



Instituto Superior Técnico

Curso de LabVIEW

Pedro M. Ramos

Pedro.Ramos@Lx.it.pt

Parte II – 21 de Outubro de 2004

IST/DEEC

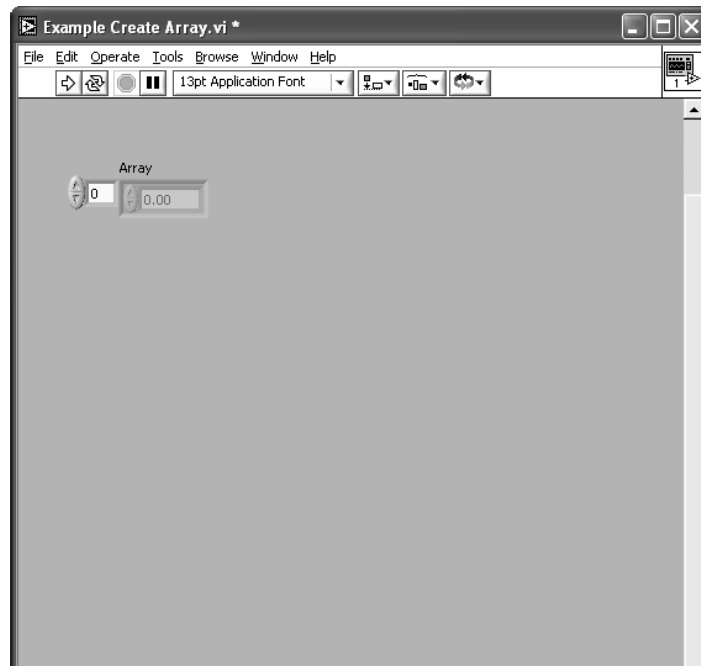
34

Arrays

- Arrays são colecções de dados do mesmo tipo
- Elementos endereçados de 0 até N-1
- Para criar um array, coloca-se uma array shell no painel frontal e coloca-se lá dentro um controlo (ou indicador) do tipo desejado

35

Criar um Control Array



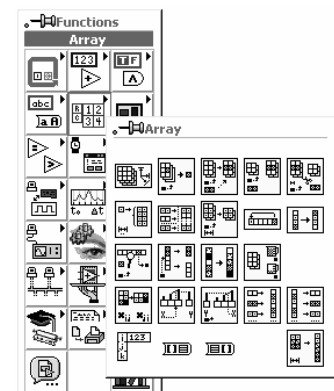
DICA: Para criar um indicador de array, em vez de um control, colocar um indicador dentro da shell

36

Funções para Arrays

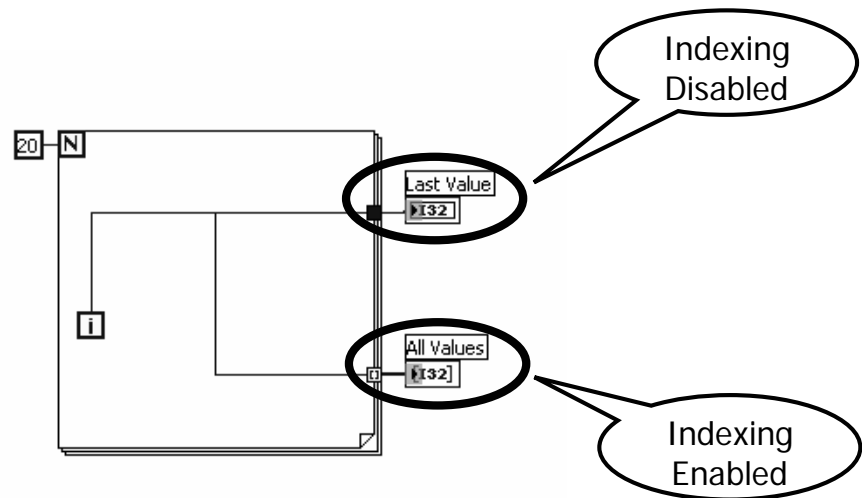
- As funções para arrays encontram-se na palette **Functions»Array** Palette e incluem entre outras, as funções:

- **Array Size**
- **Initialize Array**
- **Index Array**
- **Build Array**
- **Array Subset**
- **Array Max & Min**



37

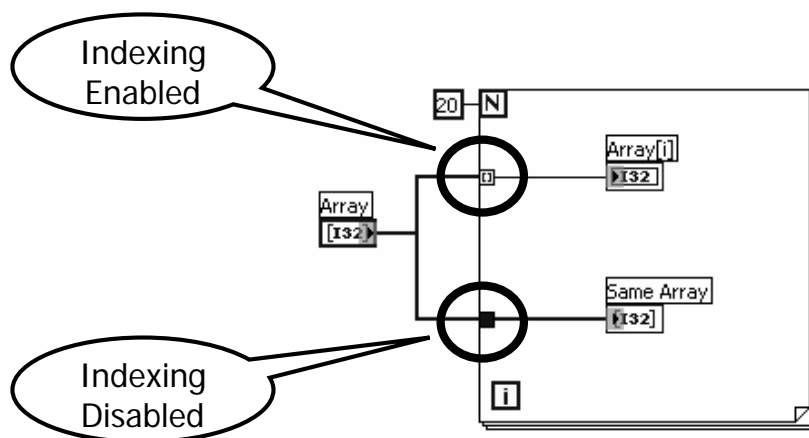
Criação de Arrays com ciclos



DICA: Para mudar o modo de indexação de saída usar o botão direito do rato sobre o túnel

38

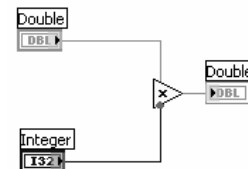
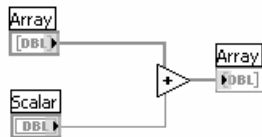
Indexação de Arrays com Loops



39

Polimorfismo

- As funções numéricas básicas do LabVIEW são polimorfas, i.e., as entradas podem ser de tipos diferentes (dentro de algumas limitações)
- Por exemplo, somar um escalar com um array tem como resultado em array
- Multiplicando um inteiro com um double, obtém-se um double



40

Clusters

- Clusters são estruturas de dados de diferentes tipos
- Equivalentes aos *records* em Pascal ou as *structures* em C
- Usados para agrupar dados relevantes e para simplificar/reduzir os fios
- Para criar um cluster, adiciona-se uma cluster shell no painel frontal e colocam-se lá dentro os controlos ou indicadores

Cada elemento do cluster tem um nome que corresponde ao controlo/indicador originalmente inserido

41

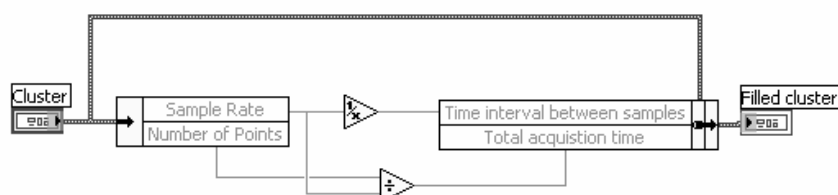
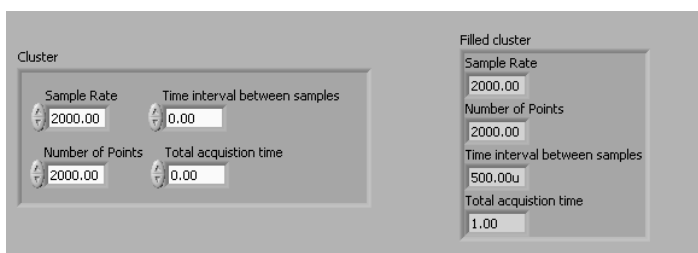
Clusters

- Para aceder ao valor de um elemento de um cluster, usa-se **Unbundle By Name** e depois selecciona-se o nome do elemento
- Para modificar o valor de um elemento de um cluster, usa-se **Bundle By Name** e depois selecciona-se o nome do elemento a modificar

DICA: Em **Unbundle By Name** e em **Bundle by Name** é possível aceder ou modificar diversos elementos do cluster. Basta adicionar uma nova entrada ou saída e seleccionar os nomes.

42

Exemplo Bundle/Unbundle



43

Cluster de Erro

- O cluster de erro é um cluster pré-definido pelo LabVIEW para gestão de erros
- Todos os VIs de comunicação têm como entrada um cluster de erro e como saída um cluster de erro
- O cluster de erro inclui os seguintes elementos:

status	[boolean]	existiu um erro ?
code	[I32]	número do erro
source	[string]	origem do erro

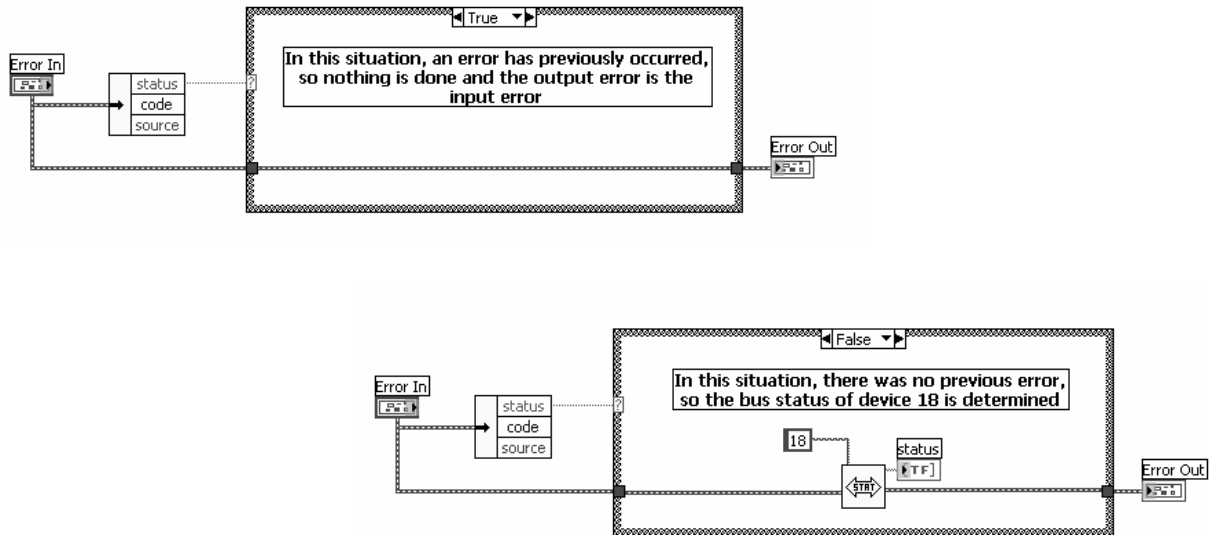
44

Cluster de Erro

- O cluster de erro **TEM** de ser usado sempre que se comunica com dispositivos
- Os subVIs de comunicação devem testar o cluster de erro de entrada para determinar a existência de erros prévios
- Se existiu um erro prévio, nenhuma comunicação deve ser executada e o cluster de saída deve ser igual ao de entrada
- Na ausência de erro prévio a comunicação deve ser executada. Se se verificar um erro, as comunicações devem ser paradas e o cluster de erro de saída é preenchido com os detalhes do erro

45

Exemplo Cluster de Erro



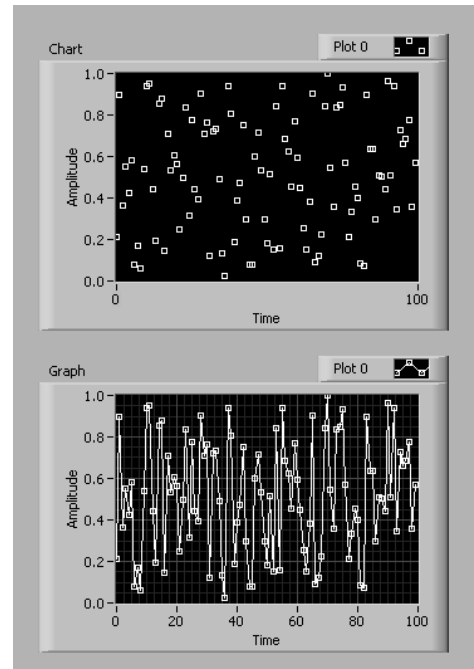
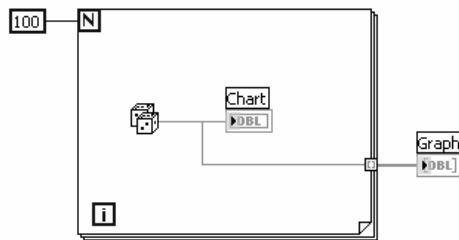
46

Gráficos

- Existem dois tipos básicos de gráficos no LabVIEW: **Charts** e **Graphs**
- **Charts** são indicadores numéricos que representam em tempo real a evolução de uma variável escalar
- **Graphs** são usados para representar arrays de números ou *waveforms*

47

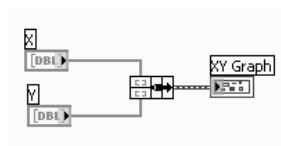
Charts vs Graphs



48

XY Graphs

- Pares de dados XY podem ser representados em **XY graphs**
- Para construir um par XY basta construir um cluster com dois elementos: o array com os pontos X e um array com os pontos Y
- Para representar múltiplos pares XY no mesmo gráfico constrói-se um array com os pares



49

Waveforms

- Waveforms são clusters pré-definidos no LabVIEW para a representação temporal de dados adquiridos uniformemente espaçados temporalmente
- Os elementos das waveforms são:
 - t0** instante de tempo do 1º ponto
 - dt** intervalo de tempo entre 2 amostras
 - Y** array de pontos

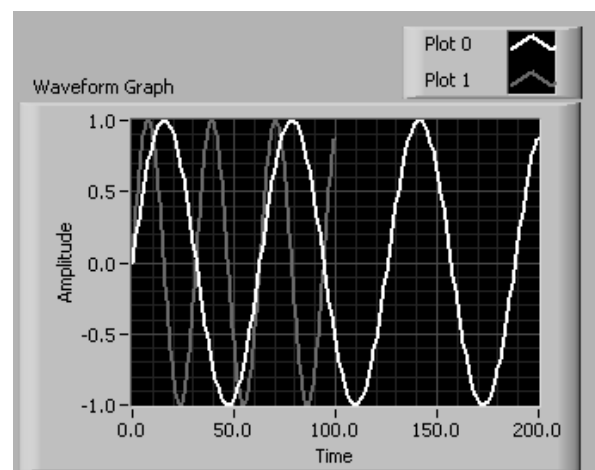
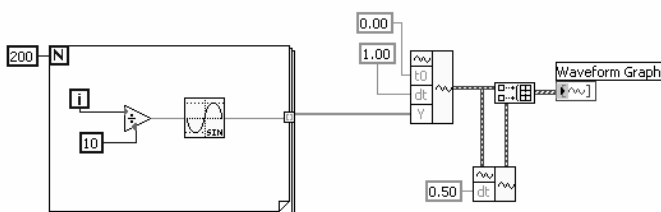
Waveform»Get Waveform Components

Waveform»Build Waveform

DICA: Para representar múltiplas waveforms no mesmo graph, criar um array com as waveforms (**Build Array**)

50

Exemplo Waveforms



51

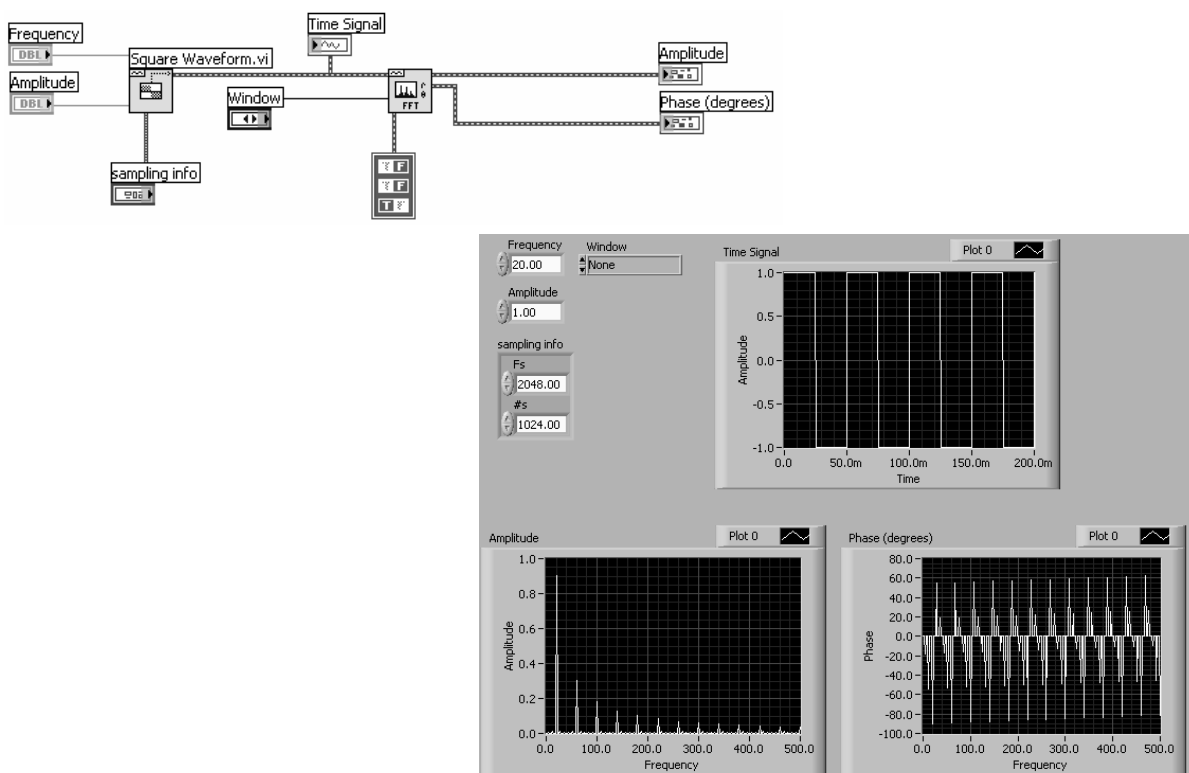
Fast Fourier Transform (FFT)

- Os VIs para as FFT estão em:
Functions»Waveform»Waveform Measurements
- As saídas são a amplitude e fase
FFT Spectrum (Mag-Phase) ou as partes reais e imaginárias **FFT Spectrum (Real-Im)**
- As entradas destes VIs são waveforms

ATENÇÃO: A janela temporal por omissão não é a rectangular!!

52

Exemplo FFT



53

Comunicação com o LabVIEW

GPIB
DAQ
RS232

54

Interface GPIB (IEEE 488.2)

- A comunicação com os dispositivos é efectuada através de um sistema de perguntas e respostas
- Cada dispositivo tem um endereço único e um conjunto específico de comandos
- Os VIs relevantes estão em
Functions»Instrument I/O»GPIB

AVISO: O cluster de erro tem de ser usado para controlar a ordem de execução (pergunta, resposta, pergunta)
Não é necessário usar uma estrutura de sequência

55

GPIB

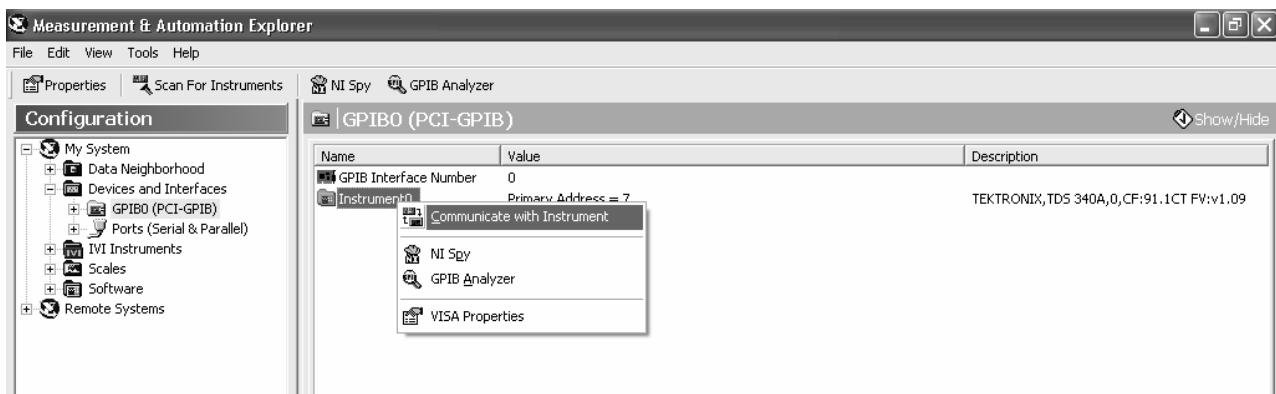
- Para mudar o endereço GPIB de um dispositivo, consultar o respectivo manual de instruções
- Para detectar o endereço de um dispositivo e testar as comunicações usar

Measurement & Automation Explorer MAX



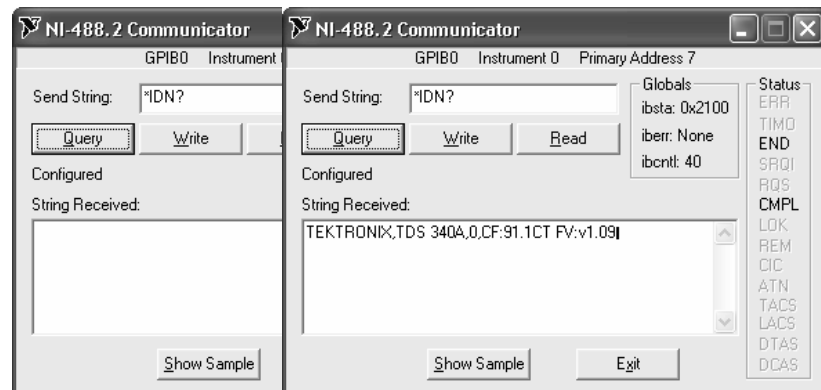
56

MAX GPIB



57

Comunicação GPIB com o MAX



DICA: Query corresponde à escrita da string e leitura da resposta do dispositivo
Write corresponde à escrita da string para o dispositivo
Read corresponde à leitura de uma string do dispositivo

58

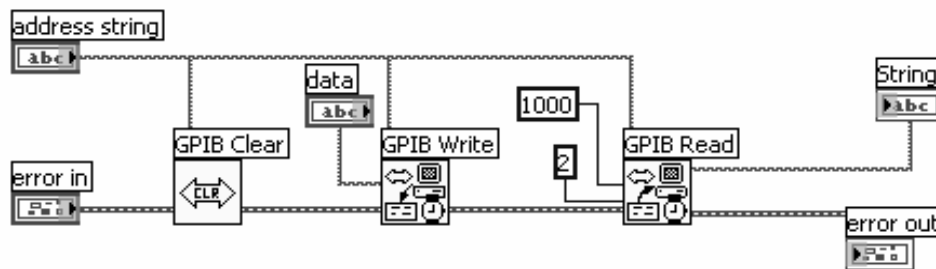
Vis de interface GPIB

- **GPIB Clear** – envia um SDC (*Selected Device Clear*) para o dispositivo (deve ser usado um por dispositivo e por programa logo no início)
- **GPIB Write** – escreve uma **string** para o dispositivo indicado
- **GPIB Read** – lê uma **string** do dispositivo com o endereço indicado. Lê o número de caracteres indicado ou até que um EOS seja recebido

AVISO: O endereço não é um número, é uma string

59

Exemplo GPIB



60

Aquisição de dados com uma DAQ

- Uma DAQ é um dispositivo de hardware capaz de executar conversões analógico-digitais das tensões de entrada a um determinado ritmo de aquisição
- Os VIs relevantes estão em:
Functions»Data Acquisition»Analog Input

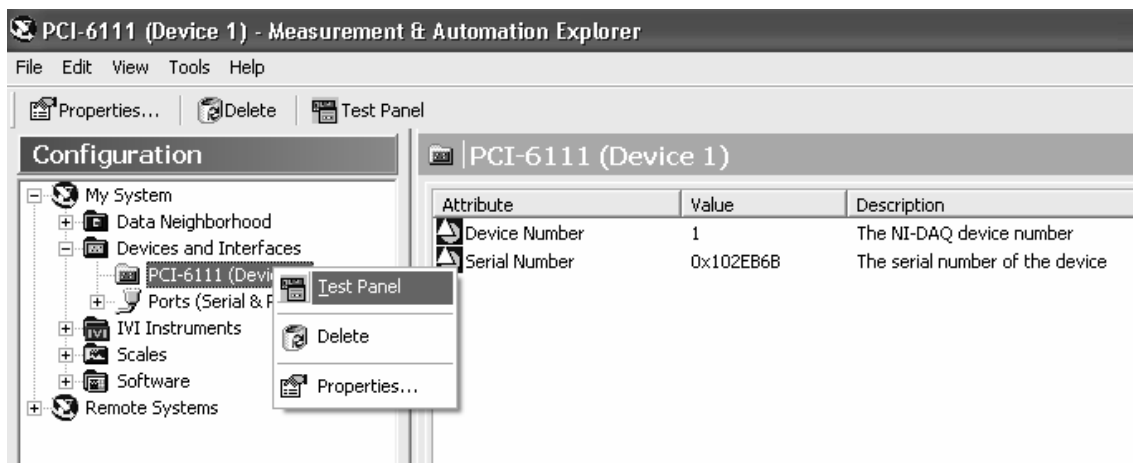
61

Aquisição de dados com uma DAQ

- Para adquirir um ponto de um canal usar:
AI Sample Channel
- Para adquirir uma waveform de um canal usar:
AI Acquire Waveform
- Para adquirir waveforms de vários canal usar:
AI Acquire Waveforms

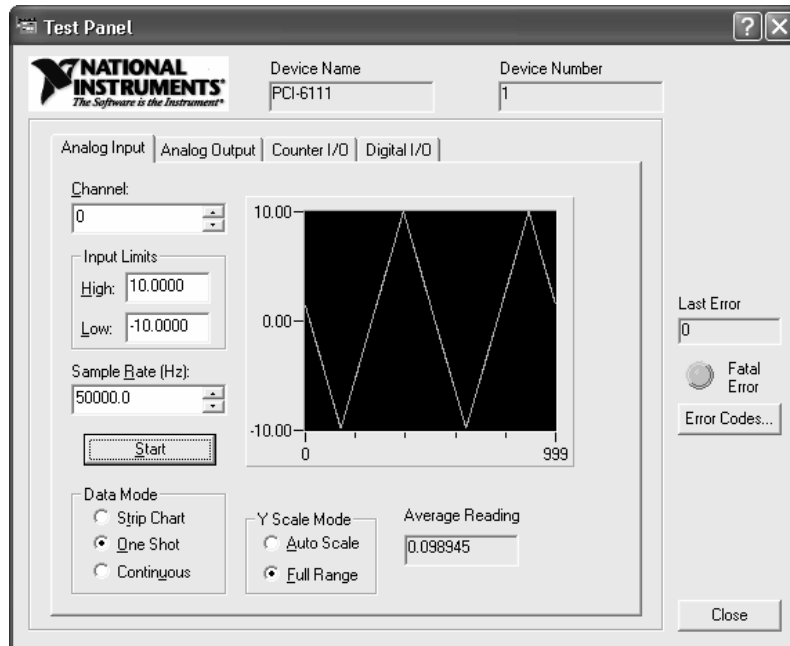
62

MAX e DAQs



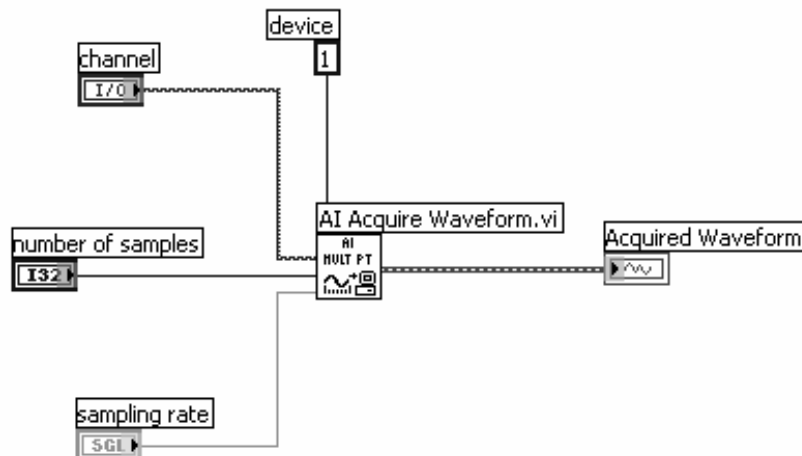
63

MAX DAQ Test Panel



64

Exemplo DAQ



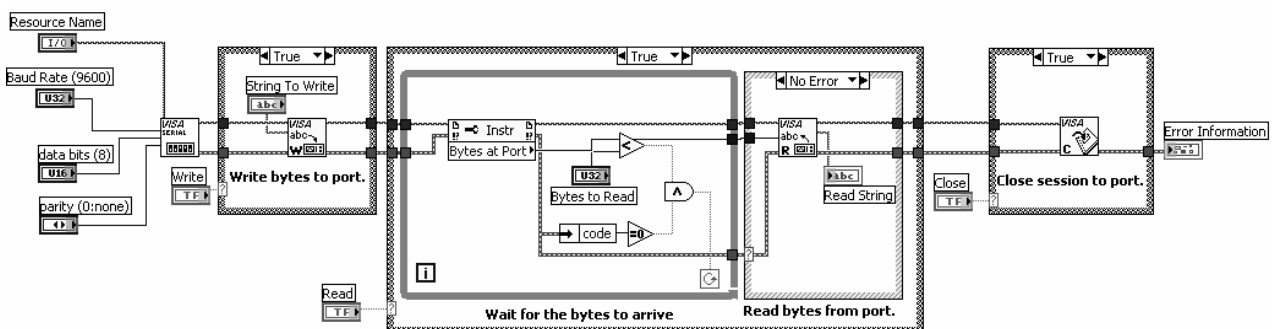
65

RS232

- Os VIs relevantes estão em:
Functions»Instrument I/O»Serial
- Tal como no GPIB, as comunicações são obtidas através de um sistema de questões (**VISA Write**) e respostas (**VISA Read**)
- A porta usada tem de estar configurada da mesma forma no LabVIEW (**VISA Configure Serial Port**) e nos dispositivos externos

66

Exemplo RS232 (VISA)



67

FIM

Ficheiros PDF da apresentação estão disponíveis no site

<http://alfa.ist.utl.pt/~instmed/iscp>