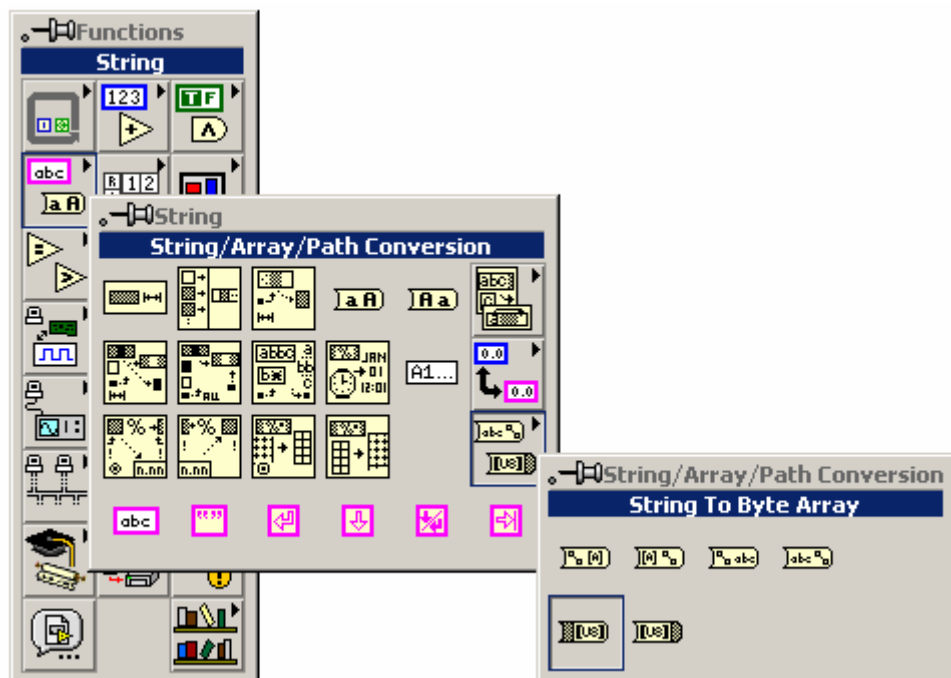
Match Pattern: procura uma **expressão regular** em uma string que inicia em **offset** e, se localizar uma correspondência, divide a string em três substrings. Se nenhuma correspondência for localizada, a **substring de correspondência** ficará vazia e a **correspondência de passagem de offset** será -1.

Convertendo dados numéricos em strings

A função **Format Into String** converte **argumentos** de qualquer formato, como dados numéricos, em uma string. Redimensione a função para aumentar o número de **argumentos**.

Convertendo strings em dados numéricos

A função **Scan From String** converte uma string que contém caracteres numéricos válidos, como 0-9, +, -, e, E e ponto final (.), em um dado numérico. A função começa a varrer a **input string** em **initial search location**. A função pode varrer a **input string** em vários tipos de dados, como numéricos ou Booleanos, com base na **format string**.



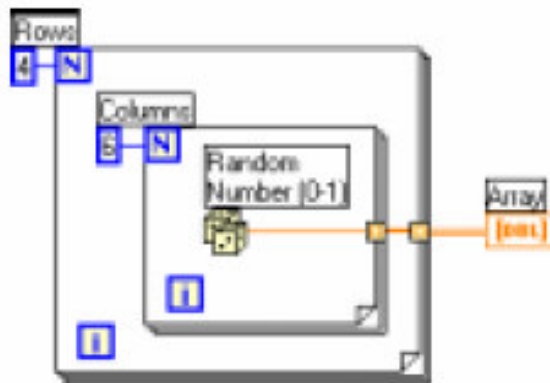
Arrays Bidimensionais

Um arranjo 2D armazena elementos em uma grade. Ele requer um índice de coluna e um índice de linha para localizar um elemento, ambos com base zero. A seguinte ilustração mostra um arranjo 2D com 6 colunas e 4 linhas, que contém $6 \cdot 4 = 24$ elementos.

		Column Index					
		0	1	2	3	4	5
Row Index	0						
	1						
	2						
	3						

Para adicionar uma dimensão de cada vez a um arranjo, clique com o botão direito na exibição de índice e selecione **AddDimension** no menu de atalho. Você também pode utilizar a ferramenta Positioning para redimensionar a exibição de índice, até que tenha todas as dimensões desejadas.

Você pode utilizar dois For Loops, um dentro do outro, para criar um arranjo 2D. O For Loop externo cria os elementos de linha e o For Loop interno cria os elementos de coluna, como mostrado no exemplo a seguir.



Funções de array:

Array Size: retorna a quantidade de elementos existentes em cada dimensão de um arranjo. Se o arranjo for n -dimensional, a saída **size** será um arranjo de n elementos. Por exemplo, a função Array Size retorna um valor **size** 3 para o seguinte arranjo.

7	4	2
---	---	---

Initialize Array: cria um arranjo n -dimensional, em que cada elemento é inicializado com o valor de **elemento**. Redimensione a função para aumentar a quantidade de dimensões do arranjo de saída. Por exemplo, a função Initialize Array retorna o seguinte arranjo para um **element** de 4, um **dimension size** de 3 e um terminal **dimension size**.

4	4	4
---	---	---

Build Array: concatena vários arranjos ou anexa elementos a um arranjo n -dimensional. Redimensione a função para aumentar a quantidade de elementos no arranjo de saída. Por exemplo, se você concatenar os dois arranjos anteriores, a função Build Array retornará o seguinte arranjo.

7	4	2
4	4	4

Para concatenar as entradas em um arranjo mais longo da mesma dimensão, como mostrado no arranjo a seguir, clique com o botão direito no nó da função e selecione **Concatenate Inputs** no menu de atalho.

7	4	2	4	4	4
---	---	---	---	---	---

Array Subset: retorna uma parte de um arranjo que inicia no **índice** e contém elemento de **comprimento**. Por exemplo, se você utilizar o arranjo anterior como a entrada, a função Array Subset retornará o arranjo a seguir para um **index** de 2 e um **length** de 3.

2	4	4
---	---	---

Index Array: retorna um elemento de um arranjo em **index**. Por exemplo, se você utilizar o arranjo anterior como a entrada, a função Index Array retornará 2 para um **index** de 0. Você também pode utilizar a função Index Array para extrair uma linha ou coluna de um arranjo 2D e criar um subarranjo do arranjo original.

Para fazer isso, ligue um arranjo 2D à entrada da função. Dois terminais **index** estão disponíveis. O primeiro terminal **index** indica a linha e o segundo terminal indica a coluna. Você pode ligar entradas aos dois terminais **index** para indexar um único elemento ou pode ligar apenas um terminal para extrair uma linha ou uma coluna de dados. Por exemplo, ligue o seguinte arranjo à entrada da função.

7	4	2
4	4	4

A função Index Array retorna o seguinte arranjo para um **index (row)** de 0.

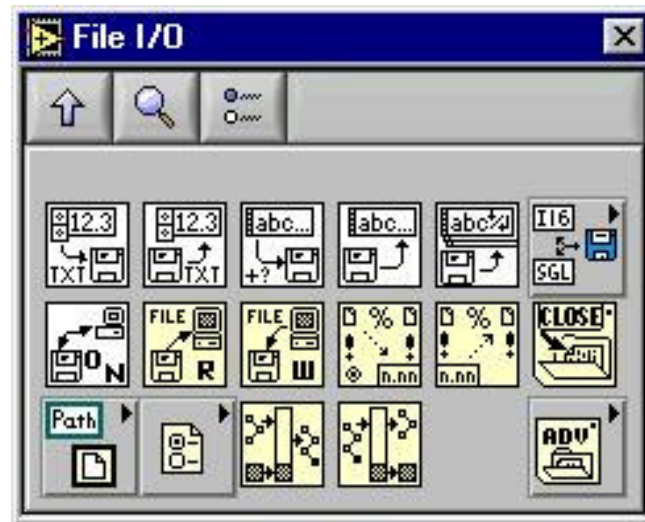
7	4	2
---	---	---

Exercícios

- 1.) *Entre com dados em uma matriz A e em uma matriz B e realize a soma dos elementos*
- 2.) *Entre com dados em uma matriz A (3 x 3) e exiba a somatória da:*
 - i. *Diagonal principal*
 - ii. *Diagonal secundária*
 - iii. *Soma da primeira e ultima linha da matriz*
- 3.) *Preencha duas matrizes randomicamente definidas pelo usuário; caso possa ser feita efetue a multiplicação das duas matrizes gerando uma terceira.*

9ª AULA LABVIEW – PROF. GALE

Arquivos Input/Output



Uma operação comum de I/O de arquivos envolve o seguinte processo:

1. Crie ou abra um arquivo. Indique o local onde um arquivo já existente está ou onde você deseja criar um novo arquivo, especificando um caminho ou respondendo a uma caixa de diálogo para direcionar o LabVIEW até a localização do arquivo. Após a abertura do arquivo, um refnum (número de referência) representa o arquivo;
2. Leia ou grave o arquivo;
3. Feche o arquivo.

Operações básicas de I/O de arquivos:

- **Open/Create/Replace File:** abre ou substitui um arquivo já existente ou cria um novo arquivo. Se o campo **file path** estiver vazio, o VI exibirá uma caixa de diálogo na qual você pode selecionar o arquivo já existente ou o novo arquivo.
- **Read File:** lê dados do arquivo especificado por **refnum** e retorna-os em **data**, em que **count** é a quantidade de dados a ser lida. A leitura começa no local especificado por **pos mode** e **pos offset** e depende do formato do arquivo especificado.
- **Write File:** grava dados no arquivo especificado por **refnum**. A gravação inicia em uma localização especificada em **pos mode** e **pos offset** para arquivos de fluxo de bytes e no final do arquivo para arquivos datalog.
- **Close File:** fecha o arquivo especificado por **refnum**.

Cria um arquivo não existente

tamanho (bytes)

42

data

String

isso é um exemplo feito bem
agora.....

Gravar dados em um Arquivo

Path

C:\teste.txt

new file path

C:\teste.txt

Texto

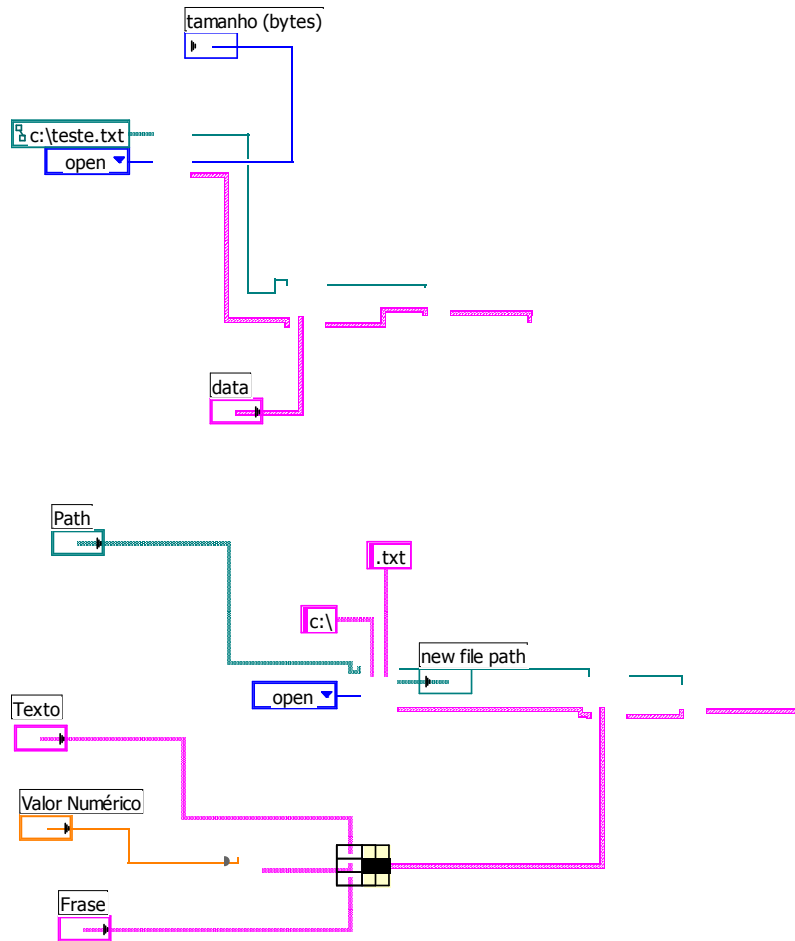
primavera

Valor Numérico

28,00

Frase

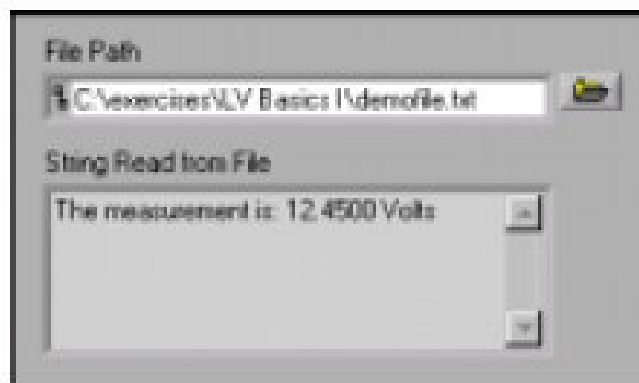
setembro



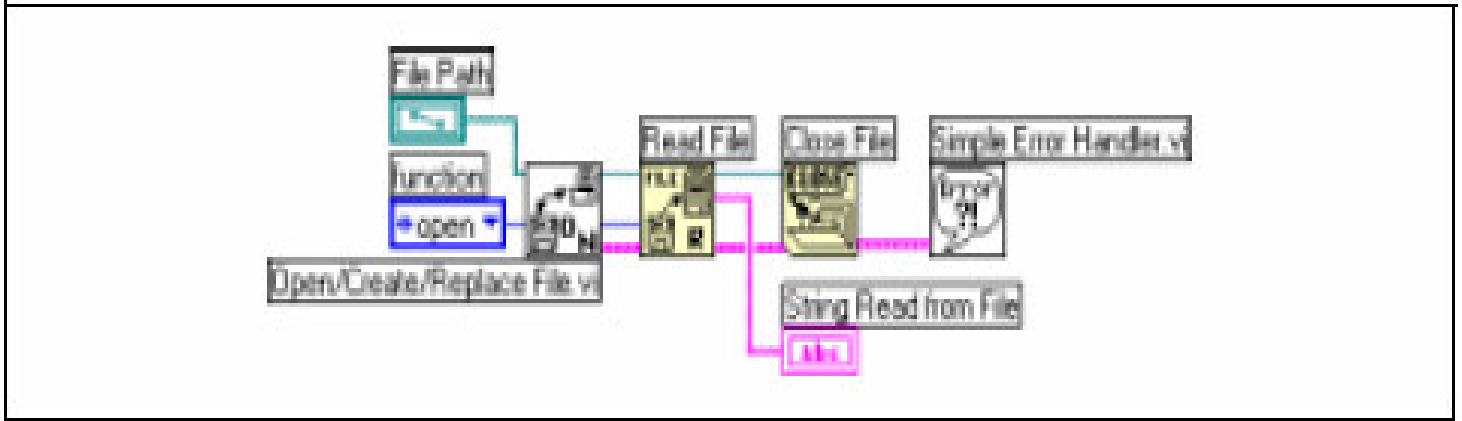
Faça Você

Panel Frontal

1. Abra um novo VI e monte o painel frontal a seguir utilizando o controle de caminho, localizado na paleta **Controls»String & Path**.



2. Monte o diagrama de Blocos



- a. Selecione o VI Open/Create/Replace File, localizado na paleta **Functions»File I/O**. Esse VI exibe uma caixa de diálogo que você utiliza para abrir ou criar um arquivo.
- b. Clique com o botão direito no terminal de entrada **prompt** e selecione **Create»Constant** no menu de atalho para criar a constante **Select Filename**.
- c. Clique com o botão direito no terminal de entrada **function** e selecione **Create»Constant** no menu de atalho. Utilize a ferramenta **Operating** para selecionar **open**.
- d. Selecione a função **Read File**, localizada na paleta **Functions»FileI/O**. Essa função lê os bytes de dados **count** do arquivo, começando no início do arquivo.
- e. Selecione a função **Close File**, localizada na paleta **Functions»FileI/O**. Essa função fecha o arquivo.
- f. Selecione o VI **Simple Error Handler**, localizado na paleta **Functions»Time & Dialog**. Esse VI verificará o cluster de erro e exibirá uma caixa de diálogo se um erro ocorrer.

3. Salve o VI como `File Reader.vi`.

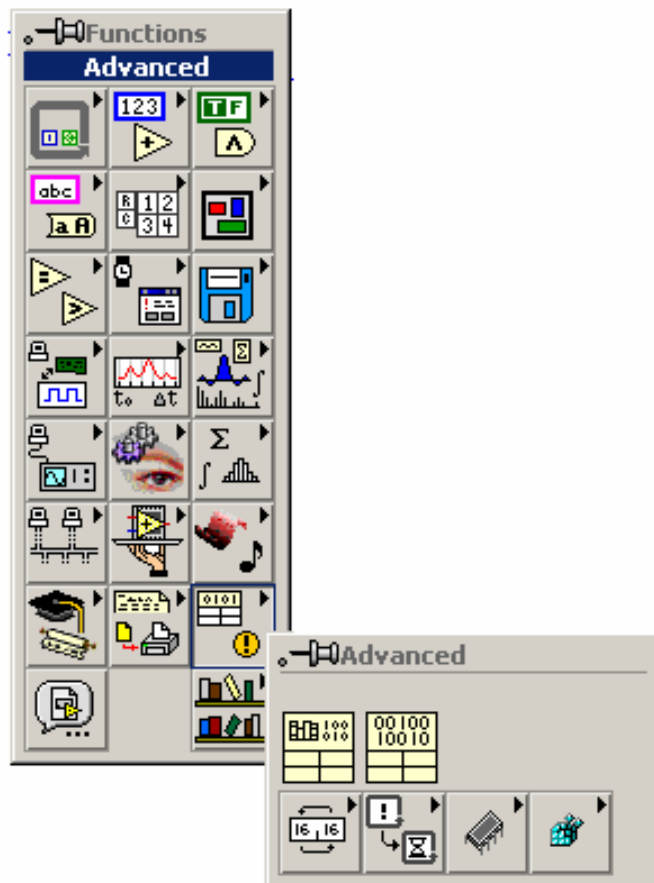
4. Exiba o painel frontal e utilize a ferramenta **Operating** para clicar no botão **Browse** no controle de caminho.

5. Acesse o diretório `demofile.txt` e clique no botão **Open** ou **OK**.

6. Execute o VI. O indicador **String Read from File** exibe o conteúdo do arquivo.

10ª AULA LABVIEW – PROF. GALE

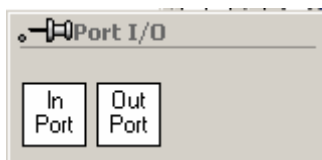
Porta Paralela



Para termos acesso a porta paralela devemos entrar em Functions – Advanced – Port I/O. Encontramos os comandos OutPort (saída de dados) e InPort(Entrada de dados). Se vc utiliza o Windows 98 não terá problema algum ao utilizar estes objetos; caso contrário, se utiliza Windows XP, NT, 2000 terá que baixar um outro componente encontrado no seguinte site:

<http://digital.ni.com/public.nsf/allkb/c80752c53094c296862565f3007ac765>

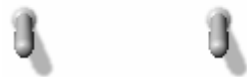
Baixando este arquivo, instale e utilize o ParallelNT; chame-o como uma VI (externa) e funcione sem problemas algum.



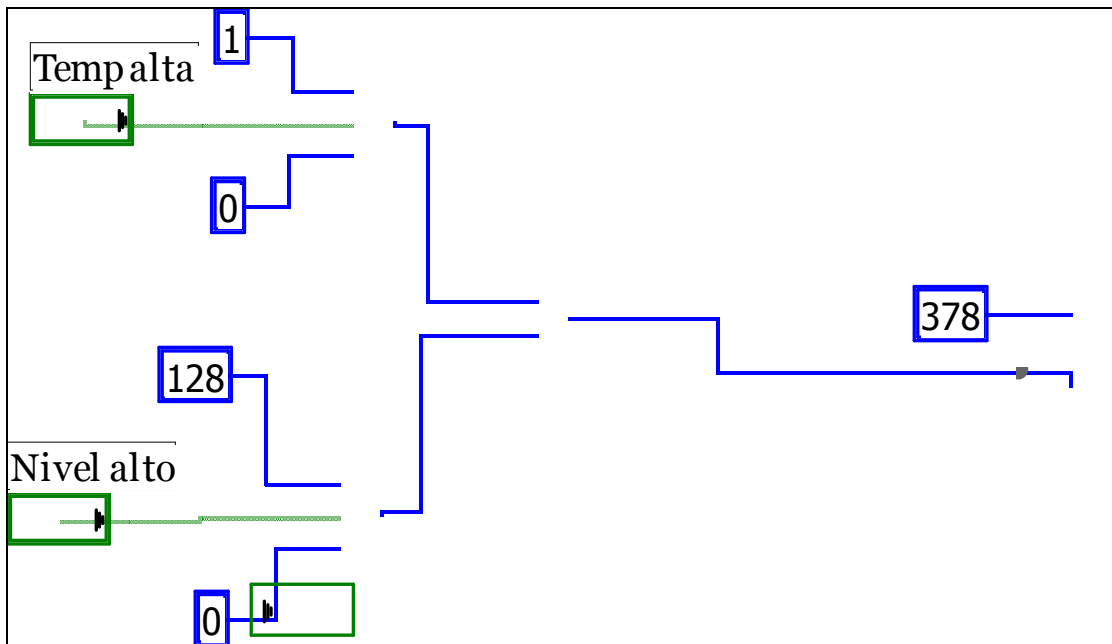
Ao utilizar os componentes observe que o valor da porta de entrada e saída está em decimal, neste caso a porta 0x378 (hexadecimal) corresponde ao valor 888 em decimal.

Observe o exercício a seguir:

Temp alta Nivel alto

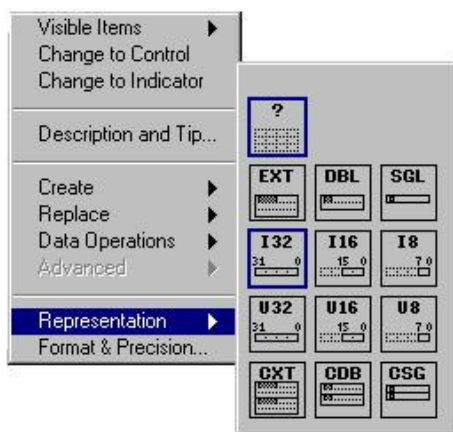


Temos duas entradas de dados que são acionadas manualmente quando o usuário clica sobre elas. Mudando a posição destes botões desejamos enviar um dado para a porta paralela. Observemos o diagrama de Blocos:



Utilizamos um “Select” encontrado na “Comparação” que possibilita enviar um ou outro sinal. Percebe-se que se acionada a chave 1 envia o valor 1 para a porta juntamente com o que está disponível na outra chave, ou seja, se estiver acionada a outra chave (nível alto) será realizada a soma de 128 + valor da primeira chave.

A entrada de dados se dá através do objeto InPort, utilizando como porta de entrada 0x379 (hexadecimal)

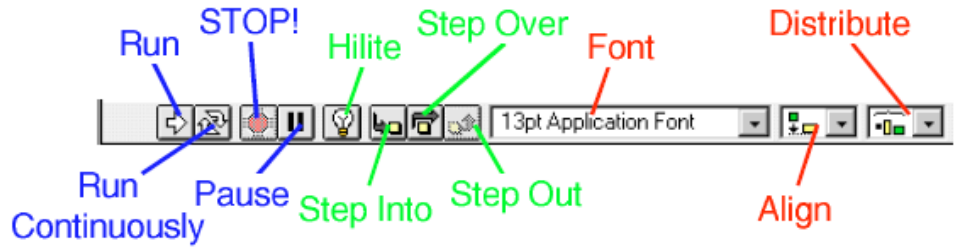


Se tiver interesse em alterar o tipo de representação do dado, ou seja, de real para inteiro, clique sobre o objeto com o botão direito do mouse e escolha o melhor tipo desejado.



One-Page Reference Guide

(J. R. Drummond, 1996)



General Editing

Pop-Up	Right-click
Rotate Tools	tab
Operating ⇌ Positioning (front panel) Positioning ⇌ Wiring (block diagram)	space
Move item one (several) pixel(s)	(Shift)-arrow
Increase/Decrease Controls faster	shift-click
add to ring control	shift-enter
Toggle help on/off	Ctrl-h
duplicate using positioning Tool	Ctrl-drag

Diagram Editing

Create subVI when repeating part of a block diagram	Select items Edit>> Create SubVI from Selection
Create a constant/control/indicator	Pop-up on function terminals Create Constant/Control/Indicator
Create a constant	drag a front panel control to block diagram
Create a control	drag block diagram constant to front panel
List errors	Pop-up on bad wires, List Errors -or- click on broken arrow icon
Open VI front panel (block diagram)	(Ctrl)-Double-click
Delete bad wires	Ctrl-b

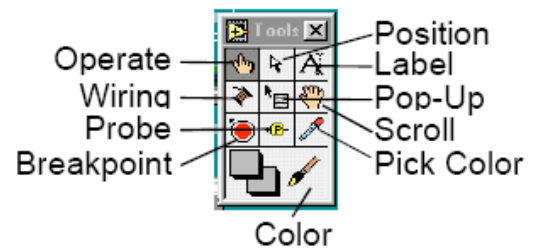
Help

Show Help	Context-Sensitive (Ctrl-h)
Lock Help	Hold that help! (Ctrl-Shift0-L)
Simple Help	Controls amount of help shown
Online Reference	Complete documentation (Ctrl-?)
Online Help	Customised Online reference for this VI
About LabVIEW	Software rev. & serial #

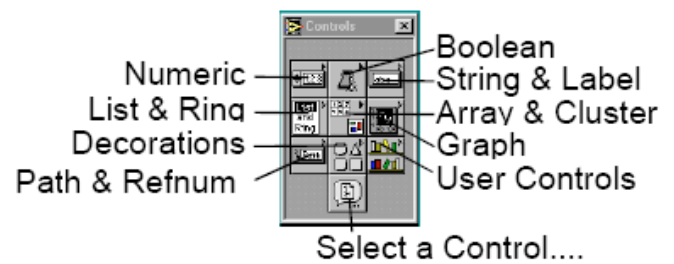
Wiring Techniques

Tack down a wire	Click
Un-tack last point	Ctrl-Click
Switch wire direction	Space
Highlight segment/wire/net	Click/Double-click/triple-click
Tack end of wire (temporary)	Double-click

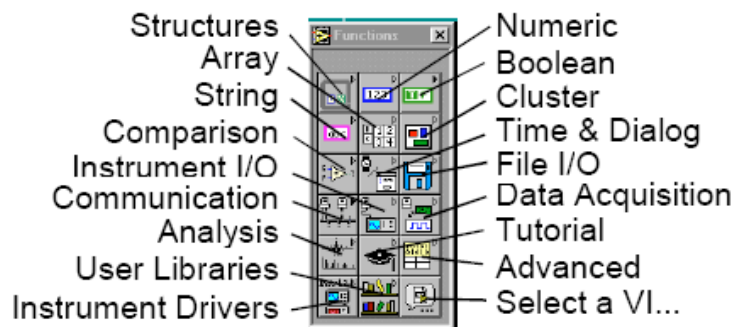
Tools



Controls



Functions





LabVIEW Documentation and Help Resources

Context Help

When you move the cursor over objects on the front panel or block diagram, the **Context Help** window displays basic information about each object. Select **Help»Show Context Help** to display the **Context Help** window.

LabVIEW Help

Includes reference information for each palette, menu, tool, dialog box, and built-in VI or function. The *LabVIEW Help* also includes step-by-step instructions for using LabVIEW features. Select **Help»VI, Function, & How-To Help** to access the *LabVIEW Help*.

LabVIEW Bookshelf

Contains links to all the LabVIEW documentation in electronic format. Use the *LabVIEW Bookshelf* to search PDF versions of all the LabVIEW manuals and Application Notes. Select **Help»Search the LabVIEW Bookshelf** to access the *LabVIEW Bookshelf*.

LabVIEW Examples

LabVIEW includes hundreds of example VIs you can use and incorporate into your own VIs. You can modify an example to fit your application, or you can copy and paste from one or more examples into your own VI. Browse or search the example VIs by selecting **Help»Find Examples**.

LabVIEW Web Resources

Includes a list of links to online National Instruments resources. Select **Help»Web Resources** to access the LabVIEW resources at ni.com. Resources available to you include the following:

Products and Services

- Upgrades
- Training courses
- Seminars and user groups

Solutions

- Systems integrators
- Customer solutions

Support

- LabVIEW support
- Manuals

NI Developer Zone

- Discussion forums
- Example code

ni.com/labviewzone

Keyboard Shortcuts

Refer to the LabVIEW Help to access this list of keyboard shortcuts.

Object/Movement

Shift-click	Selects multiple objects; adds object to current selection.
N++ (arrow keys)	Moves selected objects one pixel at a time.
Shift-N++	Moves selected objects several pixels at a time.
Shift-click (drag)	Moves selected objects in one axis.
Ctrl-click (drag)	Duplicates selected object.
Ctrl-Shift-click (drag)	Duplicates selected object and moves it in one axis.
Shift-resize	Resizes selected object while maintaining aspect ratio.
Ctrl-resize	Resizes selected object while maintaining center point.
Ctrl-drag a rectangle	Inserts more working space on the front panel or block diagram.
Ctrl-A	Performs last alignment operation on objects.
Ctrl-D	Performs last distribution operation on objects.
Double-click open space	Places free label on the front panel or block diagram if automatic tool selection is enabled.
Ctrl-mouse wheel	Scrolls through subdiagrams of a Case, Event, and Stacked Sequence structure.

Navigating the LabVIEW Environment

Ctrl-E	Displays block diagram or front panel.
Ctrl-G	Enables or disables alignment grid. On French keyboards, press the <Ctrl-;> keys.
Ctrl-J	Maximizes and restores window.
Ctrl-T	Tiles front panel and block diagram windows.
Ctrl-F	Finds objects or text.
Ctrl-G	Searches VIs for next instance of object or text.
Ctrl-Shift-G	Searches VIs for previous instance of object or text.
Ctrl-Shift-F	Displays Search Results window.
Ctrl-Tab	Cycles through LabVIEW windows.
Ctrl-Shift-Tab	Cycles backward through LabVIEW windows.
Ctrl-I	Displays VI Properties dialog box.
Ctrl-L	Displays Error Use window.
Ctrl-Y	Displays History window.
Ctrl-D	Redraws Hierarchy window.
Ctrl-A	Shows all VIs in Hierarchy window.
Enter *	Finds next node that matches the search string.
Shift-Enter *	Finds previous node that matches the search string.

* After initiating a search by typing in Hierarchy window

Debugging

Ctrl-I	Steps into node.
Ctrl-→	Steps over node.
Ctrl-t	Steps out of node.

File Operations

Ctrl-N	Creates new VI.
Ctrl-O	Opens existing VI.
Ctrl-W	Closes VI.
Ctrl-S	Saves VI.
Ctrl-P	Prints the window.
Ctrl-Q	Quits LabVIEW.

Basic Editing

Ctrl-Z	Undoes last action.
Ctrl-Shift-Z	Redoes last action.
Ctrl-X	Cuts an object.
Ctrl-C	Copies an object.
Ctrl-V	Pastes an object.

Help

Ctrl-H	Displays Context Help window.
Ctrl-Shift-L	Locks Context Help window.
Ctrl-? or F1	Displays LabVIEW Help.



Note The <Ctrl> key in these shortcuts corresponds to the (Mac OS) <Option> or <Command> key, (Sun) <Meta> key, or (Linux) <Alt> key.

Tools and Palettes

Ctrl	Switches to next most useful tool.
Shift	Switches to Positioning tool.
Ctrl-Shift over open space	Switches to Scrolling tool.
Spacebar *	Toggles between two most common tools.
Shift-Tab *	Enables automatic tool selection.
Tab *	Cycles through tools if you disabled automatic tool selection by clicking the Automatic Tool Selection button. Otherwise, enables automatic tool selection.
F4++	Navigates temporary Controls and Functions palettes.
Enter	Navigates into temporary palette.
Esc	Navigates out of temporary palette.

* If automatic tool selection is disabled

SubVIs

Double-click subVI	Displays subVI front panel.
Ctrl-double-click subVI	Displays subVI block diagram.
Drag VI icon to block diagram	Places that VI as a subVI on the block diagram.
Shift-drag VI icon to block diagram	Places that VI as a subVI on the block diagram with constants wired for controls that have non-default values.
Ctrl-right-click block diagram and select VI	Opens front panel of that VI from the Functions palette.

Execution

Ctrl-R	Runs VI.
Ctrl-. *	Aborts VI.
Ctrl-M	Changes to run or edit mode.
Ctrl-Run button	Recompiles current VI.
Ctrl-Shift-Run button	Recompiles all VIs in memory.
Ctrl-I *	Moves key focus inside an array or cluster.
Ctrl-O *	Moves key focus outside an array or cluster.
Tab *	Navigates the controls or indicators according to tabbing order.
Shift-Tab *	Navigates backward through the controls or indicators.

* While VI is running

Wiring


















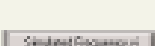
Ctrl-B	Removes all broken wires.
Esc, right-click, or click terminal	Cancels a wire you started.
Single-click wire	Selects one segment.
Double-click wire	Selects a branch.
Triple-click wire	Selects entire wire.
A	Disables automatic wire routing temporarily.
Double-click (while wiring) spacebar	Tacks down wire without connecting it.
Shift-click	Toggles automatic wiring while moving objects.
Ctrl-click	Undoes last point where you set a wire.
Ctrl-click Input on function with two inputs	Switches the two Input wires.
spacebar	Switches the direction of a wire between horizontal and vertical.

Text






























Double-click	Selects single word in string.
Triple-click	Selects entire string.
Ctrl-→	Moves forward in string by one word.
Ctrl-←	Moves backward in string by one word.
Home	Moves to beginning of current line in string.
End	Moves to end of current line in string.
Ctrl-Home	Moves to beginning of entire string.
Ctrl-End	Moves to end of entire string.
Shift-Enter	Adds new items when entering items in enumerated type controls and constants, ring controls and constants, or Case structures.
Esc	Cancels current edit in a string.
Ctrl-Enter	Ends text entry.
Ctrl=	Increases current font size.
Ctrl-	Decreases current font size.
Ctrl-F	Displays Font dialog box.
Ctrl-1 *	Displays Application font.
Ctrl-2 *	Displays System font.
Ctrl-3 *	Displays Dialog font.
Ctrl-4 *	Displays current font.


* In Font dialog box

Editing, Execution, and Debugging Tools

 Show Context Help Window —Displays the Context Help window.	 Resize Objects —Resizes multiple front panel objects to the same size.
 Run —Runs the VI.	 Reorder —Reorders objects if they overlap, including moving forward and moving backward.
 Broken Run —Indicates that the VI contains errors. Click the button to list errors.	 Highlight Execution —Shows the flow of data through the block diagram.
 Run Continuously —Runs the VI until you abort or pause it.	 Step Into —Steps into a node, which blinks when ready to execute.
 Abort Execution —Stops the VI immediately.	 Step Over —Executes a node and pauses at the next node.
 Pause —Pauses or resumes a running VI.	 Step Out —Steps out of a node and continues to the next node.
 Align Objects —Aligns objects along axes.	 Breakpoint Tool —Sets breakpoints on nodes to pause execution.
 Distribute Objects —Spaces objects evenly.	 Probe Tool —Creates probes on wires to display intermediate values in a running VI.
 Text Settings —Changes the attributes of text in the VI.	 Call Chain —Lists all the callers from the top-level VI to the opened subVI.

Terminal Data Types

Signed Integers (i)	8-bit 	Boolean (FALSE) 	 Waveform
	16-bit 	String (empty string) 	 Digital waveform
	32-bit 	Path (<Not A Path>) 	 Digital
Unsigned Integers (u)	8-bit 	Reference number 	 I/O name
	16-bit 	Enumerated type 	 Time stamp
	32-bit 	Cluster	Numeric 
Floating-Point (f)	Single 	Mixed data 	 Dynamic
	Double 	Array	 P closure
	Extended 	1D 	 Variant
		2D 	
Complex Floating-Point (c + 0.0)	Single 		
	Double 		
	Extended 		

 Note: Default values appear in ().