

Automatização de Laboratórios de Medida de Componentes Ópticos e Electrónicos através de LabVIEW

P. J. Silva, P. Mendes, J. Ferreira da Rocha, J. C. Pedro, P. Tavares, P. S. André, N. B. Carvalho

Resumo – Nesta comunicação descreve-se o trabalho desenvolvido no projecto de 5º Ano do curso de Engenharia Electrónica e Telecomunicações na área de automatização de laboratórios de medida. São mostradas algumas das aplicações finais desenvolvidas através da ferramenta de programação da *National Instruments™ – LabVIEW*, nomeadamente, uma bancada Pin/Pout, uma bancada I/V e outra I/P.

Abstract – This paper describes the work developed in the final project of the degree in Electronics and Telecommunications Engineering, at Aveiro University in the domain of laboratory automation. Here are shown some final applications developed with the programming tool of *National Instruments™ – LabVIEW*, such as Pin/Pout, I/V and I/P for device characterisation.

I. INTRODUÇÃO

Nos laboratórios modernos existe todo um conjunto de aparelhos, desde os tradicionais multímetros, geradores de formas de onda, osciloscópios, até outros mais sofisticados tais como fontes digitais programáveis, analisadores de espectro, analisadores de quadripólos, osciloscópios digitais, controladores de temperatura e muitos outros. Estes aparelhos, trabalhando de uma forma isolada ou interligados em sistemas de medida, todos eles executam funções idênticas: adquirem, analisam e apresentam dados.

Através das evoluções decorridas no campo da informática nas duas últimas décadas, foi possível concretizar a união entre os aparelhos existentes (*hardware*) e os novos programas desenvolvidos (*software*), os quais possibilitam a comunicação com os aparelhos de uma forma interactiva. Esta nova classe de aparelhos foi designada por instrumentos virtuais (*VI - virtual instruments*). Estes VIs, para além das três funções descritas atrás, vieram introduzir uma quarta função até aqui inexistente: a flexibilidade. Esta flexibilidade é conseguida pelo utilizador, uma vez que este pode criar bancadas de medida de acordo com os seus objectivos, combinando a plataforma de computação, o *hardware*, o *software* e os acessórios necessários, por forma a completar a sua aplicação específica. Actualmente, cientistas e engenheiros em todo o mundo, usam instrumentos virtuais como forma de medir e controlar sinais eléctricos, conseguindo desta forma uma maior eficiência no trabalho desenvolvido.

Reconhecendo a importância da automatização de medidas em laboratório com vista a uma maior eficácia dos seus grupos de investigação, este projecto enquadrou-se no esforço que o Pólo de Aveiro do Instituto de Telecomunicações tem vindo a fazer, no sentido de dotar os seus laboratórios de Óptica e de Rádio Frequência de capacidade automática de caracterização de componentes e teste de protótipos.

II. NORMA IEEE 488 (GPIB)

Em 1965, a *Hewlett-Packard* desenvolveu o *Hewlett-Packard Interface Bus* (HP-IB) como uma forma de ligar a sua linha de instrumentos programáveis aos seus computadores. Dada a sua elevada taxa de transferência na altura (cerca de 1 *Mbyte/s*), este interface rapidamente ganhou uma enorme popularidade. Mais tarde, em 1975 foi aceite como sendo o *standard* IEEE 488 [1], tendo evoluído para o *standard* ANSI/IEEE 488.1 em 1987 [2]. Hoje em dia, este interface é designado numa forma mais comum como *General Purpose Interface Bus* (GPIB) e não como HP-IB.

Também em 1987 surgiu uma nova actualização, tendo-se criado o *standard* ANSI/IEEE 488.2 [3] fazendo com que o *standard* inicial, como que fosse "estrangulado", uma vez que agora eram definidas com precisão as formas como os controladores e os instrumentos comunicavam. O consórcio *Standard Commands for Programmable Instruments* (SCPI) pegou nas estruturas de comandos definidos na norma IEEE 488.2 e criou um único *set* de comandos de programação compreensíveis que passariam a ser usados em qualquer instrumento SCPI. A Fig. 1 mostra-nos a evolução histórica do desenvolvimento desta norma.

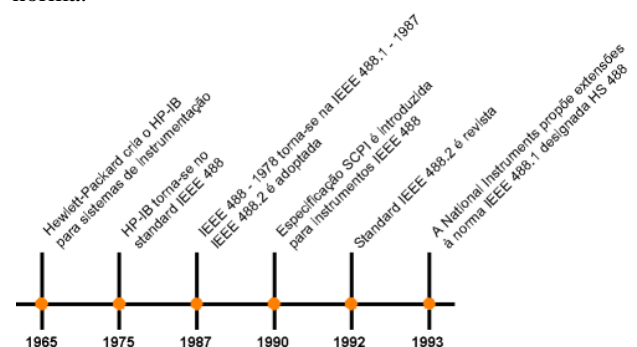


Fig. 1 - Desenvolvimento histórico do GPIB.

Os dispositivos GPIB comunicam entre si enviando mensagens dependentes do dispositivo e mensagens de interface através do sistema de interface.

- **Mensagens dependentes do dispositivo;** Normalmente designadas como dados, contêm informações específicas do dispositivo tais como instruções de programação, resultados de medida, estado do aparelho e ficheiros de dados.
- **Mensagens de interface;** Estas mensagens gerem o barramento. Normalmente designadas por comandos, as mensagens de interface desempenham funções tais como inicializar o barramento, endereçamento dos dispositivos bem como alterar o modo dos dispositivos para remoto ou local.

Os dispositivos podem ser rotulados de *talkers*, *listeners* e/ou *controllers*.

- Um *talker* envia mensagens de dados para um ou mais *listeners*.
- Um *listener* recebe os dados enviados nas mensagens do *talker*.
- Um *controller* gere o fluxo de informação no GPIB, enviando comandos a todos os dispositivos.

O GPIB funciona de forma análoga à de um barramento de computador, apenas com a particularidade que enquanto neste último os dispositivos são interligados através de uma *backplane*, o GPIB possui dispositivos independentes interligados por cabos *standard GPIB*.

O sistema de interface GPIB consiste em 16 linhas de sinal e 8 linhas de massa de retorno. As 16 linhas de sinal estão divididas em 3 grupos funcionais; 8 linhas de dados, 3 linhas para *handshake*, e 5 linhas de gestão de interface (ver Fig. 2).

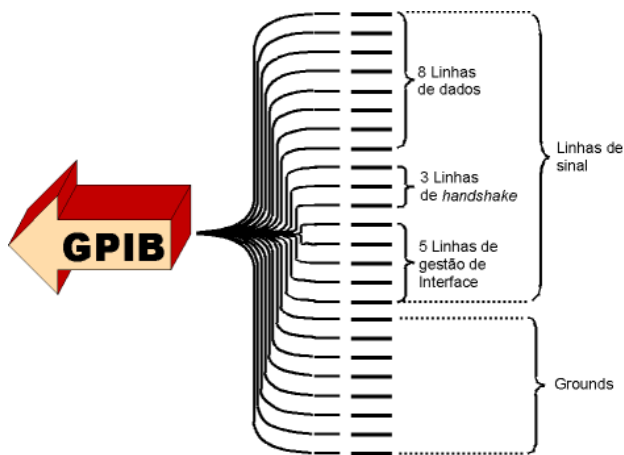


Fig. 2 - Sinais e linhas GPIB.

As 8 linhas de dados, DIO1 a DIO8, transportam tanto mensagens de dados como mensagens de comando. A linha de estado de atenção (ATN) determina se o barramento está na presença de dados ou de comandos. Todos os comandos bem como quase todos os dados utilizam uma codificação ASCII ou ISO de 7 bits, na qual

o oitavo bit, DIO8, não é utilizado ou é usado para paridade.

As 3 linhas de *handshake* controlam de um modo assíncrono a transferência de mensagens entre os dispositivos. A este processo dá-se o nome de *3-wire interlocked handshake*. Este protocolo garante que os bytes das mensagens são enviados e recebidos através das linhas de dados sem erro de transmissão através da utilização de 3 tipos de sinalização: NRFD (*not ready for data*), NDAC (*not data accepted*) e DAV (*data valid*).

O fluxo de informação no interface é gerido através de 5 linhas de gestão do interface, conforme se descreve a seguir;

- **ATN (*attention*)** - O controlador coloca o ATN no estado lógico 'V' no caso em que utiliza as linhas de dados para enviar comandos e coloca o ATN no estado lógico 'F' quando o *talker* pode enviar mensagens de dados.
- **IFC (*interface clear*)** - O controlador utiliza esta linha para fazer a inicialização do barramento e para se colocar no estado CIC (*controller-in-charge*).
- **REN (*remote enable*)** - O controlador utiliza esta linha para colocar os dispositivos em modo de programação remoto ou local.
- **SRQ (*service request*)** - Qualquer dispositivo pode utilizar a linha de SQR para assincronamente solicitar um serviço ao controlador.
- **EOI (*end ou identify*)** - Esta linha possui duas utilidades distintas. O *talker* utiliza o EOI para marcar o final da *string* de mensagem. O controlador utiliza o EOI para mandar responder os dispositivos num *poll* paralelo.

Cada dispositivo GPIB ligado ao barramento deve ter um endereço único, entre 0 e 30, definido no próprio dispositivo através de *jumpers*, *microswitches*, *firmware*, etc. Os dispositivos que utilizam o interface GPIB, são interligados através de um cabo blindado de 24 condutores com um conjunto de fichas macho/fêmea adequado, conforme se indica na Fig. 3.



Fig. 3 - Cabo GPIB (esquerda) e ficha GPIB com as respectivas linhas (direita).

No que respeita à ligação física dos dispositivos, esta pode ser efectuada através de três formas diferentes; numa configuração linear, numa configuração em estrela, ou numa combinação das duas (ver Fig. 4).

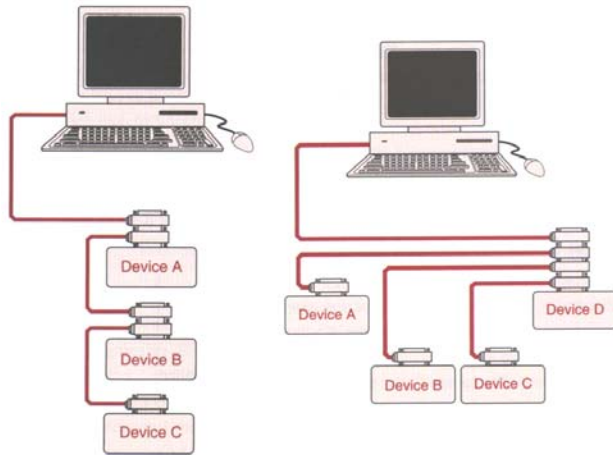


Fig. 4 - Configuração Linear (esquerda). Configuração Estrela (direita).

Em termos de ligações eléctricas o interface GPIB usa lógica negativa com os níveis *standard* TTL. Quando no *handshake* a linha DAV está no estado lógico 'V', existirá um nível lógico TTL *Low* ($\leq 0.8V$) e quando no estado lógico 'F' existirá um nível lógico TTL *High* ($\geq 2.0V$).

III. LABVIEW - PROGRAMAÇÃO GRÁFICA PARA INSTRUMENTAÇÃO

O LabVIEW (*Laboratory Virtual Instrument Engineering Workbench*), foi desenvolvido pela *National Instruments Corporation* [3]. Esta aplicação é uma ferramenta de programação para instrumentação e análise com enormes potencialidades, a qual pode correr sobre as mais populares plataformas (*Microsoft Windows, Sun, Apple Macintosh, Workstations HP-UX, etc.*). O LabVIEW, parte da natureza sequencial das linguagens de programação tradicionais e introduz um ambiente de programação gráfico, bem como todas as ferramentas necessárias para aquisição de dados, análise e apresentação. Com esta linguagem de programação gráfica, a qual foi designada por "G", a programação é desenvolvida dentro de um diagrama de blocos. Notação normalmente usada por cientistas e engenheiros. Criado o diagrama de blocos, o LabVIEW irá proceder à compilação para código máquina.

O LabVIEW consegue integrar a aquisição de dados, a análise e a apresentação num só sistema [4]. Para a aquisição de dados, este suporta diversas normas; RS-232/422, IEEE-488(GPIB), e VXI (*VME eXtensions for*

Instrumentation (barramento)), incluindo funções VISA (*Virtual Instrument Software Architecture*), bem como placas *plug-in* para aquisição de dados DAQ (*Data Acquisition*). Uma biblioteca de instrumentos com centenas de instrumentos simplifica as aplicações de controlo de instrumentos. Para a análise de dados, o LabVIEW inclui uma extensa biblioteca de análise, contendo funções para geração de sinais, processamento de sinal como filtros, janelas, estatística, regressão linear e aritmética matricial.

Dado que o LabVIEW é, por natureza, uma aplicação gráfica, está inerente a este, um pacote de apresentação de dados. O LabVIEW consegue gerar diagramas, gráficos, bem como gráficos definidos pelo utilizador.

Neste projecto utilizámos a versão 4.0.1 para PC, sobre o sistema operativo *Windows 95* da *Microsoft Corporation*.

O ambiente de programação desta aplicação é composto por dois planos: o diagrama de blocos e o painel [4] e [5]. O painel é a parte visível do programa para o utilizador. É no painel que estão todos os controladores, indicadores, botões, campos de preenchimento, etc. No diagrama está a programação em termos de algoritmo, que vai utilizar e controlar tudo o que foi colocado no painel, tratar das comunicações com os dispositivos, controlar os ciclos de execução do programa, etc.

O conjunto painel e diagrama de blocos formam aquilo a que a *National Instruments* chamou de *Virtual Instrument* (VI). Vários VIs podem ser agrupados para formar um ficheiro LLB que é aquilo a que normalmente se chama biblioteca de comandos. Os ficheiros LLB que acompanham o equipamento de laboratório, trazem um conjunto de VIs que desempenham funções básicas de comunicação com o dispositivo, sendo por isso comum designar tais LLBs como *drivers LabVIEW* do equipamento.

Tal como foi referido anteriormente, todos os programas em LabVIEW, ou VIs, são constituídos por um painel e um diagrama de blocos. As *palettes* existentes têm como função dar ao programador opções para criar ou alterar os seus VIs.

O painel frontal é o interface gráfico do VI do LabVIEW. Este interface reúne as entradas por parte do utilizador bem como apresenta as saídas geradas pelo programa. Este painel pode conter botões rotativos, botões de pressão, gráficos bem como outros controladores e indicadores.

Um exemplo do aspecto do painel frontal desta aplicação é apresentado na Fig. 5.

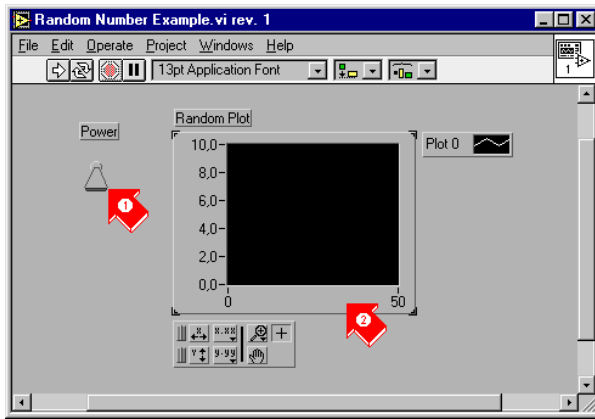


Fig. 5 - Painel frontal (1. Controlador – entrada ; 2. Indicador – saída).

No que respeita ao diagrama de blocos, este contém o código fonte do VI sob uma forma gráfica. É no diagrama de blocos que se procede à programação do VI por forma a controlar e executar funções sobre os controladores e indicadores criados no painel frontal.

Este diagrama pode incluir funções e estruturas da biblioteca de VIs do LabVIEW. Pode também conter terminais que estão associados com controladores e indicadores criados no painel frontal. Podemos observar o seu aspecto através da Fig. 6.

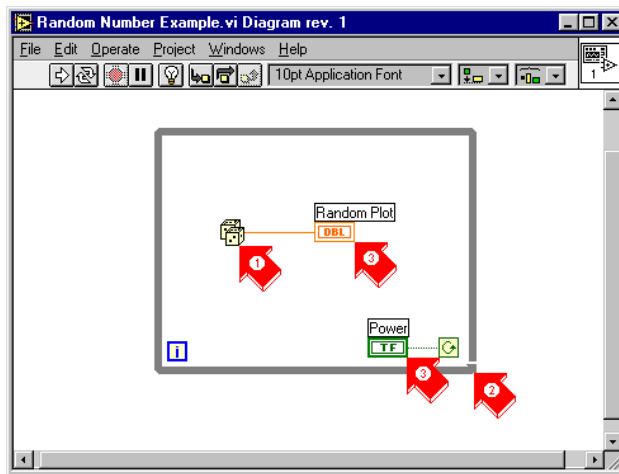


Fig. 6 - Diagrama de blocos (1. Função ; 2. Estrutura ; 3. Terminais do painel frontal).

As *palettes* do LabVIEW fornecem as opções necessárias para criar e editar o painel frontal e o diagrama de blocos.

A *palette* de ferramentas (Fig. 7) é usada quer no painel frontal quer no diagrama de blocos. Esta contém as ferramentas usadas para editar e fazer o *debugging* do painel frontal e do diagrama de blocos.

A *palette* de controlo (Fig. 7) é usada no painel frontal. Esta *palette* contém os controladores e os indicadores

usados no painel frontal para criar o interface gráfico com o utilizador.

A *palette* de funções (Fig. 7) é usada no diagrama de blocos. Esta *palette* contém todos os objectos usados para programar os VIs, tais como funções aritméticas, funções I/O de instrumentos, funções I/O de ficheiros, bem como operações de aquisição de dados.

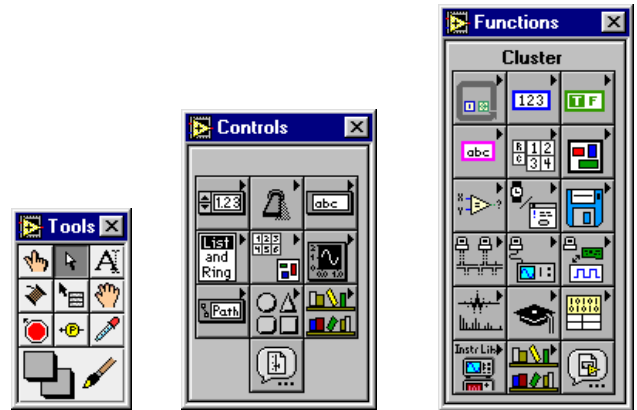


Fig. 7 - Palette de ferramentas ; Palette de controlo ; Palette de funções.

A chave para criar aplicações em LabVIEW reside no entendimento e no uso da hierarquia natural do VI. Isto é, após ter-se criado o VI, este pode ser usado como SubVI no diagrama de blocos do VI de alto nível. Se um VI possui um elevado número de *icons*, é possível agrupá-los num único VI de baixo nível por forma a manter a simplicidade do diagrama de blocos. Esta estrutura modular faz com que seja mais fácil o *debugging*, a compreensão e a manutenção das aplicações criadas.

Os SubVIs podem ser equiparados a funções ou subrotinas em linguagem de programação convencional.

O LabVIEW disponibiliza todas as funções usuais para programação, bem como bibliotecas de funções dedicadas a este tipo de aplicações.

IV. CRIAÇÃO DE DRIVERS COM O LABVIEW

Para uma correcta criação de *drivers* de instrumento através do LabVIEW, há que ter em conta diversos conceitos importantes [4] e [5]. O primeiro conceito será a definição de *driver* segundo os criadores do LabVIEW e o segundo conceito serão as formas de implementação possíveis para esses *drivers*.

Um *driver* consiste num bloco de *software*, o qual controla um instrumento em particular. O LabVIEW com o seu conceito de painéis é ideal para a criação destes *drivers*. O painel frontal pode simular a operação do painel frontal de um instrumento. O diagrama de blocos, por sua vez, envia os comandos necessários para o

instrumento por forma a este executar a operação especificada no painel frontal.

Após a construção do *driver* estar concluída, já não há necessidade do operador estar continuamente a recordar os comandos necessários para controlar o instrumento, sendo necessário apenas especificar a operação desejada (entrada do *driver*) no painel frontal.

Ter simplesmente um painel de *software* para controlar um instrumento, não dá grande valor a um *driver*. O real valor está no facto de se poder usar o *driver* desenvolvido como um SubVI em conjugação com outros SubVIs num VI de maior dimensão, por forma a controlar um sistema completo.

O LabVIEW possui uma biblioteca de *drivers* bastante grande, com *drivers* para instrumentos de diversas empresas que produzem instrumentos [6].

Dado que existem muitos tipos diferentes de instrumentos, é impossível demonstrar as técnicas para a criação de *drivers* para todos os tipos de instrumentos. Contudo, todos os *drivers* geram *strings* de comando que serão enviadas ao instrumento para efectuar as operações especificadas no painel frontal.

O projecto, desenvolvimento e uso de *drivers* de instrumentos são definidos através de dois modelos concepcionais. O primeiro modelo, **Modelo de Interface Externo**, mostra como o *driver* realiza o interface com outros componentes do sistema. O segundo modelo, **Modelo de Projecto Interno**, define a organização interna dos módulos de *software* do *driver*.

O Modelo de Interface Externo para os *drivers* do LabVIEW define a forma como o interface é realizado entre os *drivers* e os outros componentes do sistema numa determinada aplicação. Os componentes principais do modelo de interface externo inclui o corpo funcional, os interfaces de desenvolvimento programático e interactivo, o interface I/O e o interface de subrotinas, conforme se pode observar na Fig. 8.

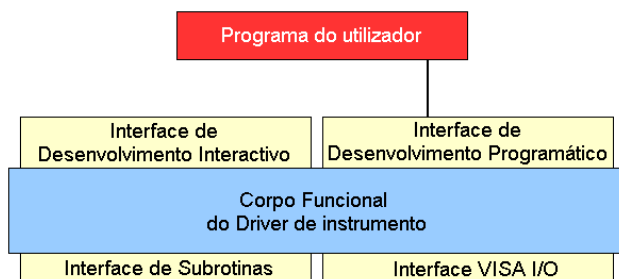


Fig. 8 - Diagrama do modelo de interface externo.

O corpo funcional representa o código do *driver*, propriamente dito. Os *drivers* de instrumentos do LabVIEW são escritos utilizando o ambiente standard de programação gráfica do LabVIEW. As vantagens para

este facto incluem *drivers* mais robustos, melhor funcionalidade e melhor controlo do programador sobre o desenvolvimento do *driver*, já que o código fonte pode ser facilmente visualizado por forma a optimizá-lo para cada aplicação desenvolvida.

Os interfaces de desenvolvimento programático e interactivo descrevem a representação gráfica do *driver*. O interface de desenvolvimento interactivo é o painel frontal do *driver* do LabVIEW. É um painel análogo ao painel físico do instrumento permitindo que seja operado interactivamente bem como verificar a comunicação. O interface de desenvolvimento programático do *driver* é o Icon/Conector [4] e [5]. Através do Icon/Conector, é possível chamar ou executar o VI do *driver* de outro VI bem como realizar de uma forma fácil sistemas de teste através da combinação de alguns *drivers* (VIs) desenvolvidos.

Os instrumentos que se pretendem controlar nas diversas aplicações podem incluir VXI, GPIB, RS-232, Ethernet e outros tipos de instrumentos. Por este motivo é de considerar durante o desenvolvimento de *drivers* o interface I/O. Os interfaces I/O para *drivers* de LabVIEW são as bibliotecas de VIs Série, VISA, GPIB e VXI.

A última parte do modelo de interface externo é o interface de subrotinas. Uma vez que os *drivers* dos instrumentos são realizados no código *standard* do LabVIEW, um *driver* de um qualquer instrumento é um programa de *software* com as mesmas capacidades como qualquer outro VI do LabVIEW.

Após termos visto como um *driver* de um instrumento faz o interface com o resto do sistema, vamos agora concentrar a atenção no projecto interno de um *driver*. O modelo de projecto interno de um *driver* de um instrumento para o LabVIEW é apresentado na Fig. 9. Este modelo define a organização do corpo funcional do *driver*. Este modelo é importante para quem pretende desenvolver um *driver* de um qualquer instrumento, uma vez que esta é a base em que assentam as linhas de desenvolvimento de *drivers* em LabVIEW.

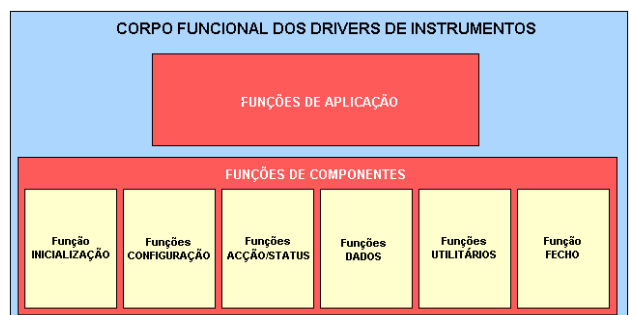


Fig. 9 - Diagrama do modelo de projecto interno.

Por outro lado este modelo é também importante para os utilizadores terminais já que todos os *drivers* de

instrumentos desenvolvidos no LabVIEW se encontram organizados segundo este modelo.

O corpo funcional de um *driver* LabVIEW consiste em duas categorias principais de VIs. A primeira categoria é a colecção de componentes de VIs, um conjunto de módulos individuais de *software* em que cada um controla uma área específica das funcionalidades do instrumento. A segunda categoria é a colecção de VIs de aplicação de alto nível, que mostra como combinar os componentes dos VIs por forma a executar testes básicos e operações de medidas com o instrumento.

Os VIs de aplicação representam o nível mais elevado da hierarquia de um *driver* de um instrumento. Estes VIs de alto nível têm o seu código fonte escrito no diagrama de blocos do LabVIEW e controlam as configurações e as medidas normalmente mais usadas em instrumentos.

Os *drivers* de instrumentos do LabVIEW possuem componentes de VIs, os quais representam um conjunto modular de VIs que contém todas as capacidades de configuração e medida de todos os instrumentos. Os componentes dos VIs enquadram-se em categorias como inicialização, configuração, acção/status, dados, utilitários e fecho. A combinação e/ou integração dos componentes dos VIs e dos VIs de aplicação, origina a construção do *driver* do instrumento.

V. MÓDULOS INDIVIDUAIS DO EQUIPAMENTO

A “construção” dos módulos individuais dos “instrumentos virtuais” através do LabVIEW serviu não só para criar um interface gráfico para uma possível utilização em modo *standalone*, mas essencialmente para criar o seu *driver* de controlo para interligação futura em bancadas de medição automática [6]. Dos nove instrumentos virtuais criados, cinco foram utilizados neste tipo de interligação. Para o controlo individual de cada dispositivo, fez-se uso de aparelhos equipados com uma porta de interface GPIB e do seu respectivo *driver* (conjunto de blocos básicos de comunicação) fornecido pelo fabricante ou pela *National Instruments*.

Antes de qualquer desenvolvimento de *software*, foi efectuado previamente um estudo detalhado do equipamento, não só tendo em conta os desejos de implementação para um dado aparelho, por parte dos utilizadores alvo destas aplicações, mas também da sua estrutura física [7] para uma melhor harmonia entre o equipamento e o *software*, donde resulta um melhor funcionamento das aplicações.

Neste projecto foram desenvolvidos interfaces gráficos individuais para nove aparelhos existentes nos laboratórios de Óptica e de Rádio Frequência, conforme referido na lista a baixo.

HP 8153A	– Lightwave Multimeter
HP 3478A	– Multimeter
HP 54120B	– Digitizing Oscilloscope Mainframe
HP 54610B	– Oscilloscope (500MHz)
HP 8593E	– Spectrum Analyser
HP 8563E	– Spectrum Analyser
Newport 5005	– Laser Diode Driver
Newport 3040	– Temperature Controller
Wiltron 67xxB	– Swept Frequency Synthesizer

Como exemplo do aspecto gráfico final destes módulos individuais, é apresentado na Fig. 10 o interface gráfico criado para o HP 54610B na versão de um canal (para este aparelho foram criadas duas versões; uma para utilização de um só canal e outra para utilização dos dois canais).

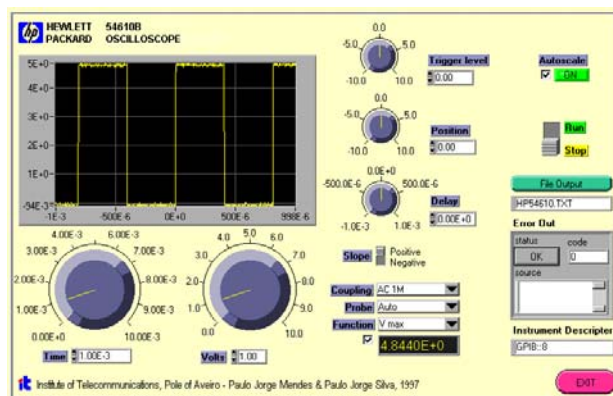


Fig. 10 - Painel frontal do osciloscópio (500MHz) com um canal.

As funcionalidades implementadas para este osciloscópio foram as indicadas abaixo;

- Visualização da forma de onda;
- Selecção da base de tempo;
- Selecção da escala de tensão;
- Ajuste do nível do *trigger*;
- Ajuste de posição vertical;
- Ajuste atraso;
- Selecção do *slope*;
- Selecção do *coupling*;
- Selecção do tipo de sonda;
- Auto escala;
- Selecção e visualização de parâmetros da forma de onda observada.

Foram também criadas novas funcionalidades através de *software* como sendo;

- Gravação da forma de onda em ficheiro;
- Controlo *Run/Stop*.

A funcionalidade de criação de um ficheiro ASCII para armazenamento dos valores obtidos é comum a todos os interfaces desenvolvidos.

VI. BANCADAS DE MEDIDA DESENVOLVIDAS

A. Bancada de caracterização Pin-Pout a 1 tom

Com os diferentes módulos (*drivers*) de cada um dos dispositivos a controlar e após uma selecção dos que nos interessariam para esta bancada (no caso o **Analizador de Espectros HP8563E** e o **Sintetizador de Varrimento W67XXB**) elaborou-se um estudo prévio dos SubVIs a serem utilizados no comando e controlo da bancada [6].

Para esta bancada foi utilizado um processo de medida algo diferente do realizado manualmente pelo investigador em laboratório. No método de medição manual, o investigador irá recolher a informação da potência de saída para as diferentes componentes de frequência para uma dada potência de entrada. Neste método efectua-se o varrimento de forma inversa, i.e., para uma dada frequência, varre-se as potências de entrada do valor mínimo até ao valor máximo e recolhendo-se as diversas potências de saída para uma matriz de duas dimensões. Após todos os valores recolhidos calculamos a transposta da referida matriz, o que do ponto de vista do LabVIEW não implica muito dispêndio de tempo mesmo para matrizes na ordem do milhar de pontos.

Esta solução apresenta diversas vantagens do ponto de vista da sua implementação;

- Diminuição do tempo da caracterização (uma vez que as ordens da frequência de saída do gerador de frequências são mais simples do que estar constantemente a alterar a Frequência Central, o *Span* e a posição do *marker* do analisador de espectros);
- Permite ao investigador ter uma ideia de como será a forma da característica para cada uma das frequências ao contrário de as visualizar apenas no final.

O aspecto do interface gráfico final para esta bancada é apresentado na Fig. 11.

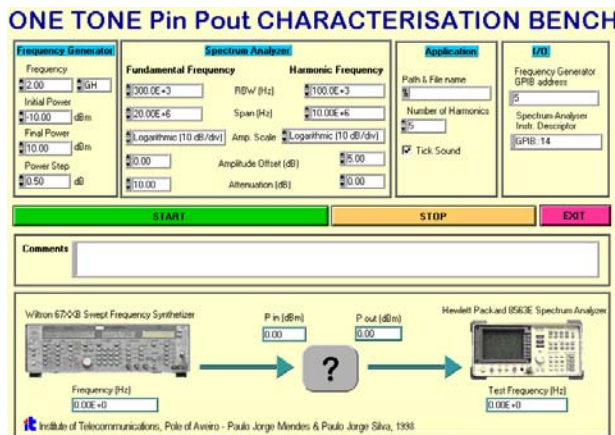


Fig. 11 - Painel da bancada de caracterização de dispositivos a 1 tom (introdução de valores).

A Fig. 11 representa a bancada de medida no início do seu funcionamento. Durante as caracterizações dos dispositivos em teste, a bancada surge com a visualização do gráfico em tempo real, bloqueando o acesso às funções de configuração da bancada (Fig. 12).

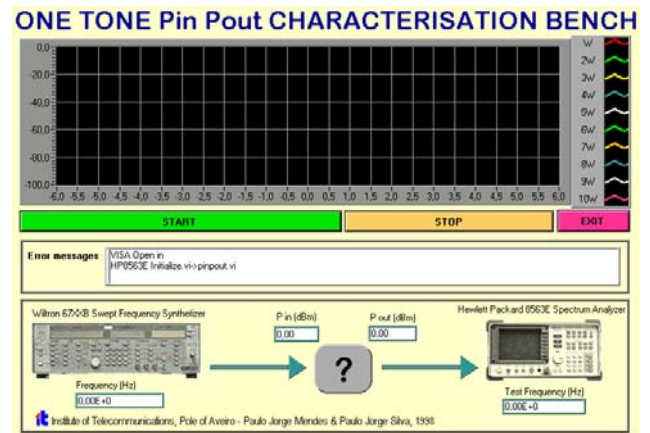


Fig. 12 - Painel da bancada de caracterização de dispositivos a 1 tom (modo de funcionamento).

O ensaio desta bancada de medição automática consistiu na utilização de um amplificador de potência a funcionar na banda dos 2GHz [9], conforme o esquema apresentado na Fig. 13.

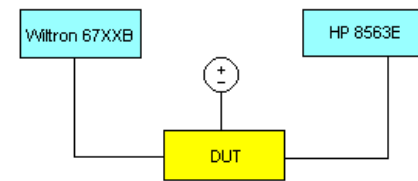


Fig. 13 - Configuração da montagem do amplificador para o ensaio da bancada.

Deste ensaio, resultou o gráfico apresentado na Fig. 14, o qual foi obtido a partir do ficheiro ASCII gerado pela bancada de medição (no caso, em formato TXT). Este ficheiro foi seguidamente importado para uma folha do Microsoft Excel, resultando no gráfico que se pode observar abaixo, onde estão representadas as curvas da característica Pin/Pout para a frequência fundamental e dos seus cinco primeiros harmónicos.

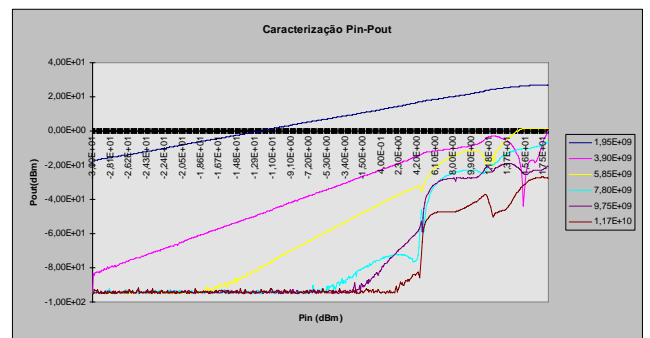


Fig. 14 - Resultado do ensaio da bancada de caracterização Pin-Pout.

B. Bancada de caracterização V-I de um Laser Semicondutor

Tal como para a bancada de caracterização anterior, também nesta procedeu-se previamente à escolha dos dispositivos a usar na bancada. Partindo dos dois dispositivos pretendidos (*Newport 5005 - Laser Diode Driver* e o *Newport 3040 - Temperature Controller*) iniciou-se o estudo dos SubVIs de cada um e o teste do seu funcionamento em simultâneo.

Os *drivers* destes dois aparelhos da *Newport* são bastante recentes, muito completos e com isso também muito complexos, facto este que revelou ser a maior dificuldade de implementação desta bancada.

Primeiramente foi necessário perceber o encadeamento lógico dos comandos dentro de cada aparelho. Esta foi a tarefa mais morosa dado que, para cada um, o exemplo de onde partimos possuía algumas largas dezenas de SubVIs dentro do seu VI.

Por forma a eliminar os SubVIs que não nos interessavam para a bancada, desenvolvemos um estudo dos aparelhos numa perspectiva sistémica. Partimos do todo pretendido e fomos subdividindo as várias etapas do problema até aos blocos base.

Um problema com que deparámos foi o conflito entre os endereços GPIB dos aparelhos. Apesar de eles estarem preparados para correr sobre o mesmo endereço, durante alguns ensaios da bancada, esta bloqueava devido a uma certa “confusão” entre os aparelhos, em que os comandos de um aparelho geravam mensagem de erro (desconhecimento do comando) no outro aparelho. Por forma a solucionar este problema, foram criadas sessões VISA distintas, bem como endereços GPIB distintos.

Atingida a versão final da bancada de caracterização corrente – tensão de um laser semicondutor, o seu aspecto gráfico á apresentado na Fig. 15 e Fig. 16.

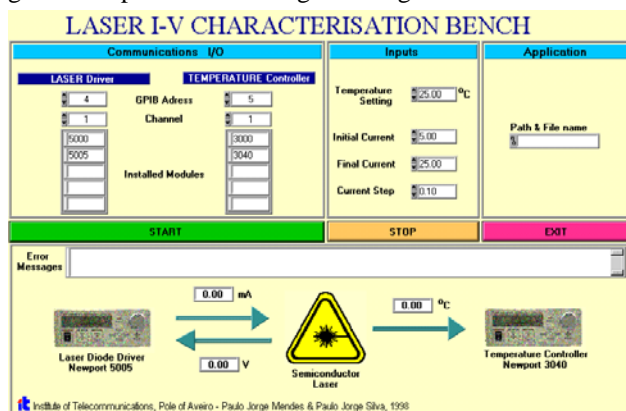


Fig. 15 - Painel da bancada de caracterização V-I de lasers semicondutores (introdução de valores).

Como se pode observar por estas duas figuras, também nesta bancada utilizou-se o gráfico de representação da característica em estudo. Neste caso a sua apresentação surge após finalizada a caracterização do laser em teste.

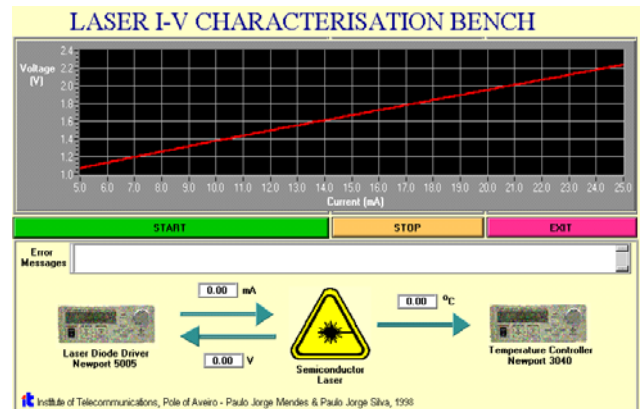


Fig. 16 - Painel da bancada de caracterização V-I de lasers semicondutores (apresentação final das medidas).

O ensaio final efectuado com esta bancada, baseou-se na caracterização de um laser semicondutor segundo a configuração da Fig. 17 [10].

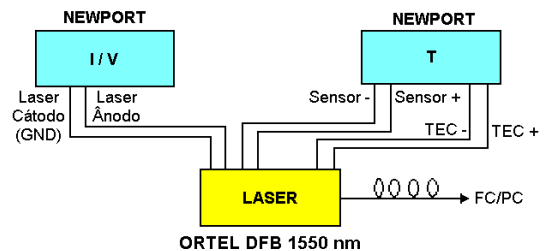


Fig. 17 - Configuração da montagem do laser para o ensaio da bancada.

A característica I-V obtida encontra-se representada na Fig. 18.

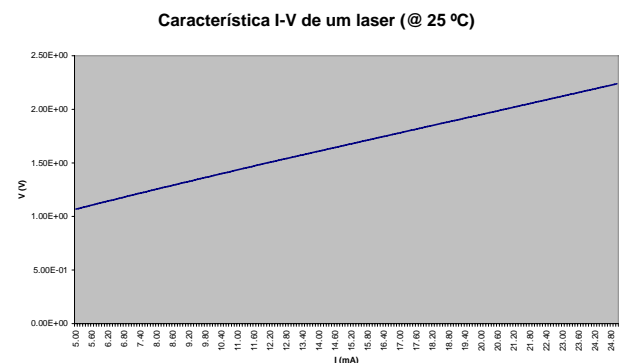


Fig. 18 - Característica I-V do laser ORTEL DFB 1550nm.

Na Fig. 19 é também apresentado o gráfico de variação da temperatura durante as medidas de modo a verificar-se a validade dos valores obtidos face à variação de temperatura permitida pelo utilizador. Note-se que neste caso, para uma estabilização da temperatura a 25°C, esta tem uma variação inferior a 0.05°C.

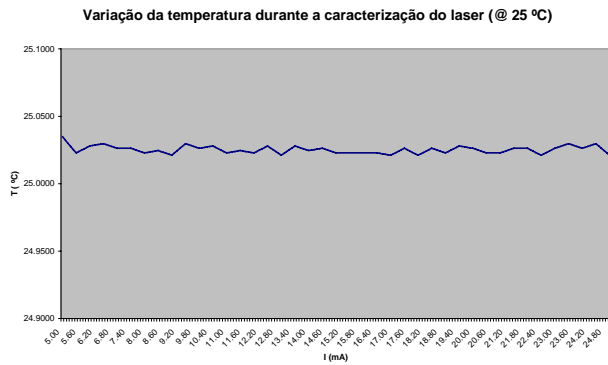


Fig. 19 - Variação da temperatura durante o ensaio.

C. Bancada de caracterização P-I de um Laser Semicondutor

Partindo da base da bancada da característica V-I, procedemos às alterações necessárias para introduzir mais um dispositivo (o *Lightwave Multimeter - HP8153A*) por forma a obtermos a leitura do valor da potência óptica do laser. A versão final desta bancada apresenta um aspecto final conforme mostrado na Fig. 20 a seguir.

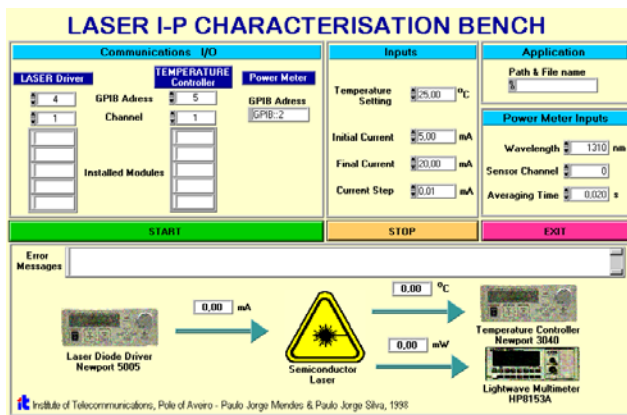


Fig. 20 - Painel da bancada de caracterização P-I de lasers semicondutores (introdução de valores).

Resultado do ensaio desta bancada, obteu-se a característica I-P apresentada na Fig. 21

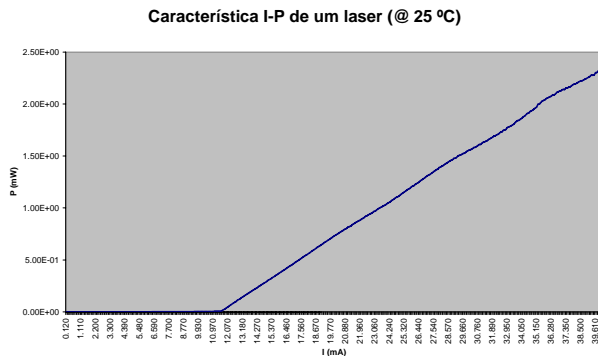


Fig. 21 – Característica I-P do laser ORTEL DFB 1550nm.

VII. HARDWARE E APOIO TÉCNICO

Dada a sua vocação de linguagem de programação visual é recomendado que o LabVIEW seja utilizado a correr numa plataforma cujas características típicas podem ser: Windows 95/98, com um processador Pentium, uma placa gráfica PCI, 32MB de memória RAM e um monitor capaz de suportar uma resolução de 1024x768 com uma boa taxa de refrescamento (>72Hz). Deve, no entanto, referir-se que para a operação de controlo de medida estas características ultrapassam largamente o mínimo necessário, dado que, usualmente, a velocidade dos instrumentos é muitíssimo menor que a velocidade de processamento de qualquer configuração mínima de um PC.

Na segunda metade deste projecto, a National Instruments lançou a versão 5.0 do LabVIEW. Esta versão apresenta já diversas melhorias face às críticas mais comuns dos programadores de LabVIEW, apresentando ainda novas ferramentas bastante interessantes, entre as quais destacamos as ferramentas de ligação à Internet (agora de mais fácil utilização), acesso a servidores de mail (por exemplo, para notificar, via correio electrónico, o final de uma dada medição automática, ou até mesmo o envio desta ao investigador) e ainda a opção de UNDO (inexistente em todas as versões anteriores).

É de salientar ainda o excelente apoio técnico fornecido pela *National Instruments* [8] através da sua página na Web.

VIII. CONCLUSÕES

Ao longo de todo este trabalho foi nosso objectivo, não só a realização de cada um dos aparelhos em causa, mas também o de os construir tão completos quanto possível. No entanto, dada a complexidade de alguns desses aparelhos optou-se por uma perspectiva de implementação das funcionalidades mais utilizadas quer para o laboratório de Rádio Frequência quer para o laboratório de Opto-Electrónica. Para que tal fosse possível foi necessário proceder a um inquérito junto dos futuros utilizadores das aplicações desenvolvidas, no sentido de reunir informação sobre os métodos de medida utilizados, as funções mais utilizadas em cada um dos aparelhos e o real interesse e importância na forma como interligar os mesmos.

Após o contacto inicial com a linguagem, bem como com todo o método da programação, comunicação e familiarização com os métodos de medida empregues nestes tipos de laboratórios, foi possível já reter algumas das potencialidades desta aplicação na automatização de métodos de medida em laboratórios. Ao longo deste tempo foi possível desenvolver um interface gráfico para nove aparelhos diferentes (para alguns dos quais foram realizadas duas versões, uma mais simples para aplicações mais básicas e outra bastante mais completa do ponto de

vista das funções existentes no aparelho em causa). Para além disso, foram ainda criadas três bancadas de medição automática: uma para caracterização Pin-Pout de dispositivos a 1 tom, uma para caracterização I-V de lasers semicondutores e outra de caracterização I-P de lasers semicondutores, facto que prova a possibilidade de interligação dos módulos separados que tinham sido criados anteriormente.

Para cada uma das bancadas, foram realizados ensaios práticos dos quais obtivemos uma óptima aceitação por parte dos futuros utilizadores destas aplicações, o que vem provar as expectativas iniciais quanto às potencialidades deste método de automatização de medidas em laboratórios.

Tendo como base os três exemplos apresentados anteriormente, conseguiu-se atingir níveis de rapidez de caracterização de dispositivos que permitiram aumentar em cerca de uma ordem de grandeza, o número de pontos no mesmo espaço de tempo, em relação a uma caracterização convencional.

Para isto contribui, em particular, o tempo dedicado à medida convencional em que, por exemplo, na bancada de caracterização Pin/Pout, uma tarde de trabalho de cerca de 6 horas permitiria obter cerca de 800 pontos. No exemplo apresentado, obtiveram-se 2886 pontos no mesmo tempo, sem necessidade do operador.

REFERÊNCIA

- [1] Eugene Fisher, C.W. Jensen, "Pet and the IEEE 488 bus (GPIB)", Osborne/McGraw-Hill, 1990
- [2] National Instruments, "Catálogo de produtos 1998"
- [3] Jack Quinn, "Digital Data Communications", Prentice Hall, 1995
- [4] Manuais do LabVIEW
- [5] National Instruments, "LabView Basics Course Manual", 1996
- [6] Paulo Silva, Paulo Mendes, "Automatização de Laboratórios de Medida de Componentes Ópticos e Electrónicos Através de LabVIEW" – relatório da disciplina de projecto de 5º Ano da Licenciatura em Engenharia de Electrónica e Telecomunicações, Universidade de Aveiro, Setembro, 1998
- [7] Manuais do equipamento de medida
- [8] <http://www.natinst.com/>
- [9] Nuno Borges de Carvalho and José Carlos Pedro, "Multi-tone Frequency domain simulation of nonlinear circuits in large and small signal regimes", Transaction on Microwave Theory and Techniques, Dezembro, 1998
- [10] J. L. Pinto , M. Pousa , P. S. André , A. Nolasco Pinto , M. R. Ferreira, Emissor para Sistemas de Transmissão Baseado na Tecnologia de SolitSigmas, Congresso 98 da Ordem dos Engenheiros - A Engenharia Portuguesa no Mundo, Lisboa, Portugal, Maio 1998.