

# Controle Digital

Notas de Aula de Teoria e de Laboratório

Manoel L. Aguiar

19 de Julho de 2016

## Conteúdo

<b>I</b>	<b>PRÁTICAS DE CONTROLE DIGITAL</b>	<b>1</b>
<b>1</b>	<b>ORGANIZAÇÃO DO LABORATÓRIO</b>	<b>3</b>
1.1	Introdução . . . . .	3
1.2	Realização de atividades e experimentos . . . . .	3
1.3	Descrição dos equipamentos do Laboratório . . . . .	4
1.4	Aquisição ou Entrada Digital . . . . .	4
1.4.1	Entrada por Conversor A/D . . . . .	4
1.4.2	Entrada por Porta Digital . . . . .	5
1.5	Saída Digital . . . . .	6
1.5.1	Saída por Conversor D/A . . . . .	6
1.5.2	Saída por Porta Digital . . . . .	7
1.6	Estrutura Geral do <i>hardware</i> do <i>LabAcquisition</i> . . . . .	8
1.7	Características Específicas do <i>Hardware</i> no Laboratório . . . . .	9
<b>2</b>	<b>INTRODUÇÃO AO LabVIEW</b>	<b>11</b>
2.1	Objetivos . . . . .	11
2.2	Generalidades . . . . .	11
2.3	Iniciando o LabVIEW . . . . .	13
2.4	Operação Básica LabVIEW . . . . .	15
2.5	Atividades Introdutórias . . . . .	16
<b>3</b>	<b>DIAGRAMAS E PAINÉIS BÁSICOS</b>	<b>19</b>
3.1	Objetivos . . . . .	19
3.2	Generalidades . . . . .	19
3.3	Atividade de aplicação . . . . .	21
<b>4</b>	<b>ESTRUTURAS NÃO TEMPORIZADAS</b>	<b>26</b>
4.1	Objetivos . . . . .	26
4.2	Generalidades . . . . .	26
4.3	Estrutura SEQUENCE . . . . .	27
4.4	Estrutura CASE ( Lógico e Múltiplo) . . . . .	27
4.5	Estrutura FOR . . . . .	29
4.6	Estrutura WHILE . . . . .	30
4.7	Estrutura FORMULA - NODE . . . . .	30
4.8	Atividade de aplicação . . . . .	31

<b>5 SAÍDAS GRÁFICAS E PARA ARQUIVO NÃO TEMPORIZADAS</b>	<b>33</b>
5.1 Objetivos . . . . .	33
5.2 Generalidades. . . . .	33
5.3 Saídas gráficas . . . . .	34
5.3.1 Saída <i>Chart</i> . . . . .	34
5.3.2 Saída tipo <i>Graph</i> . . . . .	35
5.4 Saída tipo <i>Graph XY</i> . . . . .	36
5.5 Manipulação de dados de arquivos ou para arquivos . . . . .	37
5.6 Interagindo com o Matlab por meio do LabVIEW : <i>MATLAB-Script</i>	39
5.7 Atividades de aplicação . . . . .	40
<b>6 ESTRUTURAS TEMPORIZADAS</b>	<b>43</b>
6.1 Objetivos . . . . .	43
6.2 Generalidades . . . . .	43
6.2.1 Opção <i>Tick Count</i> (ms) . . . . .	44
6.2.2 Opção <i>Wait</i> (ms) . . . . .	45
6.2.3 Opção <i>Wait Until Next</i> (ms) <i>Multiple</i> . . . . .	45
6.2.4 Opção <i>Time Delay</i> . . . . .	46
6.2.5 Opções de Data e Manipulação . . . . .	46
6.3 Atividade de aplicação . . . . .	46
<b>7 TRATAMENTO DE SINAIS ANALÓGICOS - ENTRADA E SAÍDA POR <i>HARDWARE</i></b>	<b>48</b>
7.1 Objetivos . . . . .	48
7.2 Generalidades . . . . .	48
7.2.1 Entrada Analógica e VIs de leitura analógica . . . . .	50
7.2.2 Saída analógica e VIs de escrita . . . . .	53
7.3 Atividades de aplicação . . . . .	54
<b>8 VERIFICAÇÃO EXPERIMENTAL DO TEOREMA DA AMOSTRAGEM</b>	<b>55</b>
8.1 Objetivos . . . . .	55
8.2 Introdução e generalidades . . . . .	55
8.3 Atividades de aplicação . . . . .	56
<b>9 PRÁTICAS DOS CONTROLADORES DIGITAIS</b>	<b>58</b>
9.1 Introdução . . . . .	58

9.2	Descrição do processo . . . . .	59
9.3	Relação de Grupos e Critérios de Desempenho . . . . .	59
9.4	Etapas de Avaliação . . . . .	62
9.5	Descrição da placa do Processo . . . . .	66

SEL359-2015 MLAgular

Parte I

**PRÁTICAS DE CONTROLE  
DIGITAL**



# Laboratório 1

## ORGANIZAÇÃO DO LABORATÓRIO

### 1.1 Introdução

Por meio das aulas de laboratório visa-se o estudo e aplicação de dispositivos e componentes para a realização de controladores digitais, a introdução de técnicas de controle digital, de ferramentas de análise e de implementação destas técnicas de controle bem como, a verificação experimental dos conceitos teóricos abordados nas Partes 1 e 2.

### 1.2 Realização de atividades e experimentos

O andamento do curso e respectivas atividades têm início com uma série de aulas com grande conteúdo expositivo, através das quais serão discutidas informações sobre tipos e modos de operação de sensores, atuadores, unidades de processamento e ferramentas de simulação/análise e implementação.

Nas diversas fases iniciais do curso serão propostas atividades de simulação com ferramentas de análise e também com o pacote de *software* a ser empregado na implementação de controle digital nas fases finais do curso.

Como forma de finalização e sedimentação dos conceitos teóricos e outros de caráter experimental, uma série de experimentos envolvendo diversos tipos de controladores digitais é proposta como forma de avaliação do aproveitamento do curso. Nesta fase, relativa ao último mês de aulas (5 últimas aulas), cada grupo deverá desenvolver análises teóricas, de simulação e de implementação de controladores estudados na teoria e executar um relatório pertinente dos resultados, bem como das etapas de realização dos

controladores.

### 1.3 Descrição dos equipamentos do Laboratório

Para a realização das práticas de laboratório de controle digital são disponibilizados alguns equipamentos para uso didático, mas que também são largamente empregados em sistemas industriais de automação, supervisão e controle de processos em grande escala.

O sistema empregado neste caso é o LabVIEW e o LabAcquisition da empresa National Instruments. Enquanto que o LabVIEW é um pacote de *software* muito versátil e dirigido para monitoramento e controle de processos, o *LabAcquisition* configura módulo de *hardware* que é gerenciado pelo programa LabVIEW.

No âmbito do laboratório este equipamento será configurado para controle de processos simples e locais. Normalmente os Controladores de Processos determinam e monitoram as grandezas de referência para os controladores locais em um sistema de controle distribuído ou rede.

Uma visão geral dos procedimentos e formas de uso do *software* LabVIEW é dada no arquivo “Apostila do LabVIEW” disponibilizada em arquivo PDF. Vários dos tópicos abordados na apostila serão estudados e implementados em atividades no laboratório.

Na sequência serão apresentados alguns aspectos da composição e arranjo operacional do sistema de *hardware* disponível, juntamente com a introdução simplificada de procedimentos e tecnologia de sensores e atuadores digitais.

### 1.4 Aquisição ou Entrada Digital

Por aquisição ou entrada digital entendem-se os procedimentos de fornecer ao processador uma grandeza física em forma digital para ser processada. Uma das formas é através de conversão Analógica-Digital A/D quando o sinal original é de natureza analógica ou contínua ou através de portas digitais quando a grandeza física é de natureza digital ou amostrada em níveis discretos.

#### 1.4.1 Entrada por Conversor A/D

Os procedimentos para realização aquisição ou entradas digitais de sinais analógicos envolvem os conceitos de amostragem e digitalização ou conversão da forma analógica para a forma digital.

Neste aspecto entram em cena os dispositivos de conversão A/D que podem envolver uma grande diversidade de tecnologia e configuração de operação. Conversores A/D

podem atender aos requisitos de uma única entrada de sinal em 8 Bits de discretização até requisitos de múltiplas entradas em 16 Bits em um único “Chip”.

Discretização em 8 Bits significa  $2^8 = 256$  diferentes níveis de discretização, enquanto que 16 Bits implicam em  $2^{16} = 65536$  níveis de discretização em amplitude. Considerando-se que os “Chips” comerciais A/D operam com sinais de entrada variando entre 0 a 5V, ou 0 a 10V para configurações unipolares e entre -5 a 5V ou -10 a 10V para configuração bipolar, os níveis de discretização podem variar entre os níveis de 78,125 mV a 76  $\mu$ V dependendo do caso. A configuração de operação pode ser pré definida para um determinado “Chips” ou pode ser estabelecida por meio de programação em tempo real nos “Chips” A/D programáveis. Alguns “Chips” possuem ainda tecnologia para se configurar os níveis de ganho analógico necessário nas entradas de sinais.

As taxas de amostragem ou intervalos entre aquisições são determinados basicamente pelo usuário final em função da aplicação em questão, desde que obedecendo aos limites operacionais do “Chips” em uso. Neste aspecto existem “Chips” A/D’s que podem operar a uma taxa de amostragem de até centenas de MHz. Um aspecto que também determina a taxa de amostragem é a capacidade do processador principal de gerenciar dados a uma elevada frequência.

### 1.4.2 Entrada por Porta Digital

Neste caso, assume-se ou considera-se que o sinal se encontra ou é de natureza digital. Neste aspecto o sinal pode ser naturalmente discreto e que é diretamente quantizado em níveis lógicos, como por exemplo, estados lógicos de chaves multicanal ou de estado simples, sinalizadores ou sensores de estado. Outros tipos de sinais digitais possuem característica pulsante com frequência determinada ou não.

Para sinais de estado lógico, o meio de entrada convencional é por meio das chamadas portas I/O, que somente processam sinais digitais e também podem operar em dois sentidos, ou seja, também podem ser portas de saída digital como será visto adiante. Portas do tipo I/O são acessadas via uma estratégia sequencial ou temporizada definida na estrutura do programa em uso.

Para o caso de chaves ou dispositivos de estado lógico discreto, os sinais podem ainda entrar sem que sejam previstos, tais como aqueles que provocam interrupções na CPU do processador principal, mas também podem ser lidos de acordo com uma estratégia de *software*. Neste último caso, portas lógicas de n-Bits são disponibilizadas ao processador e que então são lidas de acordo como necessário. Quando um sinal discreto não se encontra em um nível de tensão adequado aos estados lógicos da unidade de processamento digital, este deve ser compatibilizado antes de ser conectado aos

pontos de entrada digital.

Sinais do tipo pulsante, novamente se não estiverem em níveis de amplitude adequados aos níveis de estado lógico, procede-se uma conformação de níveis. Desta forma, os sinais pulsantes podem ser processados em unidade que promove contagem de pulsos ou contadores, tal que o valor da contagem possa ser lido, periodicamente ou não pelo processador digital.

A leitura do contador pode ser utilizada para se obter o valor da frequência do sinal, da largura do pulso ou simplesmente a taxa de variação de contagem de pulsos, por exemplo na medição de velocidade angular de motores elétricos através de *encoder* incremental.

## 1.5 Saída Digital

Por saída digital entendem-se os procedimentos de conduzir um valor ou grandeza na forma digital no ambiente da CPU do processador para o ambiente externo à CPU, normalmente para os módulos periféricos de saída.

Periféricos de saída típicos dos sistemas de controle são circuitos eletrônicos que providenciam isolamento elétrica e conversão da entidade digital em níveis de tensão. Este valor de tensão é então conduzido a um bloco de saída pertinente que executa alguma ação em um dispositivo externo. Novamente tal como no caso de entrada digital, pode-se realizar a saída digital por conversão D/A (digital para analógica) ou por porta digital, sendo que esta última opção pode ser temporizada ou não.

### 1.5.1 Saída por Conversor D/A

Para este caso são empregados os “Chips” de conversores D/A que operam também com várias tecnologias e promovem uma conversão do nível lógico de n-Bits em um sinal de tensão proporcional. Também é usual que os conversores D/A sejam unipolar ou bipolar, obedecendo também aos níveis de amplitude de saída descritos no caso dos A/D e apresentam resolução em Bits semelhante aos A/D.

Na operação do conversor D/A, uma estratégia definida por *software* transfere ao “Chips” ou unidade D/A um valor digital de n-Bits que é convertido em tensão “dentro” deste bloco ou módulo. O sinal analógico é então disponibilizado em um pino do “Chip” ou do módulo de saída. A característica de resposta em frequência dos conversores D/A é função da tecnologia empregada na sua construção, podendo atingir 18 Bits de resolução com 10 ns de tempo de transição ou então menor do que 10 ns para resoluções entre 10 e 8 Bits.

Normalmente o nível de potência do sinal analógico de saída é o mínimo possível por

razões construtivas dos “Chips” em tamanho reduzido. Desta forma é sempre usual a propagação do sinal de saída através de um “*buffer*” ou módulo de isolamento que também provém um certo nível de potência para os circuitos eletrônicos subsequentes.

Se o sistema a ser acionado por este sinal for de potência ainda mais elevada, o sinal do D/A deve ser novamente amplificado ou servir de comando para um bloco de potência, por exemplo um circuito PWM analógico ou um módulo de disparos para SCR’s de potência.

### 1.5.2 Saída por Porta Digital

As portas de saída digitais podem ser basicamente de dois tipos: as temporizadas e não temporizadas, associadas a “*clock’s*” externos ou internos e que podem ou não ser sincronizados.

#### Porta de Saída não Temporizada

Uma porta de saída digital não temporizada é simplesmente um barramento de dados “*bufferizado*” disponível como porta I/O digital, onde o sinal digital é convertido Bit-a-Bit em um nível de tensão de acordo com os níveis de estado lógico. Neste caso cada Bit de saída irá comutar (ligar ou desligar) o estado de um dispositivo de saída específico através de relés eletrônicos ou mecânicos.

Neste caso, as ações de comando e controle são do tipo estanque em que o módulo de saída é apenas comutado em estado. O uso da porta de saída digital também é possível em forma de palavra digital em que os n-Bits da porta comanda um módulo de saída, por exemplo em um decodificador de segmentos de *display’s* digitais ou conjuntos de led’s de sinalização.

#### Porta Digital Temporizada

Portas digitais temporizadas são na realidade dispositivos periféricos que são acessados de forma frequente pelo *software*, o qual disponibiliza dados em forma digital e que determinam o modo de operação destes periféricos. Elementos típicos desta classe são os *Timers* e Contadores que, devidamente programados executam ou sintetizam formas de ondas de tensão, as quais são disponibilizadas aos dispositivos finais de saída. Nos módulos de saída, estas formas de ondas são amplificadas ou comanda módulos de potência ou ainda são recombinaadas para gerar outros diferentes padrões de formas de onda.

Uma aplicação típica é a geração de PWM digital com o uso de *Timers* programáveis, cujos sinais pulsantes operam diretamente ou através de isoladores ópticos as bases de semicondutores de potência.

## 1.6 Estrutura Geral do *hardware* do *LabAcquisition*

Para os propósitos de controle e acionamento digital de sistemas, uma devida estrutura de *hardware* deve ser desenvolvida e, normalmente é definida pelo usuário. O usuário deve então projetar sua estrutura de forma a se obter um *hardware* dedicado ou obter no mercado um conjunto de sub-sistemas necessários e configurá-los adequadamente.

Os diversos fornecedores e fabricantes de sistemas digitais para controle e acionamento de processos possuem uma grande variedade de módulos e em muitos casos são intercambiáveis entre diferentes marcas e também podem ser programadas por uma grande variedade de *software* de apoio/gerenciamento disponível no mercado.

No caso do sistema disponível no laboratório, trata-se de um conjunto *hardware/software* da empresa *National Instruments*, o qual é designado como marca da empresa por *LabAcquisition/LabVIEW*.

A estrutura geral do *hardware* específico em uso no laboratório pode ser consultada em detalhes no arquivo NI.PCIe6321.PDF e está disponível no “*link*” Disciplinas *On-Line*. O nome do arquivo se refere ao modelo da placa de aquisição em questão, que possui 16 entradas analógicas (A/D) de 12 Bits a 100KHz de amostragem, 2 canais de saídas analógicas (D/A) de 12 Bits, porta digital I/O TTL de 8 Bits (linhas) e 2 *timers/counters Up/Down* de 24 Bits. Um diagrama de blocos desta estrutura geral pode ser encontrada no arquivo citado e é vista na Fig.1.1.

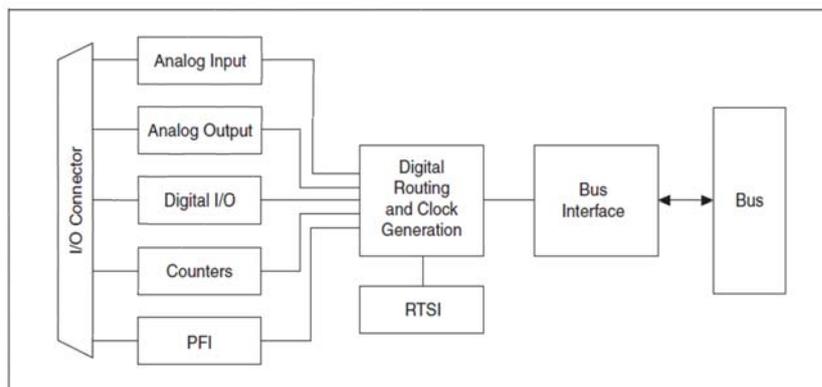


Figura 1.1: Estrutura geral da composição do *hardware* NI PCIe 6321

Este modelo de placa é alocada no barramento *PC-Express* de microcomputadores e é configurada pelo *driver* DAQ\_MX da *National Instruments*. A placa disponibiliza todos os módulos de entrada e saída por meio de um cabo de 68 pinos que é conectado a uma placa externa específica contendo conectores tipo KRE. Nesta placa estão disponíveis todos os sinais de entrada e de saída relativos ao módulo de aquisição. Dois

### 1.7. CARACTERÍSTICAS ESPECÍFICAS DO HARDWARE NO LABORATÓRIO

sinais de entrada analógica e dois sinais de saída analógicas são direcionados para o Rack contendo um *backplane* padrão 5B tipo 5B101. Os sinais de entrada analógicas são isolados por meio dos módulos 5B41. Estes módulos operam na faixa de  $\pm 10V$  e convertem esta faixa de tensões para  $\pm 5V$  que são conduzidos para a placa de aquisição. Esta configuração de distribuição de sinais é indicada na figura 1.2.

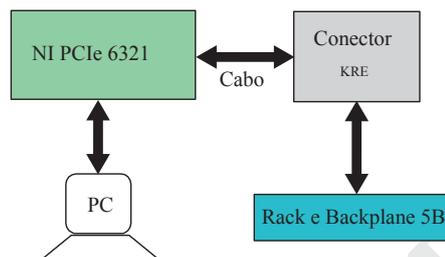


Figura 1.2: Disposição dos módulos de distribuição de sinais

O módulo 5B01 da figura 1.2 é um “*Backplane*” onde são fixados elementos isoladores da classe 5B41-03 que podem configurar ou adaptar sinais analógicos de entrada e também de saída (portas 13 e 14). Os elementos 5B são padrão e podem ser simples isoladores até complexos circuitos de conformação de sinais como amplificadores/atenuadores, filtragem, etc.

No módulo de conectores KRE da figura 1.2 são disponíveis diversos outros canais de entrada e saídas analógicas, bem como muitos outros de tipos de sinais I/O digitais, tais como indicados na figura 1.3.

Nas aulas seguintes serão abordados vários tópicos relacionados ao *software* LabVIEW que como veremos, mantém estreita relação com os tópicos de *hardware* aqui descritos. O LabVIEW pode funcionar independente do *hardware* de aquisição. Neste caso só é possível a realização de simulações com a linguagem G do LabVIEW. Para se entender as características básicas da linguagem G, encontra-se disponível no *link* Disciplinas *On-Line* um arquivo designado Apostila do LabVIEW que contém uma descrição básica dos procedimentos de programação com o LabVIEW.

### 1.7 Características Específicas do *Hardware* no Laboratório

No caso do *hardware* do *LabAcquisition* instalado no laboratório de controle, algumas configurações de operação foram definidas enquanto da instalação do *hardware* e do *software*.



## Laboratório 2

# INTRODUÇÃO AO LabVIEW

### 2.1 Objetivos

Iniciar estudo de operação/utilização do programa LabVIEW como ferramenta de instrumentação virtual e controle dedicado de processos.

### 2.2 Generalidades

O programa LabVIEW é de propriedade da empresa National Instruments e foi inicialmente desenvolvido como ferramenta de *software* de apoio para gerenciamento/supervisão de módulos de circuitos programáveis e de aquisição de dados desenvolvidos pela própria empresa.

O nome LabVIEW provem da designação:

**L**ab oratory  
**V**irtual  
**I**nstruments  
**E**ngineering  
**W**orkbench

A primeira versão foi lançada em 1986 e até 1991 só rodava em microcomputadores da Macintosh(Apple). Em 1993 surgiu a primeira versão para PCs e em 1992 e também passou a rodar em sistemas SUN. A primeira versão para operação em tempo real só foi lançada em 1999. As versões mais atuais manipulam sistemas associados à

Internet, sistemas distribuídos em plataformas *multicore* e FPGAs com algoritmos de processamento paralelo.

O desenvolvimento e aprimoramento continuado deste *software* proporcionou que o mesmo se tornasse referência em termos de *software* de gerenciamento/supervisão de circuitos ou equipamentos programáveis.

No laboratório está disponível o “*release*” ou versão 2013 que dispõe de todos os recursos necessários para o devido andamento das atividades previstas em aula.

Informações básicas do programa LaVIEW são disponibilizadas no arquivo Apostila de LabVIEW disponibilizado no link Disciplinas *On-Line*. Demais ou informações mais detalhadas podem ser obtidas na Internet no site <http://www.ni.com/LabVIEW/> . Na versão instalada, várias opções de “*Help*” são disponíveis em menus do próprio programa. O Programa da versão 2013 dever ser acessado na janela do DESKTOP através do ícone da respectiva versão, Fig. 2.1.



Figura 2.1: Icon do LabVIEW

Todo arquivo gerado no LabVIEW é designado como Instrumento Virtual ou VI da notação em inglês e terá sempre a extensão ( *.VI* ). Um instrumento virtual consiste em uma sequência de comandos de programação em uma linguagem específica que determina a ação de aquisição/geração de sinais em uma atividade específica.

O programa LabVIEW disponibiliza portanto uma série de comandos em linguagem gráfica, chamada interface de usuário, tal que se implemente uma enorme diversidade de instrumentos de medição ou geração de sinais. Outra importante característica do LabVIEW é a possibilidade de se gerar painéis ou mecanismos gráficos de entrada e saída dos dados adquiridos ou supridos ao Instrumento Virtual.

Através de combinação entre aquisição geração de sinais e dos painéis de entrada e saída, obtém-se então os mais diversos dispositivos de comando e ou controle de processos.

## 2.3 Iniciando o LabVIEW

Ao se iniciar o LabVIEW pela primeira vez, o programa irá inicializar sempre um novo VI em “branco”. Em todo VI, inicial ou naqueles armazenados, são ordenadas duas janelas distintas, mas pertencentes ao mesmo programa/arquivo ou VI, e portanto com mesmo nome.

Estas janelas são designadas, janela de diagrama (fundo cor branca-*default*) e janela painel (fundo cor cinza-*default*) tal como exemplificado na figura 2.2.

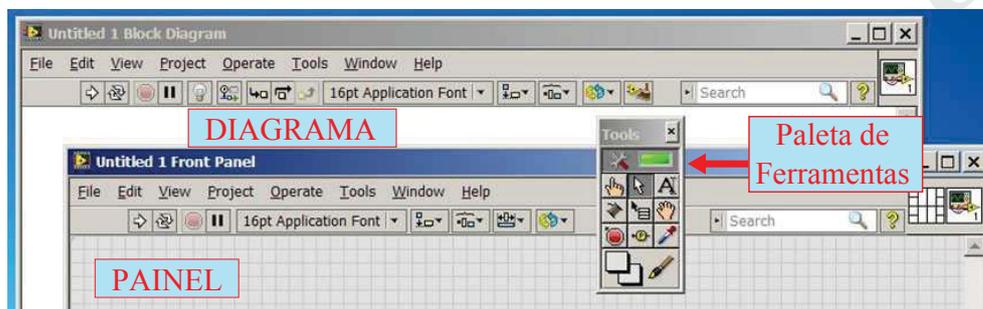


Figura 2.2: Janelas DIAGRAMA e PAINEL

Estas duas janelas possuem funções distintas. A janela DIAGRAMA funciona como a parte interna do VI onde são realizadas as operações/funções matemáticas, conexões ou fiação entre os vários blocos operacionais, gerenciamento dos sinais, pré e pós processamento do VI. A janela painel funciona como a parte externa frontal onde são alocados os *displays* de resultados (numéricos ou gráficos) e todo tipo de indicador de sinalização e o vários tipos de chaves e botões de comando.

Quando a janela DIAGRAMA está ativa, tal como indicado na figura a seguir, dispõe-se de dois pop-menus (gráficos) com as funções TOOLS e FUNCTIONS. Estes *pop-menu* podem ser acionados com auxílio do Mouse (direito) ou por meio da Barra do Menu-Superior: *Windows-Show-Tools-Palette* e *Windows-Show-Functions-Palette*.

No menu TOOLS dispõe-se de funções de edição do diagrama em termos de fiação, movimentação, edição de textos e texturas de cor. Já o menu “FUNCTIONS” permite inserir todo tipo de blocos de operações matemáticas, estruturas de programação, manipulação de arquivos e principalmente de escolha e definição dos blocos periféricos. Cada uma destas opções pode ser identificada deslizando-se o cursor do Mouse sobre os ícones dos paletes.

Quando a janela PAINEL está ativa, tal como a seguir, além do pop-menu TOOLS, dispõe-se agora do pop-menu CONTROLS, os quais são acessíveis da mesma maneira como no caso da janela DIAGRAMA. O menu TOOLS tem a mesma função que no

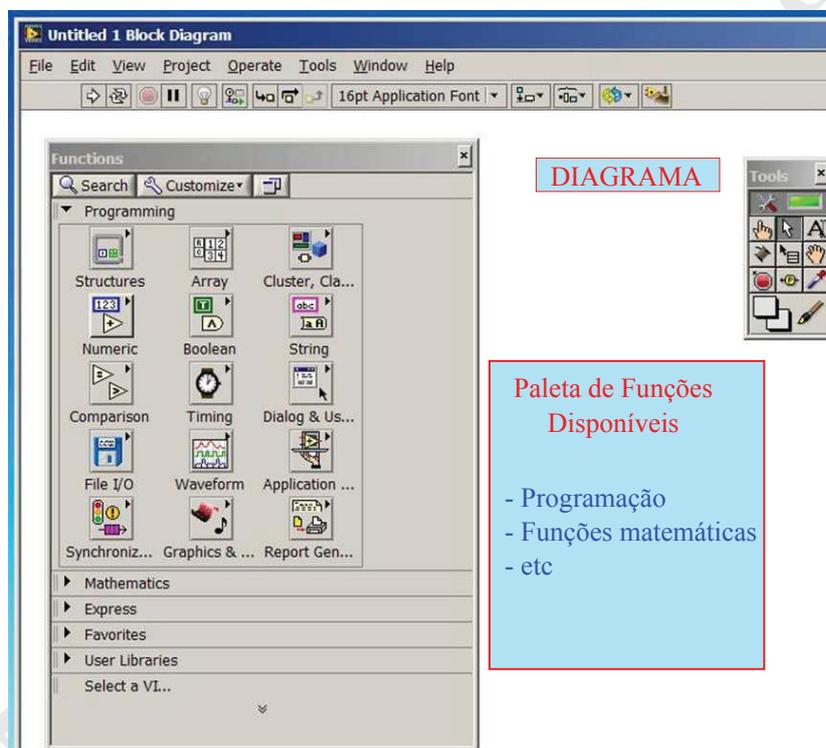


Figura 2.3: Paletes de funções e ferramentas na janelaDIAGRAMA

caso da janela DIAGRAMA, ou seja para edição do painel desejado em termos de movimentação, edição de textos, de *displays* e botões de painel. O menu “CONTROLS” possibilita a escolha e inserção de elementos de painel e blocos de pré- e pós processamento. Estes elementos são do tipo chaves, controles numéricos, botões de comando, indicadores luminosos ou numéricos, etc.

Deve-se atentar para o fato de que botões de comando e indicadores só estão acessíveis na janela PAINEL enquanto que operadores matemáticos e funcionais só podem ser alocados na janela DIAGRAMA.

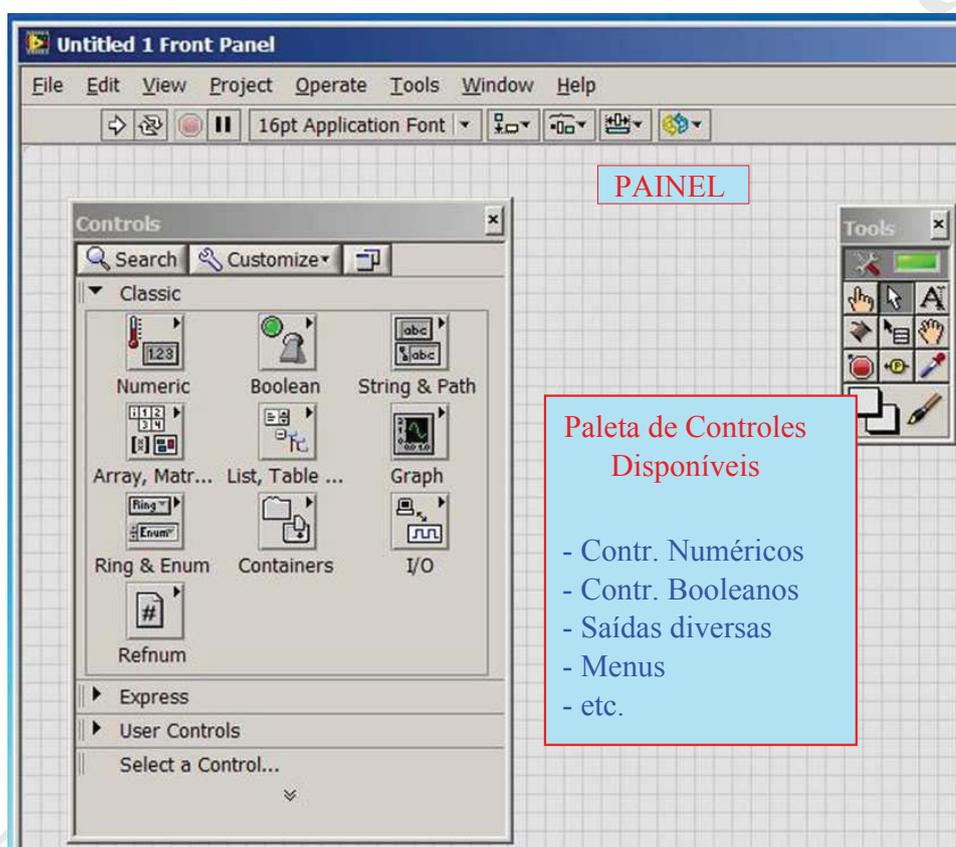


Figura 2.4: Paletes de Controles na janelaPAINEL

## 2.4 Operação Básica LabVIEW

As operações básicas do LabVIEW estão disponíveis tanto na janela DIAGRAMA como na de PAINEL e estão alocadas nas Barras de Comandos ou Menu-Superior. Na pri-

meira delas é indicado o nome do VI acompanhado da indicação Diagrama na janela DIAGRAMA. A Segunda Barra de comandos disponibiliza a abertura dos sub-menus que executam ou acessam todos os recursos do LabVIEW e que serão entendidos ao longo do uso da programa.

A próxima barra de comandos resume algumas das principais funções/recursos por meio de ícones de acesso. As 4 primeiras opções se referem à execução do um determinado VI previamente editado, tal como indicado na Fig. 2.5.

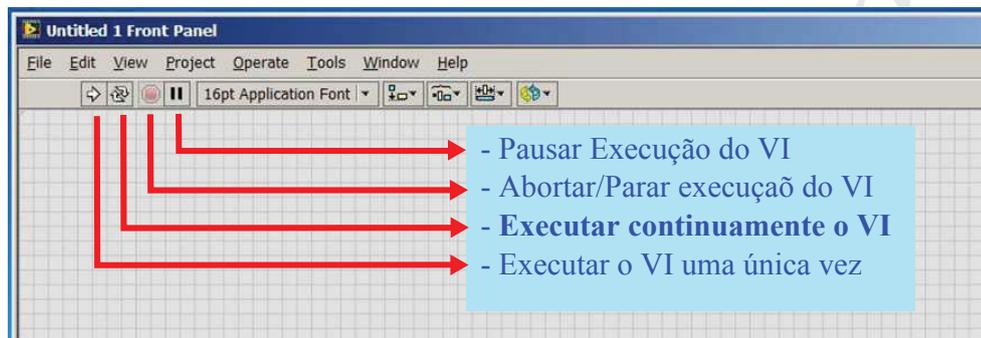


Figura 2.5: Barra de comandos principais

O primeiro comando executa o VI, ou seja suas instruções de programação uma única vez. O segundo irá executar continuamente e repetidamente do início ao fim as instruções do VI. Ao longo do curso será esclarecido o devido uso destes comandos.

Os demais comandos envolvendo o Menu CONTROLS e FUNCTIONS estão descritos na Apostila do LabVIEW. Nas próximas aulas serão realizadas atividades de reconhecimento de cada um dos principais recursos do LabVIEW até se obter relativo domínio da linguagem G para se executar as tarefas de controle a serem estipuladas ao final do curso.

## 2.5 Atividades Introdutórias

i-) Inicialize o LabVIEW e gere um arquivo de VI.

ii-) Teste do comando Ctrl-T (verifique na barra de comando/Windows que se pode alterar esta forma de janela)

iii-) Teste o comando Ctrl-E repetidamente

iv- ) Iniciando na janela PAINEL, insira o Pop-Menu CONTROL (click mouse direita) e percorra com o curso do mouse todos os sub-menus do *Pop-Menu* CONTROL.

v- ) Verifique agora o conteúdo de cada sub-menu no *Pop-Menu* CONTROL percorrendo com o curso do mouse as opções disponíveis.

vi- ) Repita os itens ( iv ) e ( v ) para o *Pop-Menu* FUNCTIONS da janela DIAGRAMA.

vii- ) Insira elementos quaisquer da janela PAINEL e verifique ao mesmo tempo o que acontece na janela DIAGRAMA.

viii- ) Insira agora elementos de operações matemáticas, numéricas, booleanas, lógicas string etc., e verifique ao mesmo tempo o que acontece na janela PAINEL. (NÃO INSIRAR POR ENQUANTO NENHUMA FUNÇÃO I/O - PRIMEIRA LINHA DO POP-MENU FUNCTIONS)

ix- ) Observe nesta janela Painel as cores dos elementos inseridos e verifique a respectiva formatação (numérica/booleana/string) do tipo de variável com ajuda da apostila.

x- ) Com um *click* a direita em um elemento do diagrama altere sua formatação de tipo escolhendo "*Representation*"

xi- ) Repita o item anterior a partir da janela PAINEL verificando o que ocorre na outra janela.

xii- ) Escolha um dos elementos Indicador no PAINEL, *click* a direita e selecione "*Show / Label* e *Show / Digital Display*".

xiii- ) Repita o item ( xii ) selecionando agora "*Replace*" e verifique que se pode fazer como resultado.

xiv- ) Selecione a ferramenta (SETA : Position/Size/Select) do menu TOOLS. Teste os recursos desta ferramenta: deslocar, alterar tamanho, selecionar, etc.

xv- ) Teste a ferramenta "*Set Color*"

xvi- ) Teste a ferramenta de texto em ambas as janelas: tamanho, cor, alinhamento,

etc.

xvii- ) Verifique o que se pode fazer o a opção “*Operate Value*” do Menu TOOLS em ambas as janelas.

xviii- ) Identifique na janela DIAGRAMA a diferença (desenho) entre um Indicador e um Controle numérico, booleano e string.

xix- ) Realize operações (aritmética ou booleanas) entre elementos de entrada e saída e execute o VI selecionando (RUN) ou (RUN CONTINUOUS).

SEL359-2015 MLAGuiar

## Laboratório 3

# DIAGRAMAS E PAINÉIS BÁSICOS

### 3.1 Objetivos

Investigar as propriedades e recursos básicos dos blocos numéricos, lógicos e booleanos do LabVIEW em termos de janela de painel e de diagrama.

### 3.2 Generalidades

Os blocos numéricos servem aos propósitos de manipulação numérica através de funções ou operadores matemáticos. Em praticamente todo tipo de procedimento utilizando o LabVIEW será necessária a realização de tratamento matemático nas grandezas, constantes ou variáveis quaisquer do processo em execução.

A manipulação matemática pode ser executada sobre uma constante própria do algoritmo específico ou sobre uma variável “lida” em um periférico ou sobre um valor proveniente de uma entrada definida pelo usuário. O resultado de uma manipulação matemática ou é repassado para processamento posterior ou é armazenado ou então é simplesmente indicado em um dispositivo de saída (PAINEL).

Neste caso, o LabVIEW disponibiliza através de uma “biblioteca” ou menu as funções e operadores matemáticos mais comuns na janela DIAGRAMA, através do Pop-Menu FUNCTIONS. Para se ter uma visão geral dos diversos blocos de funções, aciona-se o ícone de *All-Functions*, o qual irá mostrar as possibilidades de operações tal como a seguir.

As demais funções deste bloco serão estudadas em outras aulas. Na figura 3.1 estão

evidenciados os blocos de interesse de análise exclusiva nesta aula, ou seja, funções numéricas, Booleanas, *String* e de comparações. Dentro destes blocos existem ainda sub-conjunto de funções de conversão de dados e de constantes genéricas.

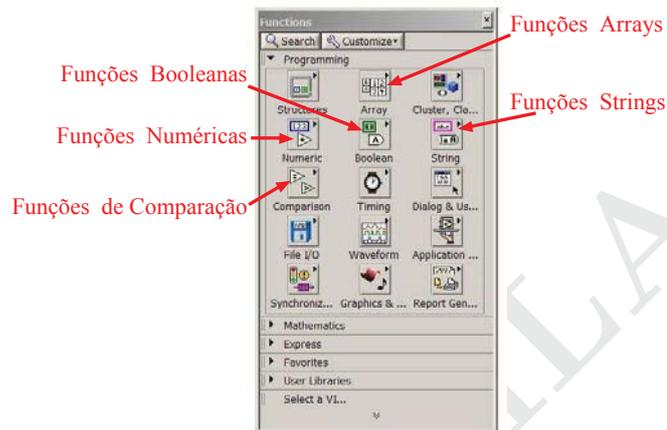


Figura 3.1: Menu de Funções da janela Diagrama

Selecionando-se o sub-menu **Numeric**, tem-se acesso aos operadores matemáticos de uso comum, os quais são indicados por símbolos ou ícones gráficos. À direita existem 4 subdivisões de funções matemáticas típicas: conversão de formatos numéricos, trigonométricas, exponenciais, manipulação de números complexos e constantes genéricas. Da mesma forma, nos demais sub-menus serão disponibilizadas funções, constantes e operações envolvendo representação booleana, *string* e modos de comparações entre representações.

No lado da janela Painel, dispõe-se também no Pop\_Menu CONTROLS de um sub-menu **Numeric**. Neste caso dispõe-se de diversos blocos ou ícones gráficos representando indicadores ou controles numéricos. Os indicadores são dispositivos de saída, ou seja para indicar o resultado de uma manipulação de sinal e os controles são dispositivos de entrada, ou seja para pré estabelecer o valor de uma grandeza ou constante para ser processada. Da mesma forma existem as bibliotecas de manipulação para funções e variáveis do tipo Booleanas, Lógicas e *String*. Na janela Painel nestes casos, são providas as respectivas formas de entrada e saída destes tipos de dados. As funções lógicas no LabVIEW são disponibilizadas exclusivamente no Pop\_Menu FUNCTIONS/*Comparison*.

A interligação ou direcionamento de fluxo entre controladores ou constantes numéricas com os operadores ou funções numéricas e os indicadores numéricos são realizados com a ferramenta (TOOLS) *Connect Wire*. Desta forma cada um dos elementos na ja-

nela Diagrama deve ser interligada formando uma sequência lógica de processamento matemático.

Lembrar que a cor da “ligação” entre os vários elementos define a formatação numérica da grandeza transportada. A normalização de cores é tratada na “Apostila do LabVIEW”.

### 3.3 Atividade de aplicação

Na sequência é sugerida uma série de procedimentos que visa o treinamento inicial com a programação de VI's com o LabVIEW. Nesta primeira atividade é proposta a execução de simples sequências de manipulação. É recomendado que se proceda a devida entrada e saída de dados/resultados por meio da janela PAINEL com devido elementos de painel.

i ) - Inserir no Painel uma chave Booleana e um *Knob* com fundo de escala entre 0 e 100. Este valor deve ser indicado num Velocímetro que deverá indicar o valor em Km/h ou este valor convertido em rd/seg de acordo com a escolha feita pela chave Booleana. OBS.: usar a Função *Select* (DIAGRAMA) e o recurso *Property Node* (PAINEL) para ajustar automaticamente o fundo de escala do Velocímetro de acordo com a escolha. Ver dica na Fig. 3.2.

ii ) - Gerar no painel uma entrada tipo *Array* Booleano para acionar um *Display* tip *Array* Booleano tal como indicado na Fig. 3.3.

iii ) - Realize por meio de entradas ou constantes tipo numérica uma manipulação Booleana de 8-Bits e indique o resultado em um *Display* de 8 leds, por exemplo, trocando-se a entrada Booleana da figura 3.3 por um *Knob* ou outra entrada numérica. OBS.: Para isto é necessário o uso de manipulação de *Arrays* e/ou conversão de dados.

iv - Realize as operações necessárias para avaliar faixas de magnitude de uma variável e representar o resultados tal como na ilustração da figura 3.4 ou de forma semelhante.

v ) - (OPCIONAL) Teste os recursos de manipulação de menu e pop-menus na janela Painel e avalie uma forma de processar a operação destes recursos. Estes recursos estão disponíveis no Pop-Menu *CONTROLS/List & Rings* . Um exemplo é indicado na figura 3.5.

vi ) - (OPCIONAL) Teste os comandos de geração e manipulação de *Arrays* na janela DIAGRAMA e na janela PAINEL. Um exemplo é indicado na figura 3.6 como ponto de partida para ser acrescido e outras opções e para ser estudado.

vii ) - (OPCIONAL) Teste e avalie os recursos de manipulação de variáveis *String*. Inspeccione o exemplo da figura 3.7.

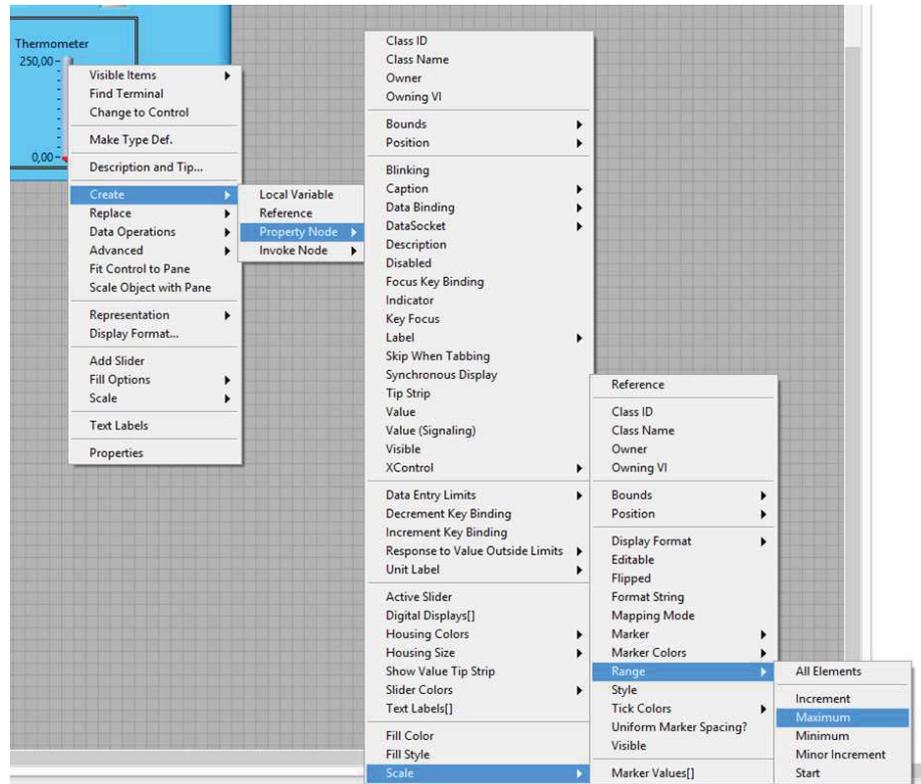


Figura 3.2: Exemplo de acesso à manipulação automática de Indicador : *Property Node*

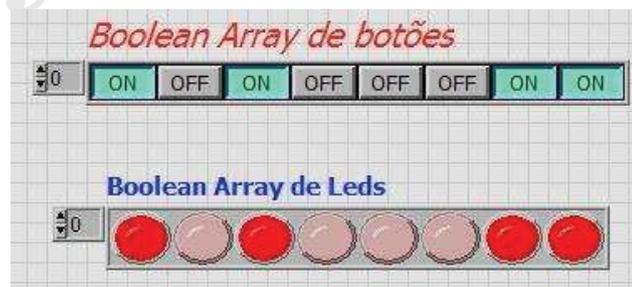


Figura 3.3: Exemplo de indicação tipo *Array* booleano

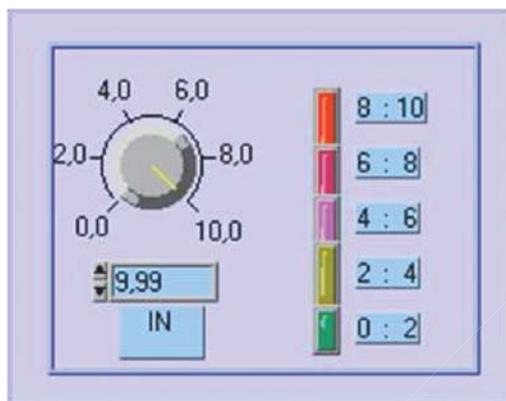
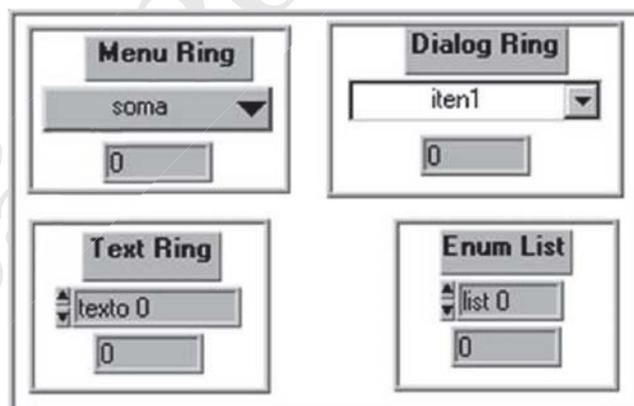
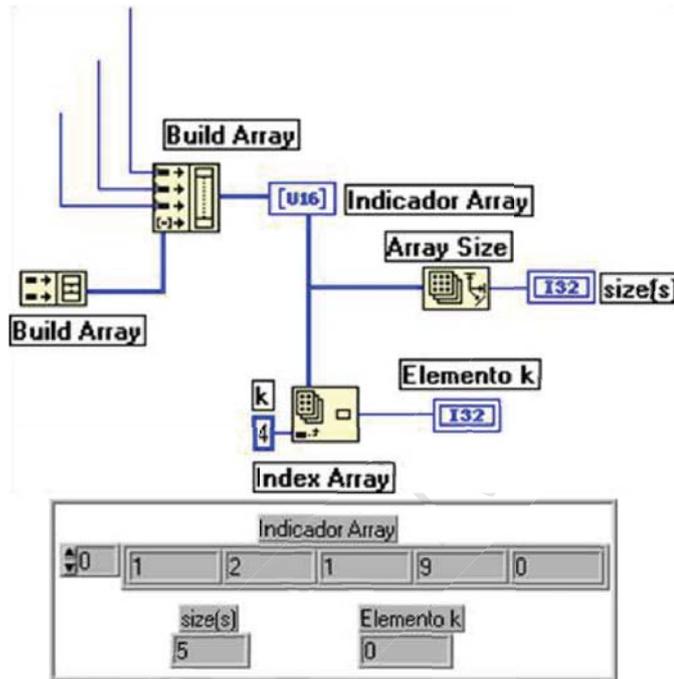
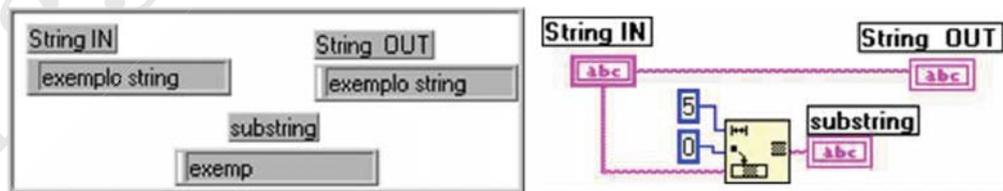


Figura 3.4: Exemplo de PAINEL para comparação de magnitude

Figura 3.5: Exemplo de pop-menus e *List-Ring*

Figura 3.6: Exemplo de manipulação de *Arrays*Figura 3.7: Exemplo de manipulação de *Strings*

**Desafio Prática 3**

viii ) - **DESAFIO DA AULA** :

Gerar um VI para converter unidades de temperatura de Graus-Celsius em Graus-Fahrenheit.

- A entrada deve ser via um *Knob* entre  $-10^{\circ}C$  e  $100^{\circ}C$  com uma casa decimal de precisão e respectivo *Display* Digital.

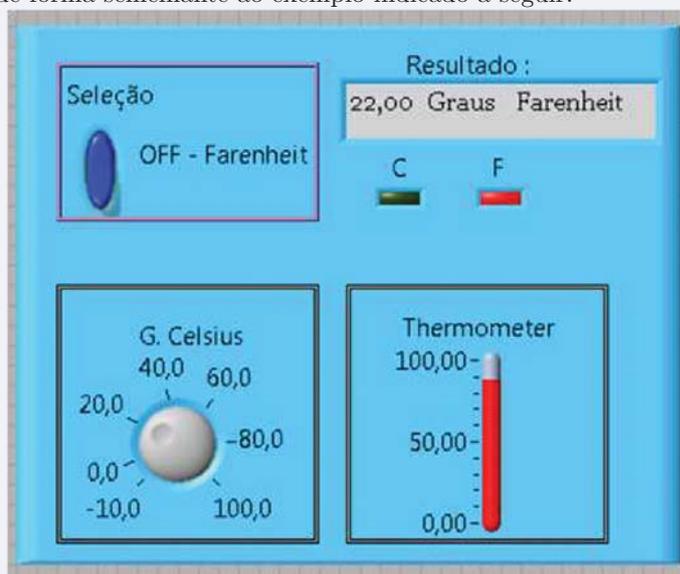
- A saída deve ser um termômetro com um Fim-de-Curso (escala) adequada à variação da entrada e com 2 casas decimais de precisão.

- Incrementar o VI com uma chave-digital que seleciona as opções ( $^{\circ}C$ ) e ( $^{\circ}F$ ) que quando acionada indique diretamente o valor do *Knob* em ( $^{\circ}C$ ) ou o valor do *Knob* convertido em ( $^{\circ}F$ ) no termômetro de saída.

- O Fundo de Escala do Termômetro deve ser alterado automaticamente segundo a seleção da chave-digital.

- Incrementar o VI com um indicador *String* indicando a frase: **RESULTADO EM** (Graus Celcius  $^{\circ}C$ ) se a seleção for em Graus Celcius. Caso seja selecionado Farenhei, escrever a frase : XX.X Graus Celcius  $^{\circ}C$ ) equivale a YY,YY Graus Farenheit  $^{\circ}F$ . Indicar ainda com um LED-Verde se o resultado for em ( $^{\circ}C$ ) ou com um LED-Vermelho se o resultado for em ( $^{\circ}F$ ).

- Incrementar o painel do VI com recursos de ilustração/cor/*Decorations* de PAINEL, de forma semelhante ao exemplo indicado a seguir.



## Laboratório 4

# ESTRUTURAS NÃO TEMPORIZADAS

### 4.1 Objetivos

Estudar e investigar os recursos de programação de modos estruturados em um VI ou em uma estrutura de controle tais como, *Loop FOR*, *Loop WHILE*, *Loop SEQUENCE* e modos estruturados de decisão *IF* e *CASE*.

### 4.2 Generalidades

Como em todo programa mais complexo ou em estratégia de supervisão e controle se faz necessário a geração de um programa estruturado com tomada de decisões, sequenciamento de rotinas ou tarefas e além de tudo de forma organizada, numerável e supervisionada.

Em linguagens convencionais estas características são proporcionadas com os recursos de comando para gerenciamento de *loops* ou sequências ordenadas. Iteração de comandos é uma das mais importantes estruturas que podem ou não ser contáveis, ou seja, que permitem enumerar e indexar resultados de cada iteração em um dado programa.

Os recursos mais comuns são disponibilizados no LabVIEW, porém como todos os outros recursos, em forma gráfica. Estes recursos são acessíveis no Pop-Menu *FUNCTIONS* da janela *DIAGRAMA* no sub-menu *Programming - Structures*. Esta forma de acesso é ilustrada na figura 4.1.

No menu de opções da janela *Structures* dispõem-se dos recursos principais : *While-Loop*, *Flat-Sequence* e *Case-Structure*, *Stacked-Sequence*, *For-Loop* e *Formula-Node*,

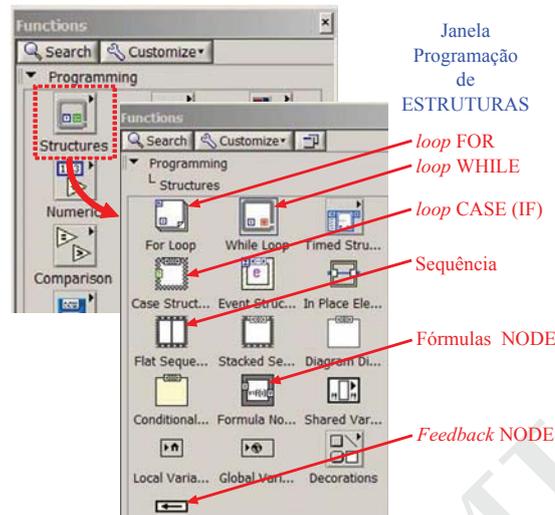


Figura 4.1: Menu e sub-menu de Estruturas

*Feedback-Node*, *Timed-Loop* e *Event-Structures*, etc. O objetivo principal desta aula é estudar o uso de alguns destes recursos, os quais serão brevemente discutidos a seguir e posteriormente exercitados em exemplos.

### 4.3 Estrutura SEQUENCE

A estrutura SEQUENCE como o próprio nome diz e também como o desenho gráfico desta, é apresentado na janela DIAGRAMA, representa uma forma de se estabelecer a sequência desejada de execução de tarefas dentro de um VI. O desenho desta estrutura é o mesmo de uma película de filme ou como um *frame* de filme. A apresentação pode ser tipo *Flat* ou *Stacked*, ou seja, com os *frames* sequenciais linearmente ou com os *frames* sobrepostos respectivamente. Os *frames* são todos executados na sequência definida. No caso de *Flat-Sequence*, a disposições do desenho já determina a direção e disposição de passagem de valores entre os *frames*.

### 4.4 Estrutura CASE ( Lógico e Múltiplo)

A estrutura CASE é por *default* uma estrutura tipo IF em que manipula um estado ou variável externa booleana *True/False*. Esta entrada deve ser providenciada externamente à estrutura. Neste caso as condições *True* ou *False* dão acesso a dois *frames* ou janelas onde são realizadas as operações do caso *True* ou do caso *False*.

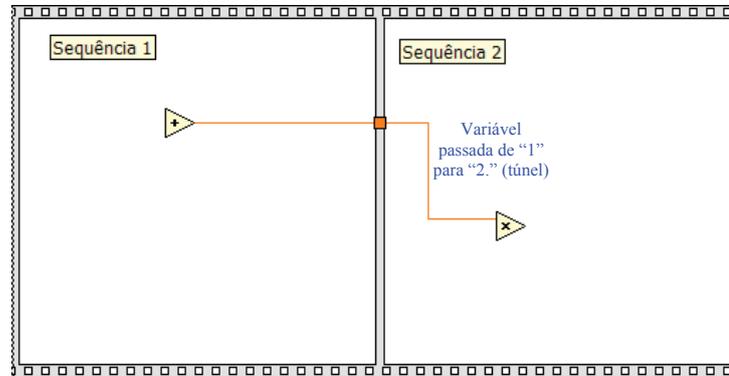
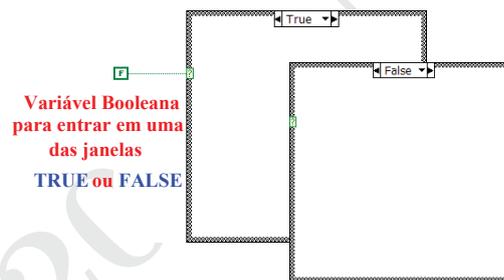
Figura 4.2: Exemplo de estrutura *Sequence*

Figura 4.3: Exemplo de CASE Lógico

Esta mesma estrutura funciona também como uma função CASE de condições múltiplas. Neste caso, em vez de *True/False* a estrutura é ordenada numericamente. Novamente, cada caso numérico representa um *frame* ou janela onde são editados os comandos relativos a cada caso a ser manipulado. Diferente da estrutura tipo *frame* de sequência, apenas uma janela é executada de acordo com a variável passada para esta estrutura.

Para converter uma estrutura CASE booleana em estruturas CASE múltipla basta conectar uma variável numérica ao símbolo de comparação logo após a inserção da estrutura no diagrama. Desta forma a estrutura booleana se torna numérica sendo "0" o caso "default". Para acrescentar novos "casos" basta acionar o Pop-Menu na borda da estrutura e escolher a opção desejada. Esta mesma estrutura pode operar com entradas tipo *string*. Para isto basta seguir o mesmo procedimento do caso numérico.

Para se obter o resultado da manipulação do CASE em uma variável, basta conectar

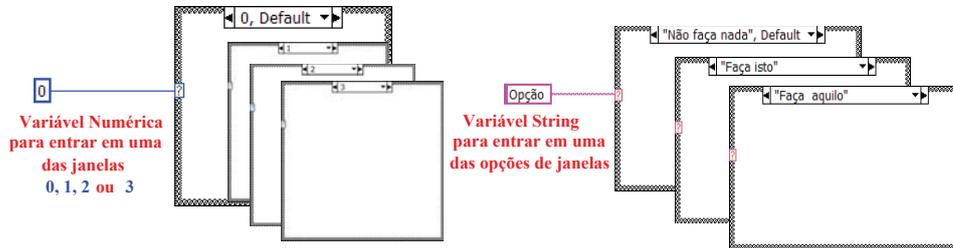


Figura 4.4: Exemplo de CASE Múltiplo

o resultado na borda do *frame*. Se todos os casos vão manipular uma única variável, então em cada frame o resultado dever ser conectado neste mesmo ponto.

## 4.5 Estrutura FOR

A estrutura FOR é uma janela em que se executam “N” vezes os comandos dentro dela, sendo “N” inicializada externamente. “N” é necessariamente uma constante inteira. Desta forma a estrutura FOR realiza uma seqüência finita e contável de operações. Para que uma variável seja disponível de forma indexada fora do *Loop* é necessário habilitar a indexação ao se executar a fiação das operações para fora (borda) do *Loop*. A forma normal é não-indexada. Pode-se ainda disponibilizar variáveis recursivas para serem utilizadas por exemplo em uma equação recursiva. Para isto, basta clicar na borda do FOR e adicionar *Shift-Register*. Isto representa que a cada iteração “k” no FOR, a variável conectada entrando no *Shift-Register* estará disponível como uma variável da iteração “k – 1” na próxima iteração. Uma opção também é usar o *Feedback-Node*. Dentro da estrutura do FOR encontra-se ainda o ícone “i”, o qual contém o valor da iteração atual em execução e pode ser usado na tarefa em execução. Esta variável interna é sempre inicializada com 0 (zero) a cada re-início da estrutura FOR.

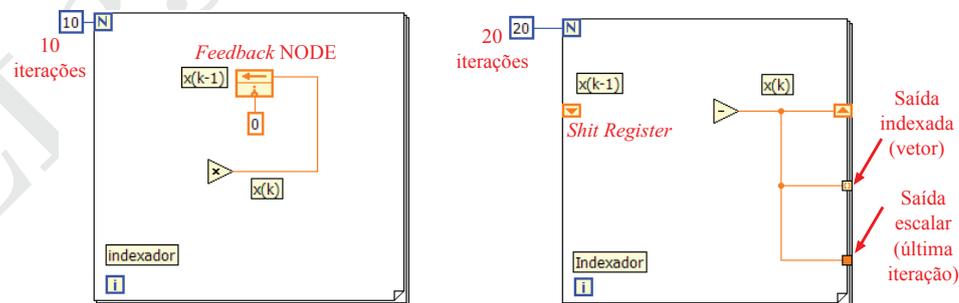


Figura 4.5: Exemplo de Loop Contável tipo FOR

## 4.6 Estrutura WHILE

A estrutura WHILE é semelhante àquela do FOR, porém o *Loop* é executado indefinidamente até uma dada condição lógica dentro do *Loop* seja avaliada e um resultado *False* determine a interrupção do *Loop*. Esta condição *False* deve então ser conectada ao ícone de interrupção da estrutura, tal como visto na figura a 4.6. Também neste caso é possível a inserção de variáveis indexadas para fora do *Loop* e variáveis recursivas para o próprio *Loop*. Na estrutura do WHILE dispõe-se também do contador de iteração cujo valor de cada iteração é disponível no ícone “i”. Se necessário os recursos de *Shift-Register* ou *Feedback-Node* podem ser usados dentro do WHILE da mesma forma como no caso do FOR.

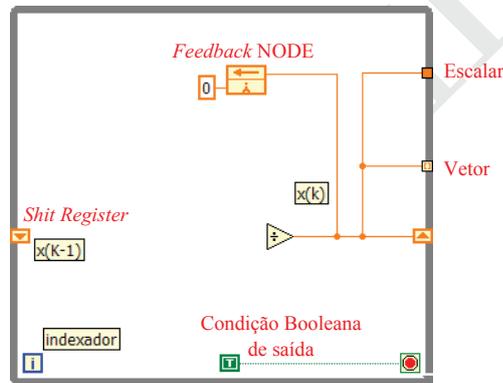
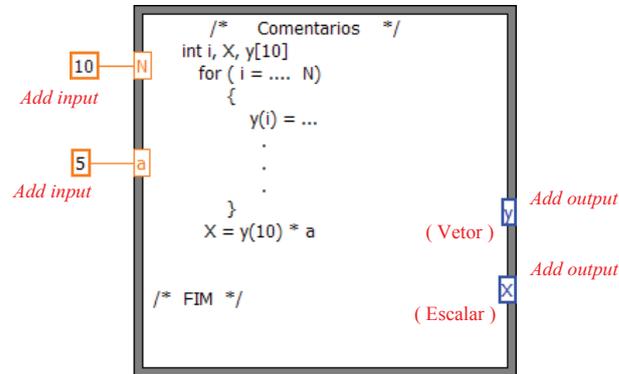


Figura 4.6: Exemplo de *Loop* não contável tipo WHILE

## 4.7 Estrutura FORMULA - NODE

A próxima opção de estruturas de programação é chamada FORMULA-NODE. Este recurso é o equivalente de se chamar uma sub-rotina externa. No caso insere-se uma sequência de comandos em linguagem equivalente a sintaxe C ou C++ em função das variáveis de entrada, internas e de saída. As variáveis de entrada e de saída devem obrigatoriamente ser nomeadas e são manipuladas dentro da estrutura pelo respectivo nome. Dentro da estrutura *Formula-Node* só podem aparecer expressões matemáticas.

Na sequência serão sugeridas algumas atividades como treinamento de uso dos recursos apresentados acima.

Figura 4.7: Exemplo de estrutura *Formula-Node*

## 4.8 Atividade de aplicação

Procure executar os VIs com as funções solicitadas a seguir sem utilizar recursos de temporização que serão vistos na próxima aula. Procure treinar também a geração e gerenciamento do VI por meio de um painel equivalente dotado entradas, saídas e mensagens “*Strings*”.

i) - execute um VI usando um *Loop FOR* para realizar a contagem de iterações e que indique a soma parcial das iterações  $s_p$  e a soma final  $s_T$  das N-iteraões.

$$s_f = \sum_{i=1}^N i \quad s_P = [s_{p1} \ s_{p2} \ s_{p3} \ \dots \ s_{pN}]$$

ii) - Execute o mesmo VI acima, mas usando um *Loop WHILE* para realizar o mesmo somatório do caso anterior. Acrescente neste VI um tipo de controle de interrupção intencional das iterações através do painel de controle.

iii) - Execute um VI que determine a execução dos itens ( i ) e ( ii ) em seqüência. Use o valor final  $s_T$  da primeira seqüência item ( i ) para determinar o valor inicial de contagem do item ( ii ) na segunda seqüência.

iv) - Inserir mais uma janela de seqüência e execute nela um VI em que uma condição lógica for *True* executa o item ( i ) e se for *False* executa o item ( ii ), ou seja, *True* executa FOR e *False* executa WHILE. Indique no Painel uma mensagem tipo *String* com a indicação de qual condição foi executada.

v) - Inserir nova janela de seqüência e execute nela um VI com FORMULA-NODE para realizar a mesma operação de soma especificada no item ( i ). Consulte o *Help* para verificar as funções disponíveis, por exemplo, os comandos das tarefas FOR ou WHILE

realizadas nos itens (i) ou (ii). No *Help* é possível verificar a sintaxe correta a ser usada neste recurso.

vi) - OPCIONAL Execute um novo VI que dependendo de uma entrada de controle tipo Menu do painel execute ou um FOR , ou um WHILE, ou um IF ou um FORMULA-NODE. Use um estrutura tipo CASE numérico ou *String*. Novamente indique com uma mensagem *String* apropriada qual CASE foi executado.

SEL359-2015 MLAguiar

## Laboratório 5

# SAÍDAS GRÁFICAS E PARA ARQUIVO NÃO TEMPORIZADAS

### 5.1 Objetivos

Aprofundar o estudo e investigações dos recursos de programação de modos estruturados não temporizados e implementar saídas gráficas em painéis e para arquivo.

### 5.2 Generalidades.

Na aula anterior foram estudados os recursos de estruturação de programas através de *Loops* e sequenciamento de rotinas e ou tarefas. Estes recursos têm como finalidade a organização de tarefas em um programa e principalmente indexar as etapas e variáveis envolvidas em cada tarefa.

É muito comum nestas ocasiões o interesse ou mesmo a necessidade de se indicar os resultados sequenciais em um tipo de forma gráfica ou em arquivos para que se possa investigar o andamento ou re-processar os dados *Off-Line*.

É importante lembrar da aula anterior que toda variável indexada em um *Loop* dever ser habilitada para indexação na saída dos *Loops* e são inerentemente variáveis tipo *Array*, ou seja, são vetores linhas ou colunas. Portanto os recursos de manipulação de *Arrays* são importantes.

Para as atividades específicas do curso de controle digital, o importante será ter um

razoável domínio dos recursos gráficos em painel e o registro de dados em arquivos que possam ser transferidos para outros pacotes de software tais como o Matlab.

### 5.3 Saídas gráficas

O LabVIEW dispõe de basicamente três tipos de saídas gráficas que são disponíveis na janela Painel e no *Pop-Menu* CONTROLS - *Graph*. Estas funções são *Chart*, *Graph* e *XY-Graph*. Cada uma destas opções pode servir a diferentes propósitos como será apresentado a seguir e posteriormente treinado em atividades no laboratório.

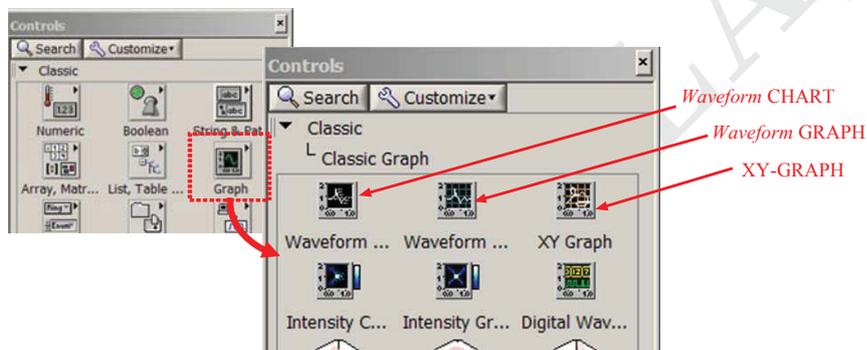


Figura 5.1: Menu e sub-menu de recursos gráficos

Toda saída gráfica alocada por meio da janela Painel irá alocar um correspondente elemento na janela de Diagrama, o qual deverá ser devidamente conectado na finalização do VI.

#### 5.3.1 Saída *Chart*

As saídas do tipo *Chart* são utilizadas para as tarefas mais simples de expressão gráfica de resultados. O *Chart* não necessita da grandeza no eixo das ordenadas e ainda pode indicar valores escalares em forma gráfica. Como entrada o *Chart* aceita valores escalares, vetores de uma dimensão e vários vetores de uma dimensão e ainda entradas tipo WDT (*Waveform-Data-Type*): que contém informações de data-hora). As saídas de *Chart* são tipo *Appended*, ou seja, cada nova variável ou *Array* passado para o *Chart* é acrescido na tela até que se exceda o tamanho do *Buffer* de entrada que na realidade é o tamanho da variável *Chart*.

As várias opções e recursos de representação gráfica podem ser acessados na janela PAINEL acionando-se o Pop-Menu do elemento *Chart*. As principais destas opções no Pop-Menu estão resumidas na Opção Propriedades e são:

**Opção Show** : onde se pode habilitar ou não a indicação das escalas XY, da legenda, *Label*, *Zoom*, cores e estilos dos gráficos e reticulado, etc.

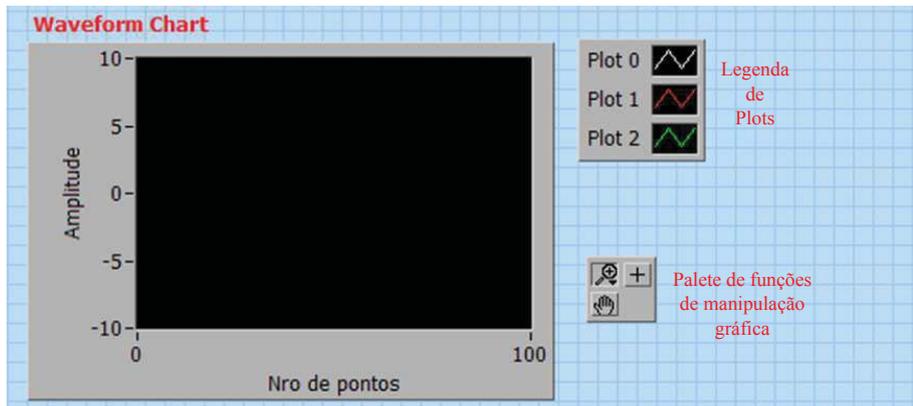


Figura 5.2: Exemplo de saída tipo *Chart*

**Opção Data Operation (Clear Chart)**: onde se pode reinicializar o *Chart*, Limpar toda tela e estabelecer os modos auto-escala XY.

**Opção Chart History length** : onde se pode estabelecer o tamanho da variável do *Chart* que será preenchida.

**Opção Overlay-Stack Plots** : esta opção tem função quando as variáveis são passadas para o *Chart* em forma de Agrupamento ou *Cluster*. Desta forma é possível colocar vários gráficos (de várias grandezas) no mesmo *Chart* ou em *sub-plots* (em *displays* gráficos separados).

Este modo pode ser gerado de duas maneiras diferentes tal como ilustrado na figura 5.3 usando-se o recurso *Build-Array* ou *Bundle*. O resultado deste tipo de apresentação gráfica é indicado na figura 5.4.

Procure inspecionar cada uma das das propriedades acessando-se o item correspondente de propriedades listadas no Pop-Menu.

### 5.3.2 Saída tipo *Graph*

Este tipo de saída permite se conectar uma variável como ordenada (eixo x) no gráfico de saída além de permitir múltiplos gráficos com a diferença de que neste caso a entrada deve sempre ser do tipo *Array* ou do tipo WDT. As opções de visualização e configuração são praticamente as mesmas do *Chart* e também acionadas via Pop-Menu.

Uma das maneiras de se usar o *Graph* é simplesmente conectar a variável ou variáveis colecionadas em *Array* ao elemento *Graph*. A principal diferença com relação ao *Chart* é que o eixo das ordenadas (eixo x) será simplesmente o número de indexação da variável,

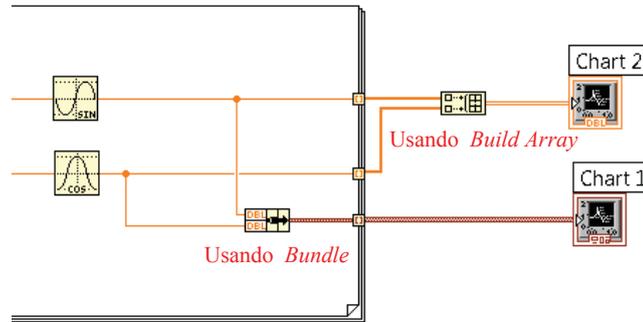
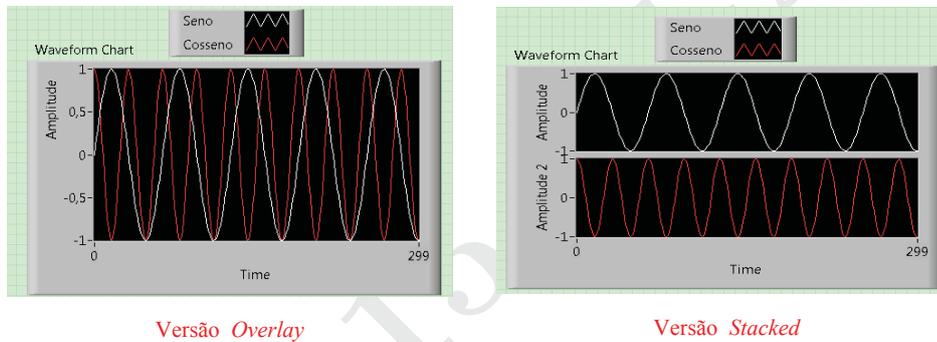


Figura 5.3: Recursos de agrupamento de dados

Figura 5.4: Visualização dos recursos *Overlay* e *Stacked* do *Chart*

ou seja do tamanho exato da dimensão do *Array* independente se se executar uma única vez ou executar continuamente. Desta forma o *Graph* se comporta tal como o *Chart*. Um exemplo de saída tipo *Graph* é visto na figura 5.5

Para se obter múltiplos gráficos no mesmo *display*, usando ou não o arranjo do eixo x, colecciona-se inicialmente os pontos em um *Array* múltiplo e conecta-se ao *Graph*.

## 5.4 Saída tipo *Graph XY*

Este tipo é semelhante ao do *Graph*, mas permite conectar diferentes variáveis respectivamente aos eixos X e Y. Para tal, as variáveis devem ser coletado com auxílio de um Agrupamento tal como na figura 5.6. Quando o elemento de saída *Graph-XY* é obtido a partir da janela *Controls/Classic* usa-se os elementos *Bundle* para agrupar os sinais a serem plotados.

Na versão do LabVIEW 2013, a inserção do elemento *Graph-XY* a partir da janela *Control/Express* já contém um módulo *Build XY Graph* na janela DIAGRAMA e que

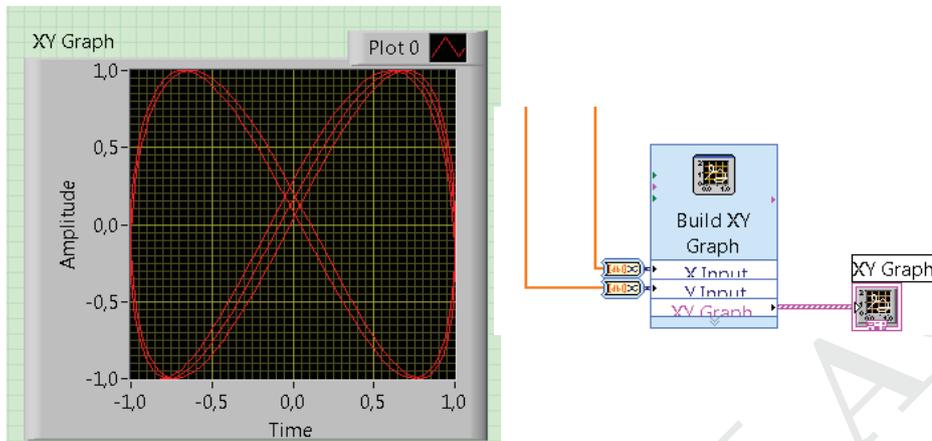


Figura 5.5: Exemplo de saída *Graph*

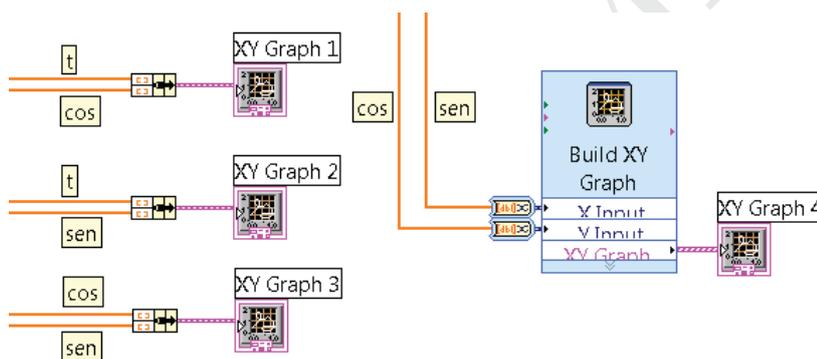


Figura 5.6: Exemplo de agrupamentos para o uso do *Graph-XY*

facilita a montagem do vetor de variáveis a serem manipuladas no gráfico XY. Um exemplo disto é visto na figura 5.7.

## 5.5 Manipulação de dados de arquivos ou para arquivos

Este tipo de operação pode ser de interesse para se fazer registro periódico ou contínuo de grandezas que determinam o funcionamento de um VI e possibilita a armazenagem ou leitura de dados em arquivos. Este recurso será usado para armazenar em arquivo todos os dados de interesse do experimento e depois carregá-los no ambiente Matlab e comparar com as simulações do Simulink.

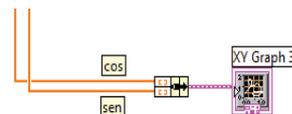
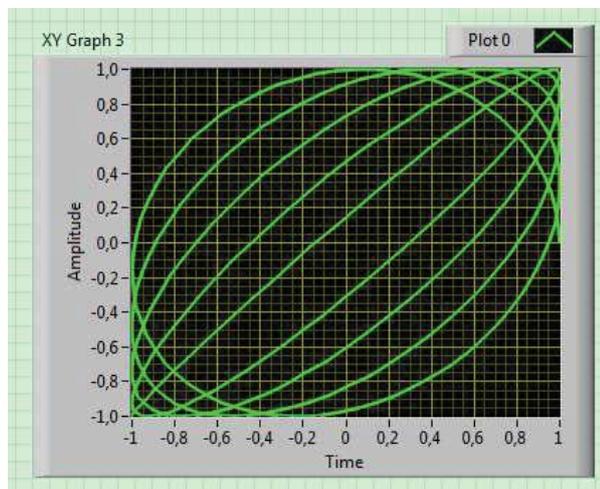


Figura 5.7: Exemplo de uso do *Graph-XY*

Com relação à disciplina Controle Digital, estamos interessados em fazer o registro das entradas, saídas, e a das ações de controle nos experimentos a serem estudados.

O acesso às funções de arquivo é feito na janela DIAGRAMA no Pop-Menu *Programming/File I/O* e contém as opções tais como indicado na figura 5.8.

As duas primeiras opções permitem escrever e ler dados numéricos em um arquivo enquanto que a terceira e quarta permitem manipular apenas caracteres (*strings*). As demais opções são de caráter específico e não serão abordadas aqui.

Para se armazenar um dado em um arquivo, escolhe-se o elemento *Write to Spreadsheet File* e provê-se a devida conexão do mesmo. As conexões principais deste elemento são:

**Path File** : onde se insere o nome do arquivo e o diretório do mesmo;

**1D ou 2D data** : entrada para a variável *Arrays* tipo 1D ou 2D.

**Append to File** : anexa novos dados em arquivos existentes ou *True* abre novo arquivo.

**Transpose** : transpõe os dados 1D ou 2D antes de armazenar, ou seja, arquiva em colunas.

As demais conexões são opcionais. O **Path File** pode ser uma constante ou uma entrada de controle a ser editada no painel. Deve-se lembrar que este elemento converte os números em equivalentes caracteres ASCII antes de armazenar. Nesta operação deve-se verificar forma de indicação do ponto decimal (ponto ou vírgula) antes de se carregar estes dados em outro programa. O Matlab, por exemplo só entende a separação decimal com ponto.

## 5.6. INTERAGINDO COM O MATLAB POR MEIO DO LABVIEW : MATLAB-SCRIPT39

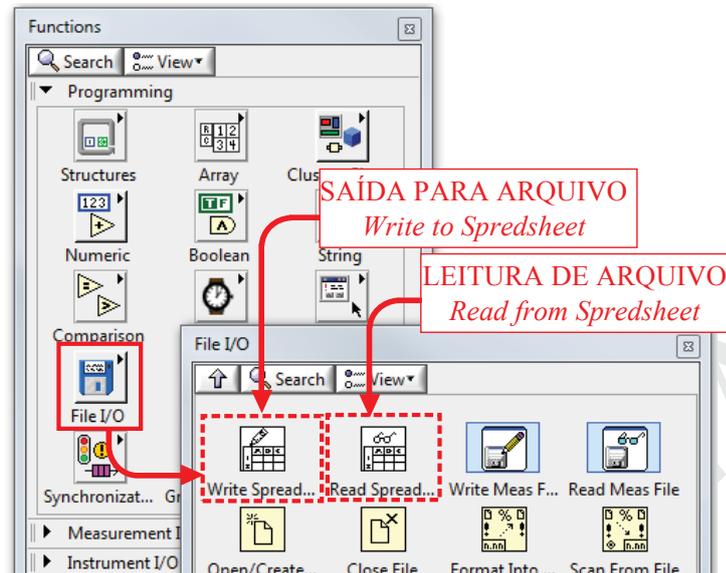


Figura 5.8: Funções de Entrada e Saída em Arquivos

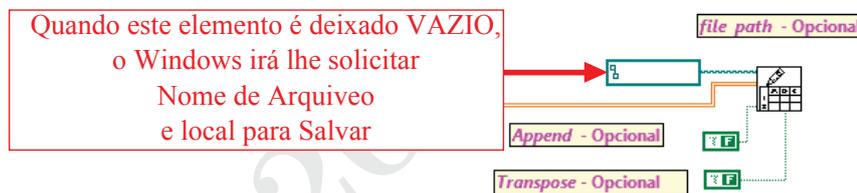


Figura 5.9: Exemplo de Saída de Dados com *Write-to-Spreadsheet*

## 5.6 Interagindo com o Matlab por meio do LabVIEW : MATLAB-Script

Este recurso, *MATLAB-Script*, permite que se executem comandos independentes do Matlab ou ainda rotinas completas editadas com a sintaxe do Matlab e armazenadas em arquivos com extensão *.m*. Este recurso faz parte das ferramentas de Fórmulas de Análise Matemática, tal como o *Formula-Node* vista na aula anterior.

A estrutura *MATLAB-Script* pode apenas receber um dado do LabVIEW e executar um comando, pode apenas gerar um dado para o LabVIEW ou pode receber um dado do LabVIEW, executar um comando (ou rotina) Matlab e retornar dados para o LabVIEW. Portanto esta estrutura pode ser configurada com apenas uma entrada, ou com apenas

uma saída ou com entradas e saídas operadas exatamente como no *Formula-Node*.

Este recurso pode ser encontrado na janela DIAGRAMA via Pop-Menu *Functions - Mathematics - Scripts and Formula* segundo a ilustração da janela DIAGRAMA da figura 5.10.

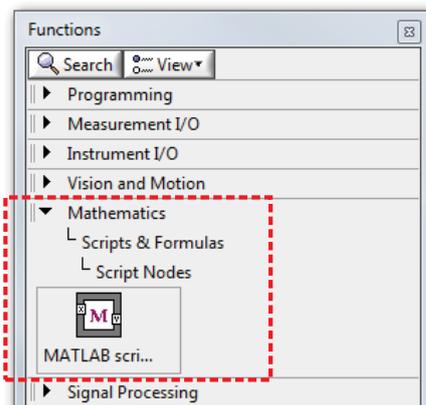


Figura 5.10: Função *Script-Node* do Matlab

Ao se selecionar o elemento *MATLAB-Script* tal com acima e colocá-lo na janela DIAGRAMA surgirá uma janela de *MATLAB-Command-Window* semelhante à janela de comandos do Matlab. O programa Matlab não precisa estar carregado na memória do Micro, mas o micro precisa ter uma instalação do Matlab.

Os dados gerados e manipulados na janela de comando via LabVIEW não interferem com a janela e dados do programa Matlab se o mesmo estiver operando ou for acionado. Quando o Matlab estiver em operação ocorrerão duas janelas de comando e que não interferem entre si.

Para que os dados gerenciados pelo LabVIEW com o *MATLAB-Script* esteja realmente disponível no programa Matlab é preciso salvar os dados de interesse em formato .MAT ou em arquivo ASCII com extensão .M.

No exemplo a seguir, a variável “Entra” gerada no LabVIEW é processada pelo comando raiz-quadrada (`sqrt( )`) do Matlab e depois é salva no arquivo `Arq_Sai.m` com o comando `save` do Matlab.

## 5.7 Atividades de aplicação

Procure executar os VIs com as funções solicitadas a seguir, sem utilizar recursos de temporização que serão vistos na próxima aula. Procure treinar também a geração e

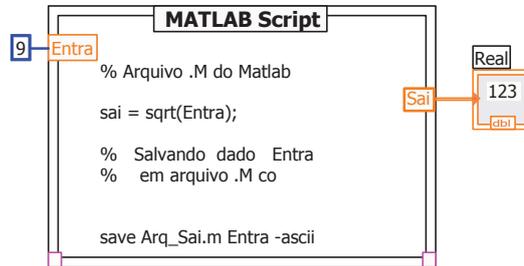


Figura 5.11: Exemplo de *Script-Node* do Matlab

gerenciamento do VI por meio de um PAINEL equivalente dotado de entradas, saídas e mensagens *strings*.

i) - Elabore um PAINEL com um indicador tipo *Chart* e conecte a um controle deslizante escalar. Teste neste exemplo simples as opções de recursos, tais como indicação gráfica (linhas cheias, pontilhadas, pontos simples, cores, reticulado etc).

ii) - Altere o VI anterior inserindo uma estrutura em *Loop* gerando uma variável *Array* e teste a mesma no indicador *Chart*.

iii) - Use um elemento de Agrupamento (*Bundle*) dentro da estrutura de *Loop* e *Build-Array* fora do *Loop* e indique os resultados em um *Chart*. Teste as opções de gráfico com as propriedades *Overlay* e *Stack Plots*.

iv) - Usando alguma estrutura de *Loop*, gere um VI com um indicador *Graph* simples.

v) - Experimente agregar um segundo *Array* ao indicador do item ( iii ).

vi) - Gere um VI com um indicador *Graph* dotado de arranjo do eixo X usando *Bundle* tal como indicado anteriormente. (Use um *Bundle* fora do *Loop*).

vii) - Gere um VI com duas variáveis do tipo *Array* e use um indicador *Graph-XY* para expressar ambos *Arrays*. (Use um *Bundle* dentro do *Loop*).

viii) - Use um do VIs anteriores e teste a armazenagem dos dados em arquivo. Tente abrir este arquivo no Matlab, plote os pontos do arquivo e compare os gráficos do LabVIEW e do Matlab.

42LABORATÓRIO 5. SAÍDAS GRÁFICAS E PARA ARQUIVO NÃO TEMPORIZADAS

**Desafio Prática 5**

ix) - Executar um VI para gerar 3 ciclos de uma senoide com 20 pontos por ciclo. Indicar a senoide numa janela gráfica de PAINEL

x) - Salvar em um arquivo de dados a senoide do item (ix) . Importar o arquivo no MATLAB, gerar os mesmos 3 ciclos de senoide com 100 pontos no Matlab e comparar graficamente os dois resultados.

USE 'x' OU 'o' PARA INDICAR OS PONTOS IMPORTADOS DO LABVIEW

OBS - OS ITENS ( ix ) E ( x ) DEVERÃO CONSTAR NO RELATÓRIO 1.  
Indicar VI e procedimento Matlab/Simulink e Resultado

## Laboratório 6

# ESTRUTURAS TEMPORIZADAS

### 6.1 Objetivos

Estudar e investigar os recursos de temporização com *timers* de tempo real em estruturas de sequenciamento de tarefas, tipo SEQUENCE e de *Loops* tipo WHILE e FOR.

### 6.2 Generalidades

Para propósitos de controle e em muitas atividades de supervisão de sistemas é essencial a disponibilidade de coordenação de rotinas e tarefas sendo realizadas em tempo real. Tempo real neste conceito significa realizar uma tarefa em períodos exatos de tempo contados em segundos.

A geração de tempo real normalmente é feita através de um *Timer (hardware)* ou contador digital de 32 Bits, o qual recebe um *clock* de uma unidade estável de tempo, como por exemplo um Cristal ou outro dispositivo de frequência estável.

No sistema em uso no Laboratório existem duas fontes de *clock* que podem ser utilizadas. Uma delas é o *clock* do *Host* ou seja, do próprio microcomputador e a outra fonte é disponível da própria placa o *LabAcquisition*.

Quando os *timers* ou contadores não estão disponíveis, alguma placa adicional deve ser incorporada para se gerar o *clock* necessário. Além dos contadores e *timers* é necessário providenciar algum meio de se programar estes componentes de modo a se especificar os desejados tempos de operação. Normalmente a operação dos *timers/counters* produzem interrupções a nível de *hardware* ou de *software* em forma de *loops* de espera e então coordenam a execução e rotinas de interrupção e sequenciamento.

No LabVIEW são disponíveis várias maneiras de se estabelecer algum tipo de sequenciamento temporizado em tempo real através de sub-VIs disponíveis em *Pop-Menus*. Neste menu são disponibilizadas funções básicas que podem ser utilizadas para executar ou implementar funções de temporização mais elaboradas. As funções básicas de temporização possibilitam ainda utilização ou geração de variáveis de data contendo Dia/Mês/Ano incluindo hora/minuto/segundo e mesmo conversão destas grandezas entre formatação numérica e caracteres.

O acesso aos recursos de temporização é obtido no Pop-Menu da janela DIAGRAMA FUNCTIONS-PROGRAMMING *Timing*. As opções de interesse neste Sub-Menu são: *Tick Count (ms)*, *Wait (ms)* e *Wait Until Next (ms) Multiple*. Estas três formas básicas são vistas na primeira linha da figura 7.1. Na linha seguinte se encontram os elementos tipo sub-VIs para manipulação de variáveis tipo data.

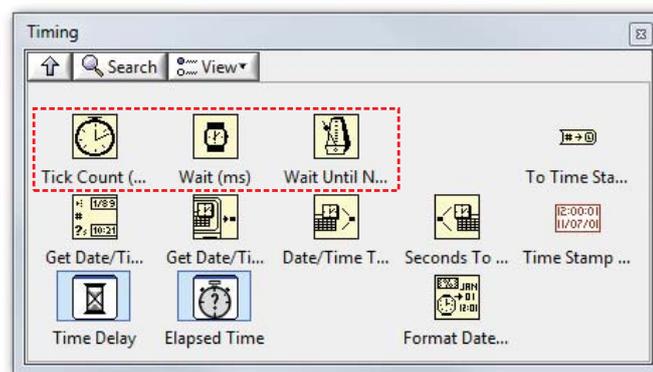
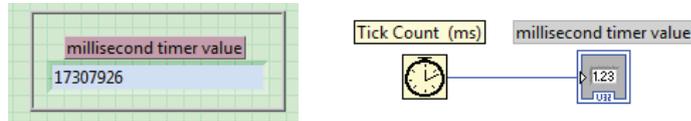


Figura 6.1: Recursos para Temporização do Menu *Functions*

### 6.2.1 Opção *Tick Count (ms)*

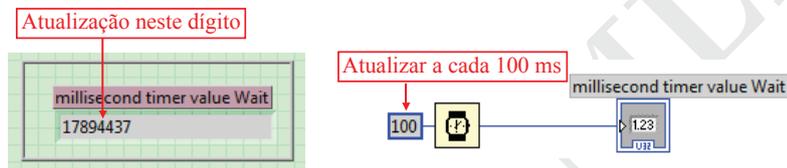
Este elemento ou sub-VI fornece a leitura da contagem do *Timer* interno do microcomputador ou *Host*. Não há como inicializar este elemento, pois a leitura corresponde ao valor de contagem atual desde que o *Host* foi ligado. Como *Timer* interno é de 32 Bits, o valor deste elemento só ira indicar zero ao contar de  $(2^{32} - 1)$  para 0. Portanto deve-se tomar cuidado ao utilizar este elemento.

A indicação numérica fornecida na saída deste elemento revela o total de milisegundos e portanto é representado por um valor numérico grande ou seja, com muitos dígitos.

Figura 6.2: Exemplo do *Tick Count (ms)*

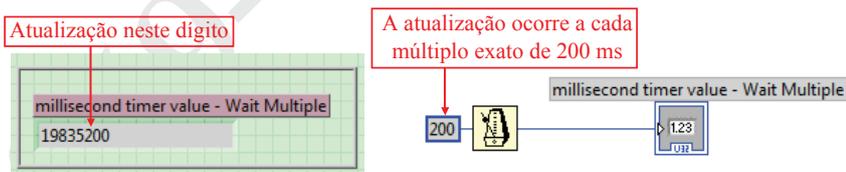
### 6.2.2 Opção *Wait (ms)*

Este elemento usa uma entrada em que se determina um valor em milissegundos até que este elemento atualize sua saída. Desta forma se a entrada for estabelecida em N milissegundos, este elemento conta de N em N milissegundos.

Figura 6.3: Exemplo do *Wait (ms)*

### 6.2.3 Opção *Wait Until Next (ms) Multiple*

Este elemento tem praticamente a mesma função do caso anterior, porém ele indicará e executará uma tarefa ou Loop a cada múltiplo exato do valor conectado à entrada.

Figura 6.4: Exemplo do *Wait Until Next Multiple (ms)*

Em geral este elemento não é o mais indicado para se estabelecer tempo real em um *Loop*, pois a primeira contagem de tempo até o primeiro múltiplo exato desejado poderá ser diferente do intervalo de tempo inicializado. Em ambos os casos, ou seja, no caso de *Wait* e *Wait Multiple* uma escala de tempo com início em zero segundos deve ser obtida utilizando-se o valor da iteração ( $i$ ) do *Loop FOR* ou *WHILE*, multiplicado pelo valor de espera conectado nas entradas dos elementos *Wait* ou *Wait Multiple*.

### 6.2.4 Opção *Time Delay*

Este elemento opera exatamente como o caso anterior, porém a inicialização é feita na janela de propriedades cliclando-se duas-vezes no ícone



Figura 6.5: Exemplo do *Delay Time (ms)*

### 6.2.5 Opções de Data e Manipulação

Este recurso pode ser útil na documentação ou simples indicação de atualização em painel. Um exemplo é indicado a seguir onde se requer apenas a indicação de uma data momentânea.

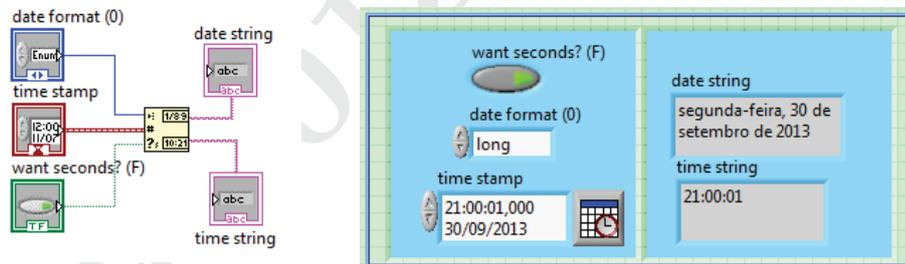


Figura 6.6: Exemplo do recurso de Manipulação de Datas

## 6.3 Atividade de aplicação

Procure executar os VIs com as funções solicitadas a seguir. Procure treinar também a geração e gerenciamento do VI por meio de um painel equivalente dotado entradas, saídas e mensagens *strings*.

- i) - Implemente e teste os recursos de geração e manipulação de datas;

ii) - Teste o elemento *Tick* (ms) , *Wait* (ms) e *Time Delay* sem utilizar *loops*. Execute o VI na forma “Uma Vez” e “Continuamente”.

iii) - Gere um VI com uma estrutura de *Loop WHILE*, insira um elemento *Wait* devidamente inicializado e execute alguma operação. Colete o resultado indexadamente em uma variável.

iv) - Use agora no VI anterior um elemento *Wait Multiple* e troque a forma de *Loop* para FOR e repita usado *Time Delay*.

v) - Incremente o VI de um dos itens anteriores com um painel dotado de uma saída gráfica.

vi) - Experimente inserir um *Loop WHILE* e altere-o para um *Timed Loop* usando fonte de *clock* interna.

#### Desafio Prática 6

vii) - Implemente em um dos VIs dos itens anteriores um procedimento para se gerar 3 ciclos de 60Hz com 20 pontos por ciclo. Cada grupo deverá gerar senoides com amplitude correspondente ao número do grupo.

viii) - Exporte estes pontos em arquivo ou use o recursos do MATLAB *Script*. Importe no Matlab os dados gerados no LabVIEW, gere no Simulink o mesmo sinal com 100 pontos e compare no mesmo gráfico os dois resultados.

Salve o VI atual para ser utilizado na aula seguinte.

OBS - OS ITENS ( vii ) E ( viii ) DEVERÃO CONSTAR NO RELATÓRIO 1.  
Indicar VI e procedimento Matlab/Simulink e Resultado

## Laboratório 7

# TRATAMENTO DE SINAIS ANALÓGICOS - ENTRADA E SAÍDA POR *HARDWARE*

### 7.1 Objetivos

Estudar e investigar os recursos de tratamento de sinais analógicos como entrada e saída por meio do LaVIEW/*LabAcquisition*.

### 7.2 Generalidades

Como já esclarecido no início do curso, o sistema disponível no laboratório é dotado de um *hardware* que proporciona a entrada e saída de dados do microcomputador (*Host*) nas mais variadas formas. Estes recursos são da maior importância na realização de procedimentos de supervisão e controle de processos.

Os assuntos abordados até agora foram de caráter essencialmente relacionado ao *software* LabVIEW. Todos os exercícios executados anteriormente independem do *hardware* de aquisição instalado ou não.

Nos exercícios subsequentes e nas tarefas de realização de controle e supervisão de processos, um *hardware* com protocolo configurado pelo LabVIEW é essencial para operação e funcionamento do VI.

O *hardware* disponível e suas características foram apresentados e discutidos na

primeira aula, sendo que uma descrição mais detalhada é encontrada no arquivo PDF denominado PCIe 6321 disponível no site de Disciplinas *On Line*.

A placa disponível possui diversos recursos, porém somente parte deles serão necessários na realização das tarefas de controle proposta neste curso. Das 16 entradas analógicas somente 2 estão devidamente configuradas e isoladas com os elementos 5B41. Das duas saídas analógicas estaremos utilizando a saída denominada OUT\_1. As saídas e entradas digitais serão alvo apenas de demonstração dos modos de operação, mas não serão necessárias na execução dos controladores propostos.

Nas etapas seguintes será dada uma breve descrição de uso das entradas e saídas de sinais por meio da placa instalada no computador configurada como sendo o *LabAcquisition* da *National Instruments* e dos diversos meios de se solicitar e configurar a operação do módulo de *hardware*.

Todas as formas de entrada e saída de dados são acessadas na janela diagrama pelo *Pop-Menu* FUNCTIONS. Neste *Pop-Menu* é disponível um sub-menu *Measurement I/O*. Caso esta opção não esteja visível, deve-se expandir o *Pop-Menu* FUNCTIONS. Dentro do sub-menu *Measurement I/O* acessa-se o sub-menu *DAQmx - Data Aquisition*.

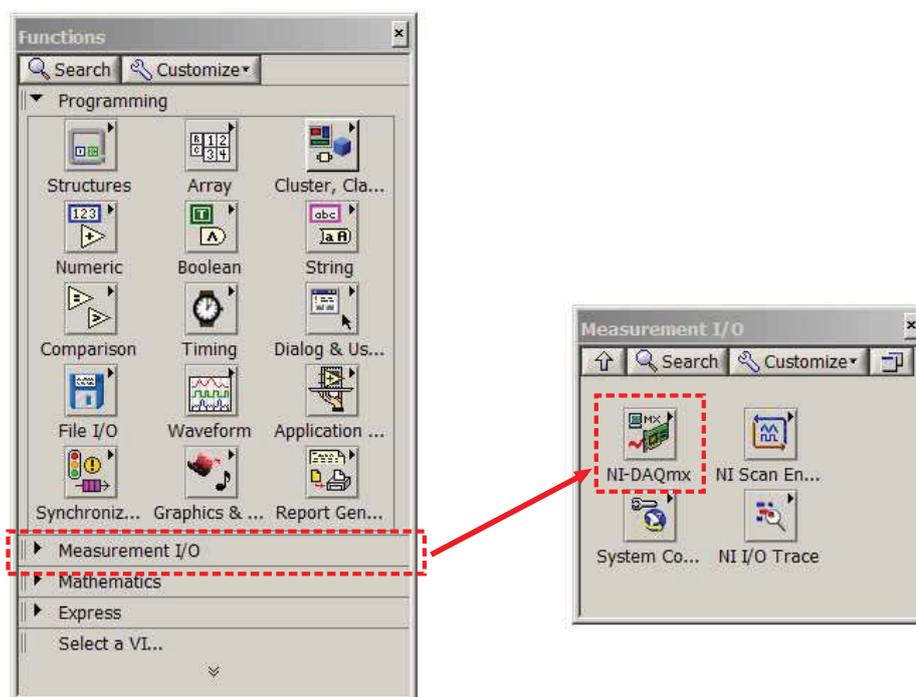


Figura 7.1: Menu de funções DAQ

Na janela seguinte do conteúdo do sub-menu *DAQmx - Data Acquisition* visualiza-se as diversas opções de Sub-VI's que coordenam as várias formas de aquisição, tal como indicado na figura 7.2. Dentre as várias opções, estaremos interessados em apenas algumas tais como : *Task Const*, *Channel Const*, *Create Channel*, *Read* e *Write*. Os demais módulos disponíveis servem para geração dedicada de tarefas de entrada ou de saída.

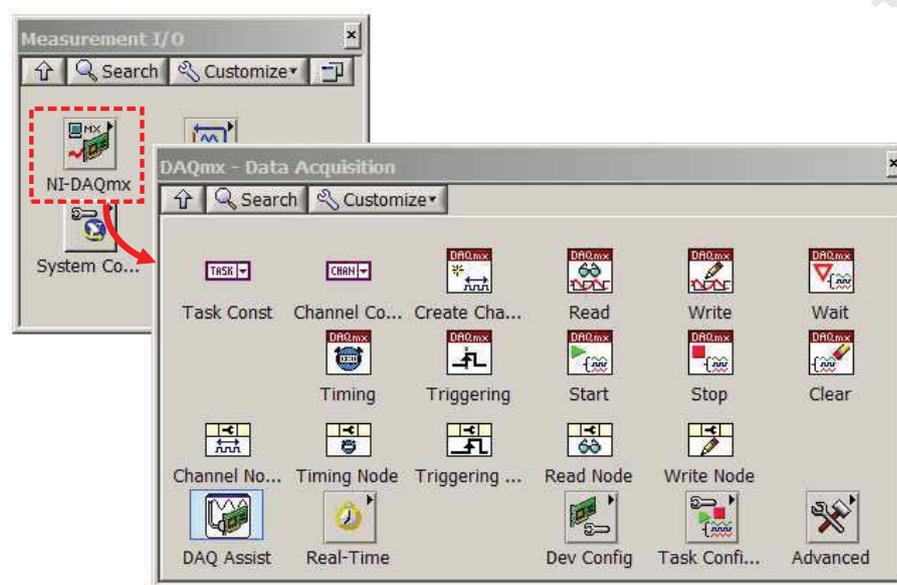


Figura 7.2: Menu de opções de I/O analógicas e digitais

Os módulos *Task Const*, *Channel Const* e *Create Channel* são utilizados para configurar a aquisição/geração de sinais (entrada ou saída) escolhendo-se a placa, o canal a ser utilizado, modo de medição limites etc. Os módulos *Read* e *Write* especificam em que instância do programa são executadas medidas de entrada ou geração de saída respectivamente.

### 7.2.1 Entrada Analógica e VIs de leitura analógica

Em função do elemento de isolamento 5B41 os sinais analógicos de entrada podem excursionar com amplitudes entre  $\pm 10$  V. A leitura do valor é feita pelo conversor A/D instalado na placa que opera com amplitudes  $\pm 5$  V, portanto como já dito ocorre uma atenuação de sinais na entrada.

O acesso e modo de operação do A/D são estabelecidos pelo usuário de acordo com a aplicação necessária. A leitura pode ser isolada a cada comando/seqüência ou pode

ser periódica e para isto diferentes formas de programação devem ser desenvolvidas. Para leituras aperiódicas, define-se a tarefa-configuração (*Task Const*) e executa-se a leitura-aquisição do sinal desejado dentro de uma determinada sequência com ou sem temporização. Leituras periódicas em geral são acompanhadas de algum recurso de temporização com uso de *loops* tipo FOR ou WHILE. Neste caso a geração da tarefa-configuração (*Task Const*) é localizada fora do *loop* e o módulo *Read* é posicionado dentro do *loop* utilizado.

### Configuração da aquisição

A configuração da aquisição é realizada por meio do módulo *Create Channel* o qual requer pelo menos a indicação de qual dispositivo deseja-se acionar, o canal a ser acessado neste dispositivo e o tipo de referência. Os demais itens disponíveis neste módulo são opcionais. A figura 7.3 indica o ícone do módulo *Create Channel* e suas conexões, em realce em verde para as conexões essenciais, que são: "*Input Terminal Configuration*", "*Physical Channels*" e "*Task Output*". A conexão *Input Terminal Configuration* determina o modo de leitura com relação ao referencial do sinal analógico, sendo que neste caso escolhe-se a opção RSE, que é o modo em que o conector 5B41 está sendo usado. A conexão *Physical Channels* determina qual o canal a ser usado com entrada do sinal a ser adquirido, que no caso da placa disponível poderia ser qualquer um dos 16 canais analógicos. Entretanto para os propósitos deste curso somente foram conectados em hardware os canais 0 (zero) DEV1/AI0 e 1 (um) DEV1/AI1. A conexão *Task Output* é a saída da configuração que deverá ser usada então no módulo de leitura READ.

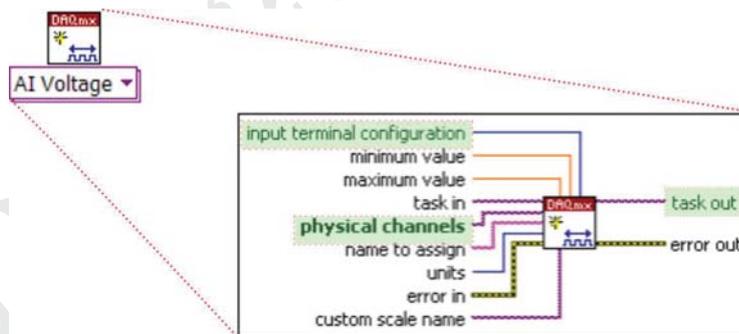


Figura 7.3: Conexões do módulo *Create Channel*

No caso padrão este módulo é do tipo *AI-Voltage* tal como visto na figura 7.4 com as demais opções de configuração de leitura de tensão, corrente temperatura, etc. As demais opções de entradas e/ou saídas, sejam analógicas ou digitais, podem ser acessadas no *List-Menu* e posteriormente procede-se a configuração.

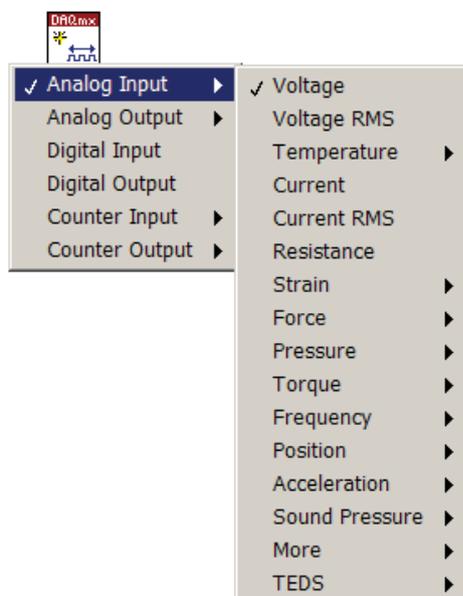


Figura 7.4: Opções de configuração de entrada analógica do módulo *Create Channel*

Para se inserir qualquer valor nas conexões do módulo *Create Channel*, o modo mais direto é posicionando-se o mouse próximo da ligação desejada e acionando-se o botão esquerdo do mouse para exibir o *pop-menu* de opções. A partir daí escolhe-se *Create-Control* ou *-Constant*. Esta forma é mais interessante, pois o próprio LabVIEW insere uma constante na janela DIAGRAMA ou uma entidade de controle na janela PAINEL com a respectiva formatação seja ela numérica, *String* ou *Booleana*.

Como já esclarecido o módulo *Create Channel* necessita no mínimo 2 definições de configuração que são a conexão do *input terminal conf* que determina a ligação de referencial usado no *rack 5B* e o canal a ser usado (0 ou 1). Com estas conexões executadas a variável *task out* pode ser direcionada ao módulo READ, esteja ele em um sequência de execução única ou em *loop*.

### Configuração da Leitura de Sinal Analógico

A realização da leitura de qualquer sinal analógico é feita com módulo *Read* e deve ser precedida da geração de uma entidade *task out* que determina o que vai ser lido e em que canal do dispositivo de aquisição. A forma padrão do módulo *Read* é do tipo *Analog-DBL-1.Chann - 1.Sample* que significa amostragem de um canal analógico, 1 única amostra e formato *Double-Precision*. Esta forma é vista na figura 7.5.

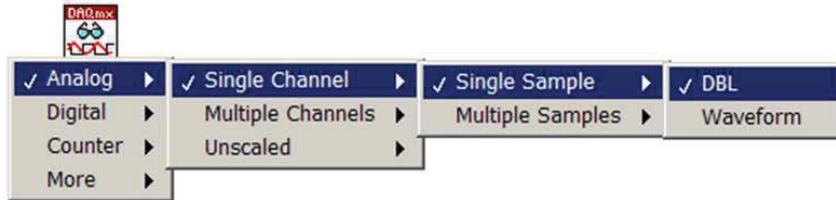


Figura 7.5: Opções de configuração de entrada analógica do módulo *Read*

## 7.2.2 Saída analógica e VIs de escrita

O *hardware* disponível permite produzir duas saídas analógicas a partir de um VI desenvolvido no *software* LabVIEW. Assim como os VIs de leituras, os VIs de escrita representam os modos de programação que configuram e operam os dois conversores D/A da placa. No *Rack 5B* está disponível o canal designado CH.1.

OS módulos de saída analógica *Write* podem ser acessíveis da mesma forma que os de leitura na janela FUNCTIONS - MEASUREMENTS I/O - DAQMX DATA ACQUISITION. Os módulos *Write* só podem ser usados após a definição de um *Task Out* definindo a tarefa de geração de sinal analógico, a qual deve conter a definição do canal do dispositivo, canal e modo referencial.

O Módulo *Write* possui as mesmas propriedades do caso *Read* visto na figura 7.5 em termo de configuração. Um exemplo completo de geração de sinal analógico no canal 1 do dispositivo Dev1 em modo RSE é visto na figura 7.6.

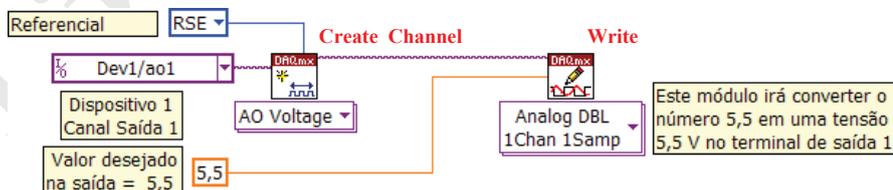


Figura 7.6: Exemplo de *Create Channel* configurado para um módulo *Write* de saída analógica

### 7.3 Atividades de aplicação

Execute os VIs descritos a seguir fazendo uso de funções de painel para a devida configuração de operação do VI.

i) - Teste o VI de entrada analógica pontual de um canal, conectando os terminais do canal escolhido a uma tensão conhecida e verifique o resultado. Execute continuamente.

ii) - Use o mesmo VI anterior em uma estrutura de *loop* não temporizado com um determinado número de iterações e mostre o resultado final em um *Chart*.

iii) - Incremente o VI anterior usando temporização e use um sinal periódico de baixa frequência. Use um gerador de sinais.

iv) - Execute um VI para gerar uma tensão fixa no canais 1 da saída analógica. Verifique com um voltímetro/osciloscópio a saída produzida.

v) - Repita o item ( iv ) para gerar uma senoide no canal de saída. Verifique a saída no osciloscópio. Fotografe a resposta.

vi) - Qual seria a frequência máxima de um sinal senoidal aceitável neste VI?

#### Desafio Prática 7

vii) - Use o VI da aula anterior e gere um sinal senoidal de 60Hz em um dos canais de saída analógica e verifique este sinal no osciloscópio. Avalie o aspecto da forma de onda e o valor real da frequência obtida.

OBS - ESTE ITEM DEVERÁ CONSTAR NO RELATÓRIO 1.

FOTOGRAFE com seu celular o resultado no Osciloscópio para documentar o relatório. Indicar VI de realização e resultado da Medição no Osciloscópio (FOTO).

## Laboratório 8

# VERIFICAÇÃO EXPERIMENTAL DO TEOREMA DA AMOSTRAGEM

### 8.1 Objetivos

Nesta prática serão exercitados alguns recursos estudados nas aulas anteriores sobre a construção de VIs com objetivo de verificação do conceito do teorema de amostragem de sinais periódicos.

### 8.2 Introdução e generalidades

Baseado nos tópicos já estudados do *software* LabVIEW e o *hardware* do *LabAcquisition* é possível a partir de agora um aprofundamento das técnicas de programação na linguagem G e também na geração de VIs mais complexos.

Existem ainda muitos outros recursos de *software* e de *hardware* disponíveis, porém não é intuito deste curso o aprofundamento em tais tópicos.

Os demais recursos de *software* e de *hardware* só fazem sentido em se estudar no caso de uma necessidade específica relacionada com alguma aplicação também específica.

Para complementar o treinamento nas técnicas de programação e de geração de VIs de aquisição é proposto nesta aula a execução de um VI composto de vários recursos de medição, indicação de resultados para verificação do teorema de amostragem.

Nesta prática estamos interessados em adquirir corretamente um sinal de frequência

e amplitude conhecida. Usaremos um Gerador de Sinais com saída senoidal de baixa frequência e a amplitude deverá ser inferior a 10V de pico.

Portanto deve-se preparar a montagem, ou seja as ligações do gerador de funções e das conexões no painel de entradas analógicas do *Lab-Acquisition*. Em seguir deve-se estipular os VIs configurando o canal de entrada e o *Loop* de medição.

Na execução destes VIs deve-se usar de forma adequada os recursos de temporização tais como estudados em aulas anteriores associados aos recursos de estruturas em formas de *Loops*.

O Teorema da Amostragem estabelece que a frequência mínima de amostragem de um sinal periódico deve ser o dobro da frequência deste sinal. Se por outro lado o sinal externo for composto por um espectro finito, a frequência de amostragem de ser o dobro da frequência máxima do espectro finito.

Propõe-se então a execução de amostragens em várias frequências incluindo uma amostragem na própria frequência do sinal de entrada. Esta amostragem na mesma frequência servirá para garantir uma melhor exatidão na frequência do sinal de teste.

### 8.3 Atividades de aplicação

Devido a certas limitações do *LabAcquisition* e como forma didática, o sinal externo não deverá exceder a frequência de 1 Hz. Quanto à amplitude do sinal deve-se respeitar os níveis do elemento de entrada 5B41, ou seja inferior a  $\pm 10V$ .

Ficará a critério de cada grupo a escolha e configuração do elemento de aquisição apropriado. O sinal proveniente de um gerador de sinais deverá ser conectado no canal 0 ou 1 do *rack* de entradas analógicas.

Inicialmente gere um VI contendo *Loop* de aquisição com uma taxa de amostragem de 1 segundo e dotado de *Chart* de saída dentro do *Loop*. Este *Loop* deverá rodar com elevado número de iterações. Após a conexão dos cabos no *rack* de entrada, execute o VI de forma contínua e ajuste o *dial* do gerador de funções na frequência de 1 Hz. Quando a frequência do gerador for realmente 1 Hz a a resposta visualizada no *Chart* será muito próxima de uma reta. Lembre-se de desabilitar a opção *Autoscale* do eixo-Y. Isto proporcionará uma melhor visualização do ajuste da senoide 1Hz. Esta é a fase de calibração apenas. Se o gerador estiver a 1 Hz e a aquisição pelo VI também a 1 Hz, a visualização da aquisição no *Chart* será uma reta.

**Desafio Prática 8**

Após a fase de calibração, exclua o *Chart* do *Loop* e determine o número de iterações para uma aquisição referente a um intervalo de 10 segundos.

Uma sugestão de execução é, definir uma estrutura tipo seqüência de 4 *frames* sequenciais e copiar o *Loop* de aquisição em cada *frame*. A seguir basta reajustar a taxa de amostragem em cada *frame* de aquisição incluindo em cada um deles um módulo de armazenagem em arquivo para salvar os dados amostrados em cada situação.

Outra solução é executar o VI de calibração 4 veze seguidamente e armazenando-se os dados.

OBS. É importante incluir no VI um recurso para salvar junto com os dados o tempo discreto  $kT_0$  para cada um dos casos de amostragem.

O sinal adquirido para as análise deverá conter a mesma janela de tempo nas várias frequências solicitadas, no caso 10 segundos no mínimo:

*Frame/Execução 1* :  $f_0 = 1\text{Hz}$  (mesma frequência)

*Frame/Execução 2* :  $f_0 = 2\text{Hz}$  (dobro da frequência = Teorema de Amostragem)

*Frame/Execução 3* :  $f_0 = 5\text{Hz}$

*Frame/Execução 4* :  $f_0 = 10\text{Hz}$

Importe os 4 arquivos no Matlab e obtenha no mesmo gráfico todos os resultados para comparação.

Cada grupo e turma terá estipulado um valor diferente de amplitude, tal que, Grupo X : Senoide com amplitude X Volts.

O primeiro relatório deverá incluir o VI, resultados e análises relativos aos DE-SAFIOS da prática 5: (senoide de 20 pontos), da prática 6: senoide 60 Hz virtual e a FOTO da prática 7 : ( geração do sinal de 60Hz em tempo real).

Da prática 8, apresentar os procedimentos de preparação e de execução do experimento, os VIs utilizados, os resultados e as análises.

NÃO IMPRIMIR *SCOPES* (SIMULINK) OU *CHARTS* (LabVIEW). SOMENTE OS *PLOTS* (MATLAB).

**DATA LIMITE PARA ENTREGA DO RELATÓRIO 1**

**DIA 06-07/10/2016 VIA MOODLE !**