

# Interface Humano para Sistemas de Aquisição de Dados e Processamento Digital de Sinal

Desenvolvimento de uma Plataforma de Teste e de um Osciloscópio Digital  
para um Sistema de Aquisição de Dados e Processamento Digital de Sinal  
Baseado no DSP TMS320C31

Relatório da Disciplina de Projecto



Autor  
Nuno Sérgio Cruz

Orientador  
Engenheiro  
José Basílio Simões

Regente da Disciplina  
Professor Doutor  
Carlos Correia

Dezembro de 2000  
Departamento de Física  
Faculdade de Ciências e Tecnologia  
Universidade de Coimbra

# Índice

Índice	2
Introdução	4
Motivações	4
Objectivos	5
Estrutura do Documento	5
História e Evolução da Instrumentação	7
No Início Era Assim...	7
A Era da Electrónica	7
Distribuição de Trabalho	8
Instrumentação Virtual	10
O Interface Gráfico	14
Introdução	14
Princípios de Projecto e Metodologia	15
Aplicação centrada no utilizador	15
Manipulação Directa	16
Consistência	16
Capacidade de Perdão	17
Retorno	18
Estética	18
Simplicidade	19
O Hardware	21
Introdução	21
Bloco de Comunicação com o PC	22
Bloco de Processamento	23
Bloco de Digitalização	23
Outros Blocos Presentes no Módulo	24
Plataforma de Teste	25
Introdução	25

Software do DSP	25
Software do PC	27
Osciloscópio Digital	32
Introdução	32
Software do DSP	33
Software Servidor	33
Software Cliente	36
Conclusões e Trabalho Futuro	40
Bibliografia	42

# Capítulo 1

## Introdução

### Motivações

Tem-se vindo a verificar, nos últimos anos, um crescente desenvolvimento e dedicar de atenção a sistemas de aquisição e tratamento digital de sinal, existindo diversos factores que têm proporcionado a evolução nesta área:

- O avanço das tecnologias microelectrónicas, que têm possibilitado o aumento das capacidades e velocidades dos DSP's, elemento central numa aplicação de tratamento digital de sinal, sendo ele que confere ao sistema o nível de inteligência, autonomia e versatilidade exigido.

- A crescente *performance* dos computadores pessoais, tal como a sua relação qualidade/preço e a confiança que eles garantem aos seus utilizadores.

- A existência de cada vez mais e melhores ferramentas de desenvolvimento de software, que permitem criar aplicações de alto nível com avançados interfaces gráficos, facilitando a comunicação do homem com a máquina.

- O desenvolvimento de novas tecnologias de comunicação que permitem o controlo remoto de instrumentos usando a internet como veículo de transmissão de dados.

O presente trabalho encontra nos pontos referidos as motivações para a sua realização, proporcionando um contacto prático e real com os temas mencionados.

## Objectivos

Cada vez mais uma aplicação é avaliada pela sua facilidade de utilização, não bastando que realize um conjunto de instruções com um bom desempenho para ser aclamada pelos seus utilizadores. É essencial que seja de fácil utilização, isto é, que tenha um interface humano de alto nível que permita uma aprendizagem rápida e eficaz de todas as suas potencialidades.

Foi tendo em conta estes factos que surgiu a necessidade de criar uma plataforma que permita o fácil desenvolvimento e teste de programas para um módulo de aquisição de dados baseado no DSP TMS320C31<sup>®</sup>.

Depois de desenvolvida a plataforma de teste, os conhecimentos de interfaces gráficos e do módulo de aquisição de dados e processamento digital de sinal foram aprofundados, com o desenvolvimento de um osciloscópio e analisador multicanal digital com capacidade de controlo e visualização de dados remoto usando a internet como meio de transporte de dados.

## Estrutura do Documento

Após uma breve introdução ao trabalho apresentado neste documento (Capítulo 1) farei uma apresentação histórica da evolução da instrumentação (Capítulo 2), com destaque para alguns desenvolvimentos recentes nesta área de investigação.

No Capítulo 3 é apresentada a importância do interface humano e a metodologia para um projecto eficaz, colocando em evidência os aspectos relevantes dessa metodologia.

No Capítulo 4 faz-se uma breve referência ao hardware que compõe o sistema e serviu de base ao software desenvolvido.

No Capítulo 5 é apresentado o software da plataforma de teste para o módulo de processamento digital de sinal baseado no DSP TMS320C31<sup>®</sup>.

No Capítulo 6 é apresentado o software de aplicação desenvolvido para a plataforma Windows<sup>®</sup> constituído por um osciloscópio digital e um analisador multicanal de processamento digital com capacidade de controlo e visualização de dados remota.

Este documento acaba com uma análise ao trabalho realizado e à possibilidade de desenvolvimentos futuros do sistema apresentado (Capítulo 7), seguido da bibliografia mais relevante usada durante o projecto.

# Capítulo 2

## História e Evolução da Instrumentação

[Goldberg]

### No Início Era Assim...

Há alguns séculos atrás não conhecíamos nada do funcionamento da electricidade. Após o trabalho de Ohm, Oersted, Ampere, Watt e alguns outros, definiram-se parâmetros como a tensão, corrente, potência, nascendo o campo de estudos da electricidade.

Em seguida desenvolveu-se a tecnologia eléctrica. Ela permitiu a distribuição da electricidade, o movimento de máquinas, o aquecimento e iluminação de casas, o transporte de bens e produtos. Tornou-se cada vez mais importante medir os parâmetros eléctricos dos aparelhos. Foram, por isso, desenvolvidos instrumentos de medida que se tornaram cada vez mais precisos, acompanhando a evolução da tecnologia. Durante grande parte do século XX as medidas centravam-se nos parâmetros eléctricos tensão, corrente, potência, frequência, etc.

### A Era da Electrónica

A era da electrónica começou com os tubos de vácuo, rádio e televisão. Durante a II Guerra Mundial muita electrónica foi desenvolvida para fins militares, modificando a forma de navegação, comunicação e o controlo de instrumentos, que passou de mecânico e visual para eléctrico e electrónico. Foram desenvolvidos instrumentos para medir os parâmetros eléctricos envolvidos.

Com o desenvolvimento dos transístores, deixámos de estar tão dependentes do tamanho e da dissipação de potência. Os instrumentos de medida tornaram-se menores em tamanho e em consumo, permitindo o aparecimento de instrumentos portáteis a funcionar com baterias.

Estes produtos eram “puros instrumentos de medida”. Eram compostos por fontes de alimentação, sensores, descodificadores e mostradores. Na maior parte dos casos, eram feitas ligações manuais, estabelecia-se os limites da leitura e copiava-se, fisicamente, os valores do mostrador para um bloco de notas. A posterior utilização dos dados não fazia parte do pacote do instrumento.

Na década de 50, o controlo industrial desejava mais que a simples medida dos parâmetros físicos. Como resultado desta necessidade, apareceram os primeiros sistemas de controlo muito rudimentares. Relays foram colocados nos instrumentos de medida para que os processos pudessem ser accionados automaticamente com a variação dos parâmetros. Rapidamente se evoluiu para sistemas com vários relays, permitindo o controlo de vários pontos do processo em simultâneo. Em seguida usaram-se detectores com taxas de aquisição diferentes e integradores para criar o sistema de controlo PID.

## Distribuição de Trabalho

Havia desenvolvimentos simultâneos em outros campos, cujo maior destaque vai para o campo da computação. A capacidade aumentava enquanto o tamanho diminuía. Apesar disso, os dois campos, instrumentação e computação, mantinham-se separados. Depois do aparecimentos dos microprocessadores, o tamanho, custo e potência dissipada caíram, permitindo a sua integração na instrumentação.

Os computadores ainda eram lentos, de capacidade limitada e necessitavam de programação específica para executar as suas tarefas. O armazenamento estava limitado a fita magnética ou grandes discos e

cilindros. Os computadores eram essencialmente para uso off-line. Eram utilizados para pós-processamento, depois de os dados terem sido gravados pelos instrumentos de medida em discos ou fitas magnéticas.

Isto não significa que a computação não fizesse parte dos instrumentos. Tornou-se um hábito incluir capacidades computacionais num instrumento, no entanto, eram aparelhos com aplicações específicas, desenvolvidas a pensar nas suas operações específicas. Estes módulos computacionais rudimentares, com o advento da microelectrónica permitiram um novo nível de utilidade para a instrumentação.

Ainda era impossível na década de 1980 e no início da década de 1990 usar computadores comerciais para aplicações de tempo real, essencialmente devido à sua baixa performance, e tempos de processamento demasiado elevados para a maioria dos casos práticos. Apesar disso, a necessidade de processamento adicional tornou-se cada vez mais óbvia, à medida que os instrumentos se tornavam cada vez mais complexos e os engenheiros cada vez mais impacientes com equipamentos de medida que se limitavam a apresentar valores num écran.

Tornou-se habitual um instrumento de medida aceitar um sinal, condicionando a entrada, linearizando-a, formatando-a, limitando-lhe a banda passante, etc. Em seguida era digitalizada, permitindo a manipulação dos dados em placas de processamento desenvolvidas para aplicações específicas de controlo ou de decisão analítica. Os instrumentos de medida podiam gerar sinais que transmitidos ao sistema em desenvolvimento permitiam testar o seu funcionamento. A necessidade de especificidade e velocidade deste tipo de instrumentação exigiam a utilização de equipamento desenvolvido para cada aplicação.

Os sistemas de controlo e de medida expandiram-se de simples equipamentos de medida de parâmetros eléctricos e electrónicos para outros campos de estudo como a física, mecânica, química, engenharia civil, medicina, etc. Um ramo significativo da medida e controlo dedicou-se à

indústria e a aplicações de grande velocidade, em que o tempo morto era um problema primordial.

A velocidade e capacidade dos computadores de aplicação genérica cresceu exponencialmente, permitindo a sua adaptação a aplicações que requerem medida e controlo em tempo real. Os computadores de âmbito comercial tornaram-se parte da instrumentação.

## Instrumentação Virtual

Um grande desenvolvimento verificou-se no interface gráfico, com o aparecimento do software tipo Windows. A simplicidade da operação juntou-se à crescente capacidade de processamento para estabelecer a união entre os computadores e a instrumentação. Tornou-se habitual os instrumentos serem embutidos em computadores de aplicação geral, permitindo medidas diversas e manipulações complexas, juntamente com grandes capacidades de armazenamento em memória ou disco, monitorização inteligente, apresentação gráfica de fácil compreensão e controlo dos processos envolvidos. Existem instrumentos completos, capazes de realizar todas as medidas e cálculos necessários, compostos por partes comerciais de computadores.

Com a integração de computadores e capacidade de computação nos instrumentos de medida o seu preço aumentou. Para fazer face a este aumento de preço surgiu a instrumentação virtual. Os computadores de grande capacidade e velocidade já não estavam disponíveis apenas por encomenda para integração em instrumentos caros. Os computadores de utilização geral incorporaram grande parte do hardware e do software necessários pelos instrumentos, para a sua aplicação específica. Eles tornaram-se também suficientemente rápidos para trabalhar em tempo real. Com o software especializado e algum hardware adicional, um computador de bancada num laboratório pode executar tarefas que há algum tempo só

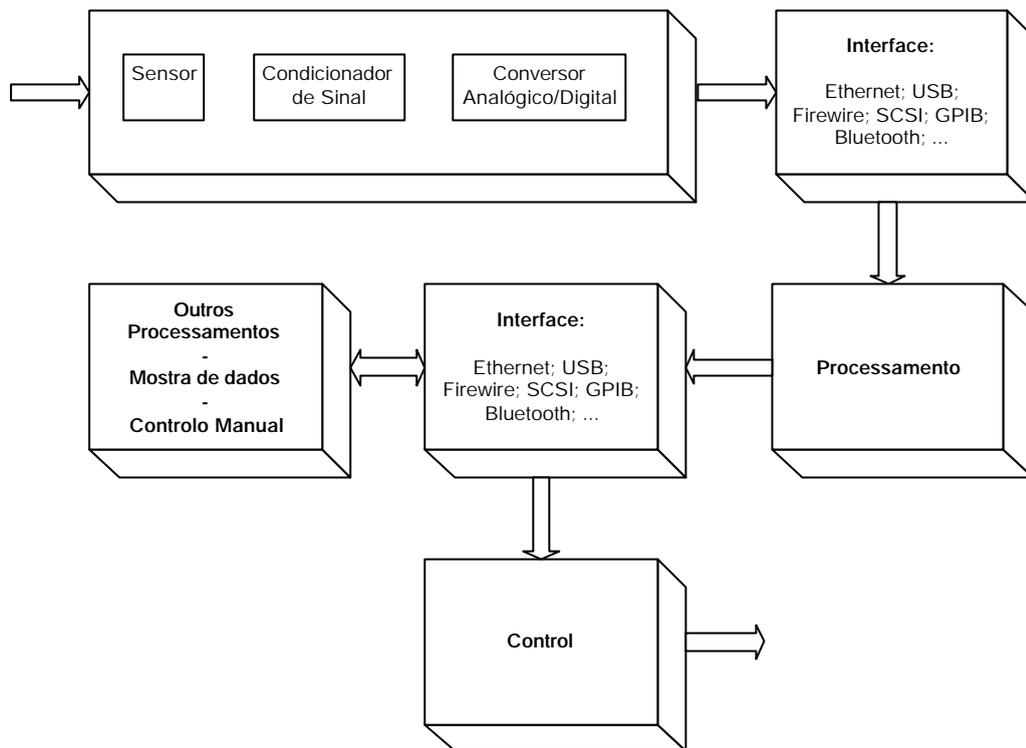
podiam ser realizadas por instrumentação específica de elevada performance e preço.

Sendo assim, para quê fornecer o poder de computação junto com o instrumento se o computador de bancada do utilizador podia realizar esse trabalho? Podemos diminuir os custos se fornecermos apenas o hardware específico que o computador não tem e o software para o utilizar. Eis o instrumento virtual.

O instrumento virtual é composto por algumas unidades especializadas integradas num computador de utilização geral, com o software e o conhecimento que os coloca a funcionar. O instrumento já não está dentro de uma enorme caixa num laboratório longe do gabinete, podendo mesmo estar na secretária, no computador em que temos o nosso processador de texto.

A complexidade do instrumento virtual pode ser muito variada, mas existem algumas componentes essenciais. Consideremos um sistema virtual de medida e controlo. Para medir tem de existir um sensor. Caso o parâmetro que queremos medir não seja eléctrico temos de incluir nele um transdutor para transformar a informação num sinal eléctrico. Tem de ter um circuito de condicionamento de sinal, que pode incluir amplificadores, filtros e rectificadores. Finalmente temos um conversor de analógico para digital.

Depois de estar no formato digital a informação pode ser processada, misturada, comparada, manipulada e armazenada conforme as necessidades da aplicação ou as especificações do instrumento. Em seguida pode ser apresentada no écran no formato gráfico desejado. Os dados podem ser colocados de novo no formato analógico para controlar processos. A figura 1 mostra um diagrama de blocos de um sistema típico.



**Figura 1 - Esquema típico de um Instrumento de Medida e Control**

Note-se que todas as operações de processamento podem ser encontradas num computador pessoal usual e que a velocidade deste é compatível com as necessidades da maioria dos instrumentos.

Os recentes desenvolvimentos da tecnologia das redes locais (LAN - Local Area Network) permite-nos separar fisicamente o computador da restante instrumentação. Muitas tecnologias podem ser usadas como o RS232, GPIB, USB, Ethernet, SCSI, dependendo das taxas de transferência de dados necessárias e das distâncias entre os componentes, não esquecendo o factor económico.

Assim sendo, onde está o instrumento? As suas diversas partes podem estar espalhadas por diversas partes do local de trabalho, ou até do mundo, se usarmos a Internet para comunicar entre os vários módulos. Com a introdução de novas tecnologias, como o Bluetooth, os módulos nem sequer necessitam de estar fisicamente interligados. Usando ligações de alta

velocidade na Internet os nossos dados podem ser adquiridos num local, processados em diversos sistemas computacionais espalhados pelo mundo, colocados numa base de dados comum a vários utilizadores, e apresentados nas mais diversas formas gráficas, consoante a escolha do engenheiro, técnico ou cientista que deseja consultar a informação. Isto é o que chamamos de Instrumentação Virtual. Virtual porque é algo que existe em função, mas não na sua forma habitual.

Este tipo de instrumentação exige a transferência rápida de dados entre dispositivos, software de processamento e de aplicação com interface gráfico avançado, sensores e controladores de elevada precisão. Com a velocidade dos avanços tecnológicos, cada vez mais equipamentos e instrumentação vem incluída com o PC padrão. Na vertente do software, as tecnologias vão-se tornando cada vez mais normalizadas, permitindo um desenvolvimento mais rápido e eficaz dos produtos.

O campo de estudos da instrumentação virtual está a dar os primeiros passos. Nos próximos anos serão desenvolvidas diversas sub-unidades projectadas para serem integradas em instrumentos virtuais. Serão os blocos para a construção da nova geração de instrumentação e medida.

# Capítulo 3

## O Interface Gráfico

[Microsoft]

### Introdução

O projecto do interface do software afecta, mais que qualquer outra coisa, o modo como o utilizador irá usufruir de um produto. Sendo assim, é importante promover um bom desenho do interface com o utilizador, com consistência visual e funcional.

As melhorias no interface gráfico e do utilizador na plataforma Windows<sup>®</sup> proporcionam uma evolução de um interface centrado na aplicação para um centrado na informação. Desta forma, é necessário repensar o interface do software, dos componentes básicos, das suas operações e das propriedades que se lhes aplicam, por forma a que os utilizadores possam interagir com a informação sem terem de pensar na aplicação, permitindo-lhes uma maior concentração nas suas tarefas.

Realizar um bom projecto para o software gráfico de uma aplicação permite ao utilizador transferir as suas capacidades e atenção de uma tarefa para a seguinte e aprender novas tarefas facilmente. A evolução para um projecto centrado na informação quebra a linha da aplicação tradicional, tornando qualquer inconsistência no interface mais óbvia e com mais elementos capazes de distrair o utilizador.

Por outro lado, um bom projecto de interface gráfico não garante uma boa aplicação. Existem outros factores a ter em consideração como o projecto da aplicação, análise de tarefas, criação de protótipos e a avaliação da sua facilidade de utilização.

## Princípios de Projecto e Metodologia

Um interface ao utilizador bem projectado é baseado em princípios e num processo de desenvolvimento centrado no utilizador e nas suas tarefas. Este capítulo faz um sumário dos princípios básicos do projecto de interface. Inclui ainda a técnica e metodologia a empregar na projecção de um interface real homem/computador.

### *Aplicação centrada no utilizador*

Um principio importante do projecto de interface ao utilizador é o facto de que este se deve sentir sempre com controlo sobre o software, em vez de controlado por ele. Este principio tem um conjunto de implicações.

A primeira implicação é o facto de ser o utilizador a iniciar uma acção e não o computador ou o software. O utilizador deve ter um papel activo e não reactivo. É vantajoso usar técnicas de automação de tarefas, mas devem ser implementadas de forma a permitir ao utilizador escolher e controlar a automação.

A segunda implicação diz respeito aos utilizadores, devido à sua diversidade de capacidades e preferências, terem a possibilidade de personalizar alguns aspectos do interface. O software do sistema proporciona ao utilizador o acesso a alguns destes aspectos. O software da aplicação deve reflectir a configuração do sistema, tal como cor, fontes e outras opções.

A implicação final prende-se com o software ser tão interactivo e com capacidade de resposta quanto possível. Deve-se evitar módulos de processamento extensos sempre que possível. Um módulo de processamento é um estado que exclui a interacção com o utilizador. Quando um módulo de processamento é a única ou a melhor alternativa de projecto, deve-se fazer com que a evolução do processo seja explicitamente visível e resulte de uma escolha explícita do utilizador. Estes módulos devem também ser facilmente cancelados pelo utilizador.

## *Manipulação Directa*

O software deve ser projectado de forma a que os utilizadores possam manipular directamente as representações da informação. Quer arrastando um objecto para o realocar, quer navegando para outro local do documento, o utilizador deve poder verificar como a sua acção afectou os objectos no écran. A visibilidade da informação e das possíveis escolhas também reduz o esforço mental do utilizador. Os utilizadores podem reconhecer um comando mais facilmente do que recordar a sua sintaxe.

Metáforas familiares permitem um interface directo e intuitivo às tarefas do utilizador. Permitindo aos utilizadores transferir o seu conhecimento e experiências, as metáforas simplificam o processo de aprendizagem do comportamento do software baseado em representações. Quando se usam metáforas, não é necessário limitar a implementação computacional ao "mundo real". O objectivo da utilização da metáfora no interface é proporcionar uma ponte cognitiva: a metáfora não é um fim em si mesmo.

As metáforas facilitam o reconhecimento dos comandos. Um utilizador lembra-se do significado associado a um objecto familiar mais facilmente que se lembra do nome de um determinado comando.

## *Consistência*

A consistência permite ao utilizador transferir o conhecimento já adquirido para novas tarefas, aprender novos conceitos com maior facilidade e focar a sua atenção nas tarefas que deseja realizar e não nos comandos que deve aprender para as realizar. Ao proporcionar estabilidade e consistência, o software torna o interface mais familiar e previsível.

A consistência é importante em todos os aspectos do interface, incluindo o nome dos comandos, representação visual da informação e comportamento operacional.

Para dar consistência ao software deve-se ter em conta os seguintes aspectos:

- Consistência no produto - Apresentar funções comuns usando um conjunto de comandos e interfaces semelhantes. Por exemplo, evitar que um determinado comando apresente comportamentos diferentes em situações diferentes, tornando-o imprevisível para o utilizador. Uma regra que se deve sempre seguir é usar o mesmo comando para realizar funções que parecem semelhantes ao utilizador.
- Consistência com o sistema operativo - Ao manter a consistência com as regras e convenções do sistema operativo sobre o qual a aplicação está a correr, o software beneficia da capacidade do utilizador de aplicar conhecimentos e capacidade de interacção já apreendidos.
- Consistência das metáforas - Se um determinado comportamento é característico de um objecto, então o utilizador vai ter alguma dificuldade em associar esse objecto a outro comportamento. Por exemplo uma imagem de uma incineradora transmite metaforicamente o sentido de destruição total e irrecuperabilidade, enquanto o cesto de lixo ainda permite a recuperação dos objectos que tem dentro.

### *Capacidade de Perdão*

Os utilizadores gostam de explorar o interface e aprendem muitas vezes por um processo de tentativa erro. Um interface de qualidade permite uma descoberta interactiva. Para tornar isso possível, o interface só deve apresentar as escolhas possíveis, retirando a permissão aquelas que não estão disponíveis; avisar o utilizador de escolhas potencialmente perigosas para a integridade do sistema e dos dados; e finalmente, permitir a

reversibilidade e recuperação de todas as acções de um utilizador mais desatento.

Mesmo nos interfaces melhor projectados, os utilizadores cometem erros. Estes erros podem ser físicos - apontar acidentalmente para os dados ou comando errados, e mentais - tomar uma decisão errada devido ao desconhecimento do comportamento de um comando ou conjunto de dados. Um projecto correcto evita situações que podem resultar em erro e permite recuperar facilmente de situações de erro criadas pelo utilizador.

### *Retorno*

Deve-se sempre fornecer retorno às acções do utilizador. O retorno pode ser visual ou sonoro e serve para confirmar ao utilizador que o software está a responder à ordem, comunicando sempre que possível os detalhes que distinguem a natureza da acção.

Para que o retorno seja eficaz é necessário que seja periódico no tempo. Mesmo quando o processamento de uma determinada tarefa é demorado, o retorno deve ser realizado de forma a dar informação ao utilizador sobre o estado do processo e se possível a forma de o cancelar. Nada é mais "desconcertante" para o utilizador que um écran que não responde a comandos, nem fornece informação sobre o que está a fazer. Um utilizador típico apenas tolera durante alguns segundos um interface estático.

Outro aspecto a ter em conta é que o tipo de resposta fornecida seja apropriada para o tipo de tarefa que está a ser executada. A mudança de local de um apontador ou uma barra de estado comunicam informação simples de uma forma eficaz. Retorno mais complexo pode exigir outras formas de comunicação como uma caixa de diálogo.

### *Estética*

O design gráfico é uma parte importante do interface do software. As características visuais fornecem uma impressão valiosa e comunicam

importantes pistas sobre o comportamento de interacção de um objecto particular.

Outro aspecto a ter em conta é que todos os objectos no écran competem pela atenção do utilizador. Deve-se fornecer um ambiente agradável, que contribua para que o utilizador compreenda toda a informação que lhe é fornecida, tendo em atenção que o excesso de informação pode deixar o utilizador confuso.

Um designer gráfico pode ser muito útil para ajudar a ponderar o aspecto da aplicação em desenvolvimento.

### *Simplicidade*

Um interface deve ser simples, fácil de aprender e fácil de usar. Deve permitir o acesso a toda a funcionalidade proporcionada pela aplicação, maximizando a funcionalidade e mantendo a simplicidade do interface. Um projecto eficaz faz o balanço equilibrado entre estes objectivos.

Uma forma de manter a simplicidade é reduzir a informação apresentada ao mínimo necessário para comunicar adequadamente. Por exemplo, deve-se evitar o uso de descrições exaustivas em nomes de comandos ou mensagens. Frases irrelevantes destroem a eficácia na utilização do software porque impedem o utilizador de extrair a informação essencial com facilidade.

Outra forma de desenhar um interface simples e útil é usar semântica e colocação dos elementos de uma forma natural. O arranjo e apresentação dos elementos afectam o seu significado e associação.

É possível ajudar os utilizadores a gerir a complexidade duma aplicação fornecendo a informação de uma forma progressiva. A informação progressiva exige uma organização cuidada, de forma a ser mostrada apenas na altura correcta. Ao "esconder" informação que ainda não é necessária ao utilizador, diminuimos a quantidade de dados a processar. Por exemplo, ao

seleccionar um menu aparecem as suas escolhas, mas o uso de uma caixa de diálogo pode diminuir o número de opções do menu.

Por outro lado, a informação progressiva não deve usada com técnicas não convencionais para revelar a informação, como exigir um modificador para aceder a uma função básica ou forçar o uso de uma longa sequência de menus ao longo de uma interacção hierárquica desnecessária. Desta forma podemos tornar o interface mais complexo e confuso.

# Capítulo 4

## O Hardware

### Introdução

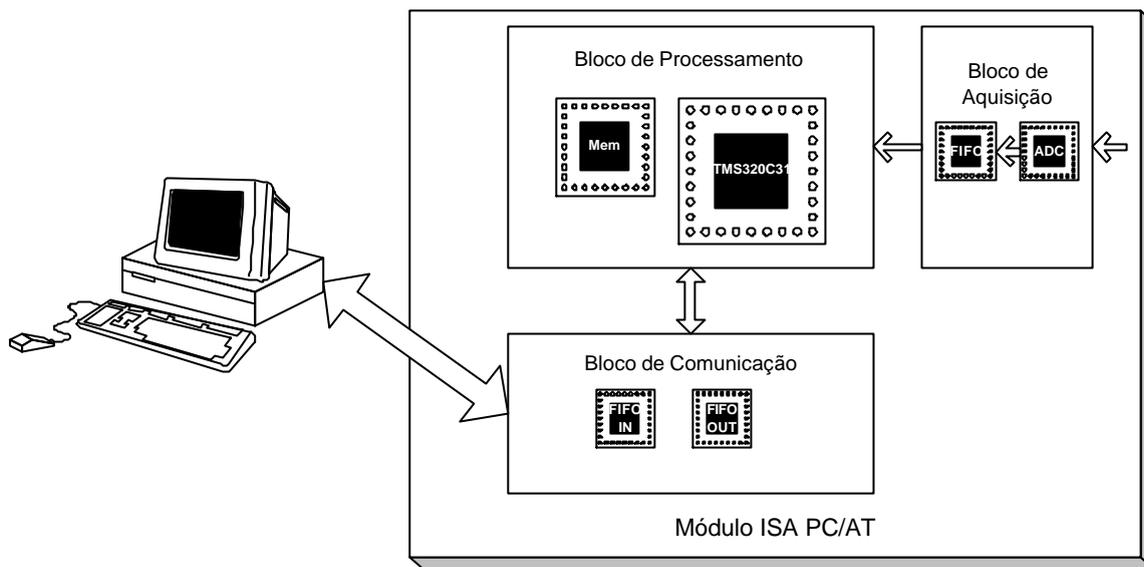
Neste capítulo está descrito de forma breve o hardware do sistema de aquisição de dados e de processamento digital de sinal que serviu de base para o desenvolvimento do software do projecto apresentado neste documento.

O sistema é composto por um PC a correr o sistema operativo Windows™ e um módulo PC-AT (8 bits), desenvolvido pelo Grupo de Electrónica e Instrumentação para aplicações genéricas na área de Espectroscopia Nuclear.

Este módulo é constituído por três blocos principais:

- Bloco de comunicação com o PC
- Bloco de processamento
- Bloco de digitalização e aquisição

Apresenta-se na figura um esquema do hardware do sistema.



**Figura 2 - Esquema do Hardware do Sistema**

## Bloco de Comunicação com o PC

A comunicação entre o módulo e o PC é realizada através de duas memórias sequenciais do tipo FIFO, correspondendo cada uma a um dos sentidos da comunicação. A transferência de dados de e para o PC é feita através de escritas e de leituras nas respectivas memórias FIFO. Adicionalmente, tanto o PC como o DSP têm acesso à sinalização de conteúdo das memórias FIFO, sabendo em cada momento se essas possuem dados, se encontram cheias pela metade ou se estão completamente cheias.

Além dos dados que lhe chegam através do FIFO, o PC tem ainda acesso, por intermédio de um porto de leitura e outro de escrita, a informação variada sobre o estado do módulo. Além da sinalização de conteúdo das memórias FIFO, já mencionada, o PC tem acesso a duas *flags* de sinalização do DSP, muito úteis durante a fase de desenvolvimento e diagnóstico.

O PC dispõe também de um porto de escrita, através do qual pode controlar alguns aspectos do funcionamento da placa. Assim, através desse

porto pode provocar um cancelamento e inicialização das memórias FIFO, do próprio DSP e das linhas de interrupção deste último. Tem ainda capacidade de provocar uma interrupção no DSP sempre que seja forçoso chamar a sua atenção (na actual versão do software, essa capacidade só é utilizada no carregamento inicial do programa do DSP) e de colocar o DSP em estado de espera.

## Bloco de Processamento

O bloco de processamento é baseado no processador digital de sinal de vírgula flutuante TMS320C31, disponibilizado pela Texas Instruments. Este processador de 32 bits combina facilidade de utilização com elevadas performances sendo capaz de realizar 40 MFLOPS e tendo um ciclo de procura (*fetch*) e execução, de uma instrução simples, de apenas 50 ns. As suas elevadas performances são ainda potenciadas pela capacidade de execução de instruções em paralelo ou pelo *pipelining* interno. Contém ainda canais de DMA, *timers* internos configuráveis e portos de comunicações (série). Estas características tornam-no ideal para este tipo de aplicações onde a velocidade de transferência e processamento de dados, é um factor preponderante.

Para além do DSP este bloco é ainda constituído por memória externa que permite ao processador armazenar os dados antes e após serem processados.

## Bloco de Digitalização

O bloco de digitalização de sinal gira em torno de um Conversor Analógico-Digital (ADC) tipo *flash* de 8 bits. As frequências de aquisição possíveis deste ADC vão até aos 50 MHz. É ainda possível uma enorme gama de outras frequências de aquisição, seleccionáveis por software usando os *timers* do DSP.

Entre o ADC e o DSP foi necessário intercalar uma memória tampão, constituída por duas memórias FIFO de  $8\text{ k} \times 8\text{ bits}$  cada uma, alternando uma com a outra devido ao facto de, individualmente, não poderem suportar a frequência máxima do ADC..

No momento em que essas memórias ficam cheias ou meio cheias pode ser accionada uma interrupção do DSP. A rotina de serviço a essa interrupção procede à leitura do conteúdo das memórias FIFO, copiando os dados relevantes para uma zona de memória interna da placa. Os dados ficam assim disponíveis para o DSP, a fim de lhes aplicar um tratamento numérico ou proceder ao seu envio para o PC, quando para tal solicitado.

## **Outros Blocos Presentes no Módulo**

Este módulo tem também outros blocos de instrumentação relevantes como os blocos analógicos de Analisador Multicanal (MCA) e Contador Multicanal (MCS). [Malaquias] Estes blocos por não terem sido usados no âmbito deste projecto não serão descritos neste documento.

# Capítulo 5

## Plataforma de Teste

### Introdução

Neste capítulo é apresentado o software que tem por objectivo permitir o teste de aplicações desenvolvidas para o módulo de processamento digital de sinal.

No desenvolvimento desta plataforma deu-se especial atenção ao processo de comunicação entre o PC e DSP, na medida em que só através de uma comunicação eficaz seria possível ler os dados e variáveis do DSP sem interferir na aplicação que estava a correr e a ser desenvolvida.

Outro factor importante foi a visualização dos dados provenientes do DSP. A plataforma de teste permite visualizar graficamente zonas de memória do DSP e visualizar o valor de algumas variáveis pré-definidas pelo utilizador. Esses dados podem ser guardados num ficheiro para posterior análise, usando de novo o software aqui apresentado ou outras ferramentas de análise dados.

Apesar do software mais visível desta aplicação ser aquele que corre no PC, é importante realçar que grande parte da eficiência do sistema se deve ao software esqueleto que foi desenvolvido para o DSP.

### Software do DSP

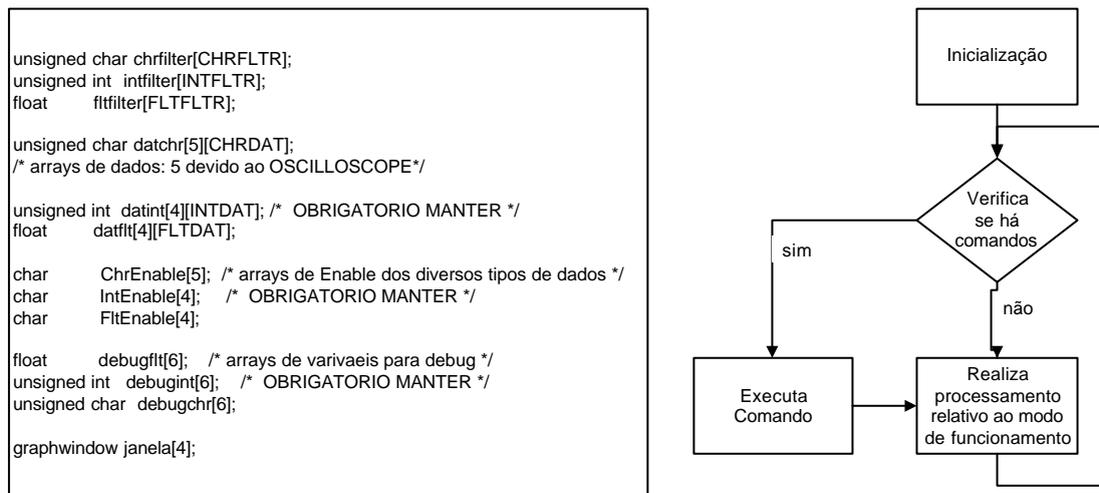
O software que corre no DSP é um programa servidor, que executa comandos que lhe chegam do PC através dos FIFOS. Quando não existem comandos para executar o DSP corre as funções referentes aos modos de funcionamento que estão activos. Nestas funções o programador que utiliza a

plataforma de teste pode implementar as suas funcionalidades que podem ser de processamento ou de leitura de dados do ADC.

Para que o utilizador da plataforma de teste se pudesse concentrar apenas no código do seu processamento específico, foi criada uma estrutura de dados e comandos que estabelecem toda a comunicação entre o DSP e o PC.

A estrutura de dados é composta por um conjunto de vectores com todos os tipos de variáveis para maior flexibilidade. Esta estrutura é enviada para o PC sempre que este faz o pedido. Desta forma, é possível ver graficamente os dados que estão na memória do DSP, como veremos na próxima secção deste capítulo, permitindo o *debug* do processamento em desenvolvimento.

Na figura 3 está representada a estrutura de dados existente no DSP e o fluxograma do seu funcionamento.

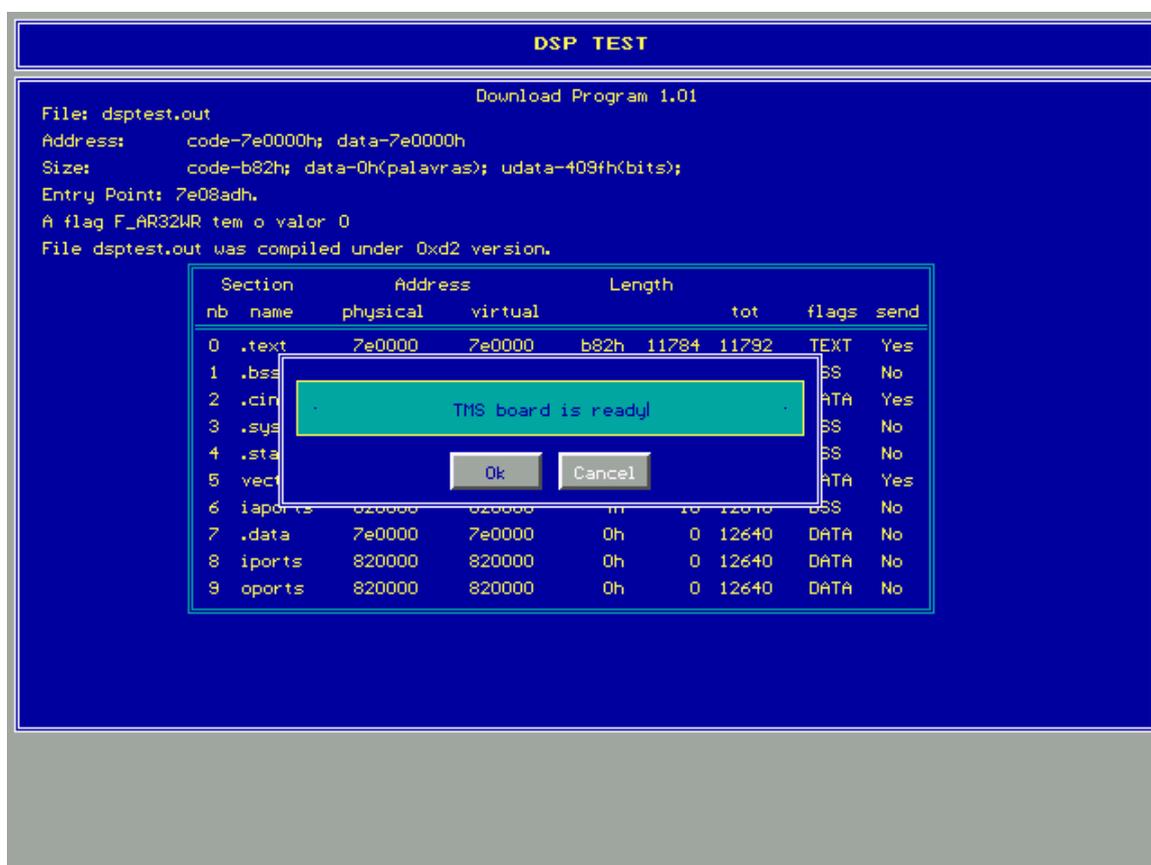


**Figura 3 - Fluxograma e estrutura de dados do DSP**

## Software do PC

O software que corre no PC foi desenvolvido com o objectivo de proporcionar um ambiente gráfico de controlo e visualização dos dados que se encontram a ser processados no DSP.

O software começa por inicializar o módulo de hardware descarregando o software do PC para o DSP, através dos FIFOS de comunicação. Uma imagem do software após efectuar esta acção é apresentado na figura 4.



**Figura 4 - Imagem da inicialização do software**

De seguida podemos configurar os diversos parâmetros do software de forma a visualizar os dados desejados na forma desejada. Para isso foi desenvolvido um menu de opções de dados. Na figura 5 podemos ver esse

menu. Conforme se pode verificar podemos inibir ou mostrar os dados, dar-lhes um título, configurar o número de pontos, tal como os limites mínimo e máximo do eixo dos YY no gráfico a apresentar, permitindo ainda que esta configuração seja automática. Finalmente devemos definir se os dados são de tipo *float*, *int*, ou *char*. No fim de configurar todos os parâmetros, é dada a opção de gravar um ficheiro onde é guardada a configuração.

Na figura 6 é apresentado o menu que permite configurar as variáveis que estarão visíveis em rodapé para *debug* durante a aquisição. Tal como para os vectores de dados é possível activar/desactivar, configurar o nome, o tipo de dado e o seu índice na estrutura de dados.

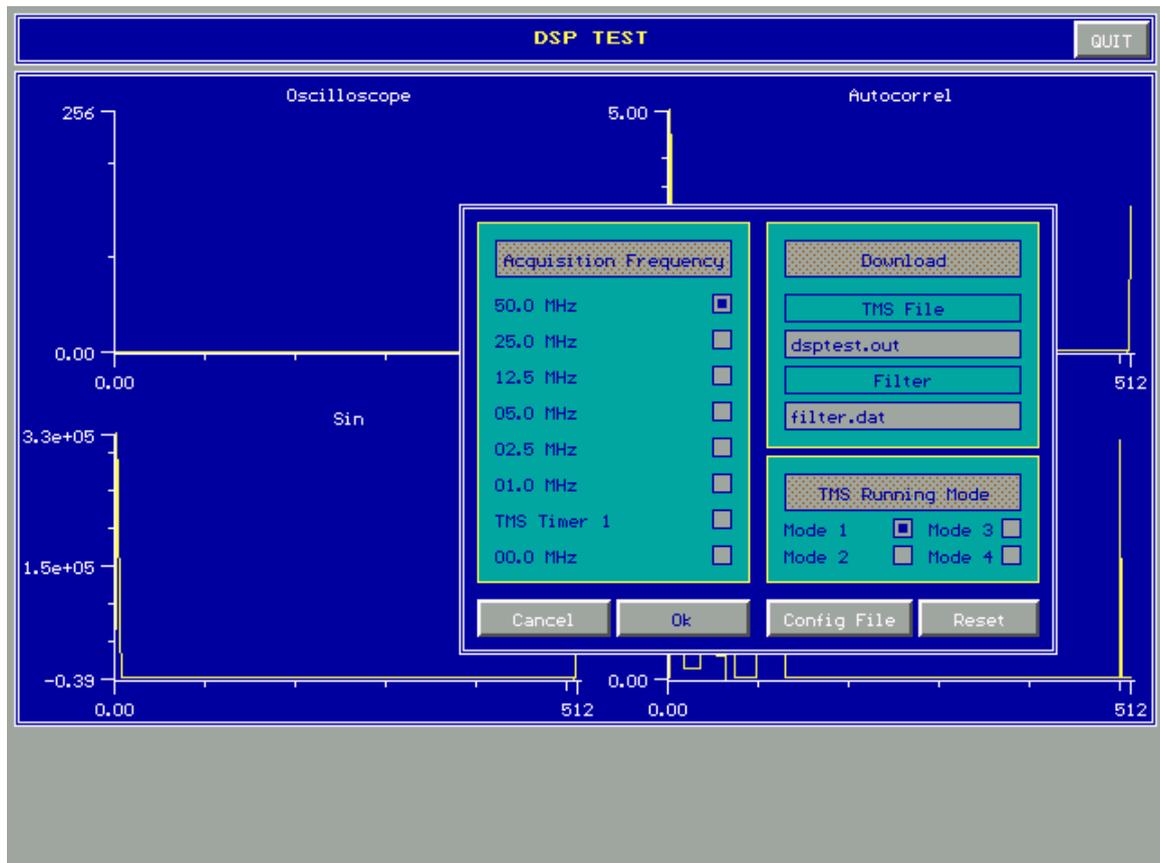


**Figura 5 - Configuração dos dados e sua visualização**



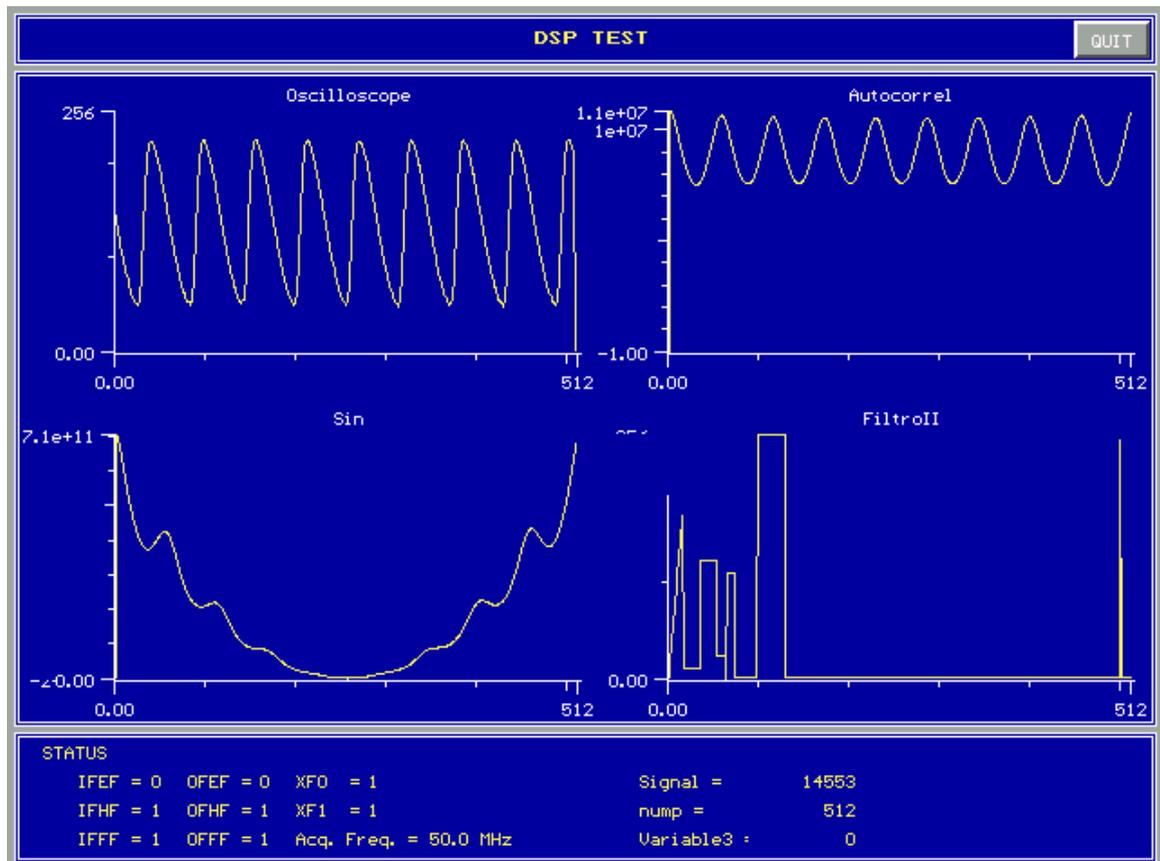
**Figura 6 - Configuração de visualização das variáveis**

Após a configuração dos dados e variáveis a visualizar, podemos configurar as opções mais relacionadas com o hardware. Nesta janela de diálogo é possível configurar a frequência de aquisição, o ficheiro que desejamos fazer *download* para o DSP, tal como o filtro, e ainda o modo em que queremos que o DSP corra, correspondendo a cada modo uma função de processamento a que o utilizador deseja fazer *debug* ou simplesmente ver os dados a ser processados.



**Figura 7 - Janela de diálogo com configurações diversas**

Estamos finalmente em condições de carregar no botão *Start* que inicia o processo de aquisição e processamento dos dados. Os dados são mostrados no écran de acordo com as configurações executadas. Na figura 8 podemos ver as 4 janelas de dados que pedimos para serem monitorizadas. Em rodapé podemos ver as variáveis de estado do sistema como as *flags* dos *FIFOs* e os pinos XF0 e XF1 do DSP, e ainda as variáveis que configurámos para serem mostradas como a *signal* que indica o número de medidas adquiridas e o número definido de amostras de cada medida.



*Figura 8 - Imagem da aplicação em modo de aquisição*

# Capítulo 6

## Osciloscópio Digital

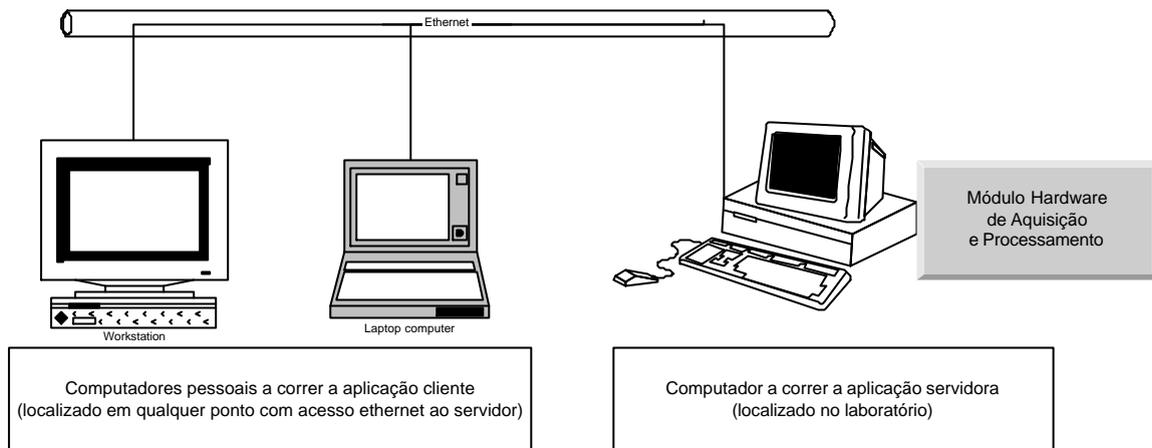
### Introdução

Neste capítulo é descrito o osciloscópio digital e analisador multicanal digital com possibilidade de controlo e visualização de dados remota, desenvolvido com base no mesmo módulo de hardware.

O sistema é dividido em três grandes blocos de software:

- Software do DSP - O software de mais baixo nível corre no DSP, tendo por objectivo gerir os dados adquiridos pelo ADC, processando-os de forma a gerar os dados do analisador multicanal de acordo com o procedimento descrito no projecto de curso desenvolvido em paralelo com este.[Cardoso]
- Software Servidor TCP/IP - Este software tem por objectivo fazer o interface entre o software de visualização de dados e controlo remoto e o módulo de aquisição e processamento.
- Software Cliente TCP/IP de Visualização e Controlo - Este software é a parte mais visível, tendo por objectivo servir de interface entre os dados adquiridos e o utilizador.

Na figura está representado um esquema do sistema baseado na tecnologia servidor/cliente.



**Figura 9 - Estrutura do Sistema**

## Software do DSP

O software que corre no DSP nesta aplicação é em tudo semelhante ao descrito no capítulo anterior, pelo que não será de novo descrito neste capítulo.

A única diferença foi a integração, da forma descrita no capítulo anterior, das rotinas desenvolvidas no projecto de processamento digital de sinal [Cardoso] que permitem a integração de um analisador multicanal no sistema.

Os dados do osciloscópio digital são fornecidos directamente do que é lido do ADC usando o *trigger* de hardware disponibilizado pelo módulo, ou um *trigger* de software baseado em rotinas de detecção de pulso.

## Software Servidor

O software que comunica com o módulo de aquisição é o chamado programa servidor. Trata-se do programa que gere todos os acessos ao hardware, não permitindo que mais que um cliente aceda simultaneamente ao hardware, o que poderia resultar num mau funcionamento ou mesmo avaria do sistema.

A partir deste software o operador tem acesso, através de um interface gráfico, às funcionalidade básicas do sistema como inicialização, descarregamento do programa do DSP na placa, gestão dos portos, escrita e leitura de bytes com comandos simples, etc.

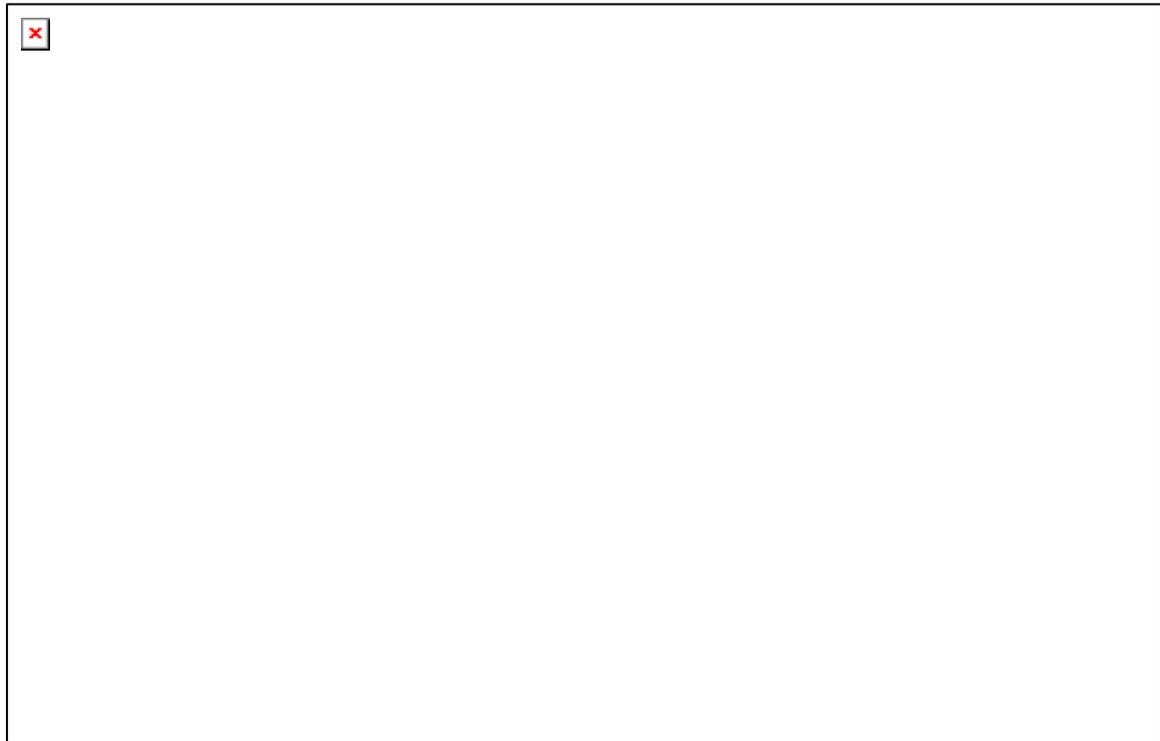
Por outro lado, o programa servidor efectua a função à qual deve o seu nome. Quando instruído para tal pelo utilizador, o programa servidor inicia um *socket* TCP/IP, abrindo um canal de comunicação ao exterior, ao qual se poderão ligar os programas clientes localizados no mesmo computador ou em qualquer ponto da rede. A programação orientada por objectos utilizada na construção do programa permite a fácil substituição desse meio de comunicação por qualquer outro método, ou forma de comunicação entre aplicações. Os motivos que levaram à escolha dos *sockets* TCP/IP foi o facto de ser uma tecnologia presente em todos os modernos sistemas operativos e ser o protocolo de comunicação mais usado na internet, permitindo desta forma o controlo do sistema de qualquer ponto do globo.

A partir do momento em que o servidor expõe o seu canal de comunicação, qualquer programa cliente com acesso à Internet poderá ligar-se, a fim de enviar controlos segundo um protocolo muito simples, mas altamente eficaz, ou receber os dados adquiridos, podendo monitorizar o que está a acontecer com uma qualquer peça de equipamento ou detector de física nuclear no laboratório.

As mensagens recebidas pelo servidor podem ser executadas pelo próprio servidor, caso lhe sejam dirigidas ou de imediato reencaminhadas para o FIFO de comunicação entre o PC e o DSP, para que o DSP execute o comando.

Sempre que se trate de mensagens que careçam de resposta, o programa servidor manterá a ligação com o programa cliente que originou o pedido. A resposta ser-lhe-á enviada logo que o comando tenha sido executado e os dados estejam disponíveis no servidor, uma vez que, quase sempre, esses dados têm origem no DSP.

Como se pode ver, trata-se de um esquema de encaminhamento de informação muito simples, o que lhe dá uma enorme flexibilidade, assim como uma enorme eficácia.



**Figura 10 - Software Servidor**

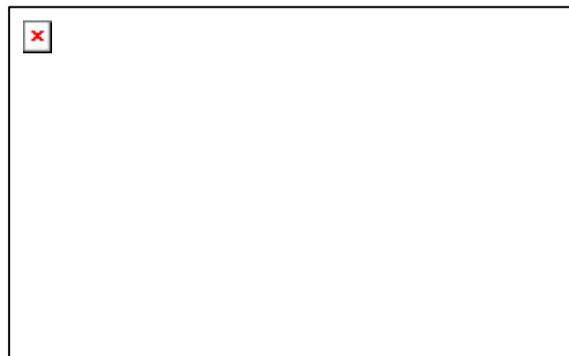
Na figura está representado o software servidor. Do lado esquerdo da figura podemos ver o estado de algumas variáveis do sistema como por exemplo se existe um programa do DSP na memória (*Program Loaded*), se ele foi enviado para o DSP (*Program Downloaded*), se o servidor está a funcionar (*Server Available*), tal como todos os registos do porto de leitura do módulo de hardware (PCEF, PCFF, DSPEF, DSPFF, HOLDA, XF0, XF1). Do lado direito podemos ver os pormenores do programa do DSP e as zonas de memória por ele usadas.

Podemos também verificar pela figura alguns comandos básicos da aplicação como Server->Enable; Server->Disable; Program->Download; Fifo->Send Byte; Fifo->Get Byte...

## Software Cliente

O programa cliente é aquele em que foram investidos mais cuidados em termos de apresentação gráfica e interface com o utilizador. Este programa faz o interface entre todo o sistema e o utilizador, sendo a parte mais visível do trabalho desenvolvido.

Quando inicia o programa cliente tem de se ligar ao servidor através do seu endereço IP ou nome na rede. Assim sendo o primeiro comando a ser usado é o Board->Connect, que faz activar a janela de diálogo do servidor, onde é pedido o endereço do servidor, que pode ser a própria máquina ou qualquer outra ligada a esta por rede ethernet e que esteja a correr o software servidor.



**Figura 11 - Menu de Ligação ao Servidor**

Em seguida estamos em condição de comunicar com o software servidor, podendo optar por lhe enviar um conjunto de comandos básicos, como iniciar ou parar a aquisição usando os comandos Oscilloscope->Start ou Oscilloscope->Stop. No caso do MCA temos acesso aos comandos MCA->Start, MCA->Stop e MCA->Reset.



**Figura 12 - Menu Opção**

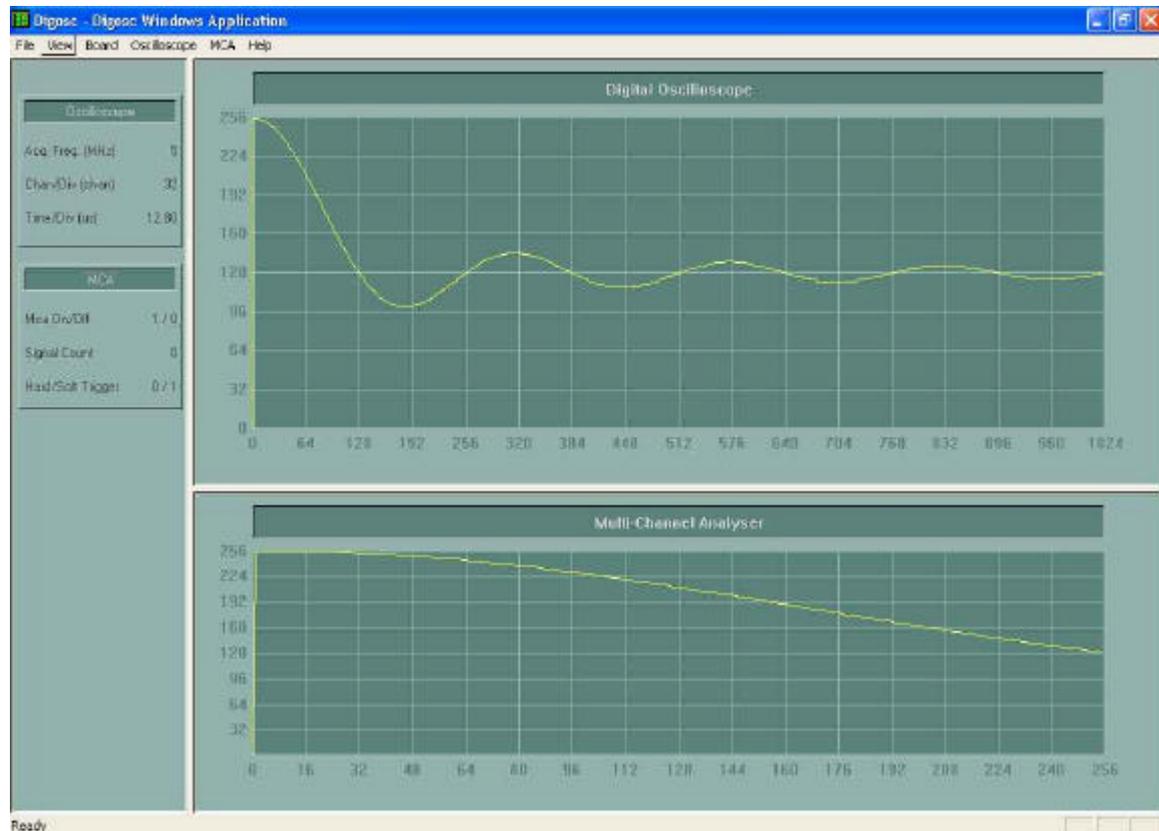
Para além disso o software tem um menu de opções onde é possível configurar o sistema. O menu opções está representado na figura, onde se pode escolher o número de amostras adquiridas em cada medida, o intervalo de refrescamento do MCA em número de contagens e o intervalo de tempo entre medidas do osciloscópio. Este intervalo de tempo depende essencialmente da velocidade do PC usado e da largura de banda da comunicação entre o cliente e o servidor. Finalmente podemos escolher neste menu o nível de *trigger* e se usaremos um *trigger* por software ou o *trigger* de hardware disponível no módulo.



**Figura 13 - Menu de Configuração da Taxa de Aquisição**

No menu da placa podemos também configurar a taxa de aquisição do ADC usando o comando Board->Clock. A opção de reinicialização da placa mostrou-se necessária, uma vez que podendo o utilizador estar fisicamente

afastado do módulo de hardware, seria útil colocar remotamente todo o sistema no seu estado inicial.



**Figura 14 - Vistas da Aplicação Cliente**

Uma vez que, como já foi sublinhado, este software é essencialmente de visualização de dados e de interface ao utilizador, foi dada especial atenção a estes aspectos. Na figura podemos ver o aspecto da aplicação, onde estão representados do lado direito os dados do osciloscópio e do MCA provenientes do módulo de hardware. Do lado esquerdo estão representados um conjunto de parâmetros que habitualmente se encontram em todos os osciloscópios e MCA. De notar que estas três representações estão implementadas em vistas diferentes, sendo totalmente independentes umas das outras. Para permitir uma melhor visualização dos dados adquiridos, foi implementado no menu vistas uma opção de *zoom*, que tem como opção

padrão o *zoom* automático, mas que pode ser definido para valores entre 50% e 400%.

Finalmente, temos um comando básico em termos de software, mas que acrescenta um enorme valor a um osciloscópio. Trata-se da possibilidade de gravar num ficheiro os dados adquiridos, para posterior análise e visualização.

# Capítulo 7

## Conclusões e Trabalho Futuro

O trabalho apresentado provou a sua inovação e utilidade na área da instrumentação pelo número de publicações que originou em conjunto com outros trabalhos desenvolvidos no Laboratório de Electrónica e Instrumentação (LEI). [Malaquias2][Cruz][Basílio].

Para além da utilidade do trabalho desenvolvido, um projecto de curso tem como objectivo a formação do aluno que o desenvolve. Neste campo penso ter sido fortemente beneficiado no trabalho que tive a oportunidade de desenvolver. A minha formação ficou fortemente enriquecida nas áreas de processamento digital de sinal, estrutura de sistemas de aquisição de dados, desenvolvimento de software de baixo nível (para o DSP) e de alto nível (interfaces gráficas em ambiente Windows™). Apesar desta excelente formação que me foi proporcionada na área técnica e científica, aquela que eu considero de maior valor para a minha formação como engenheiro foi a possibilidade de um trabalho em equipa com uma estrutura e organização eficaz, que permite o aproveitamento de trabalhos desenvolvidos e recursos do Laboratório de Electrónica e Instrumentação e a certeza de que o trabalho por mim desenvolvido será aplicado e útil em outros projectos desta área, tornando-se muito mais que um simples trabalho académico.

Algumas opções de desenvolvimento futuro são um novo módulo de hardware com capacidade de multiprocessamento para uma maior capacidade de processamento de impulsos em tempo real e o desenvolvimento de aplicações cliente para outras plataformas, aumentando a portabilidade do sistema para outras plataformas. Nesta área, o recente desenvolvimento das tecnologias de TCP/IP sobre rede móvel, permitem o desenvolvimento da aplicação cliente para PDA's e outros aparelhos ultra

portáteis que correm sistemas operativos como o WindowsCE™. Baseado neste trabalho foram realizados alguns desenvolvimentos no LEI tendo em conta estas perspectivas de trabalho.[Basílio]

## Bibliografia

[Goldberg] - Harold Goldberg, "What is Virtual Instrumentation?", IEEE Instrumentation & Measurement Magazine, Dezembro 2000.

[Microsoft] - Microsoft Corporation, "The Windows Interface Guidelines for Software Design, 1996.

[Malaquias] - José Luis Malaquias, "Sistema Integrado de Aquisição de Dados para Física Nuclear", Tese de Mestrado, 1996.

[Cardoso] - João Manuel Cardoso, "Processamento Digital de Impulsos em Espectroscopia Nuclear", Relatório da Disciplina de Projecto, 1997

[Malaquias2] - José Luis Malaquias, Pedro Almeida, Paulo Amílcar, Nuno Cruz, Carlos M. B. A. Correia, "A integrated System for Nuclear Data Acquisition", IEEE Transactions on Nuclear Science, vol 44, no. 3, pp 411-415, Junho de 1997.

[Cruz] - Nuno Sérgio Cruz, Pedro Almeida, Paulo Amílcar, J. Basílio Simões, José Luís Malaquias e Carlos M. B. A. Correia, "Um Sistema de Alto Desempenho para Aquisição e Processamento Digital de Sinais", comunicação apresentada na FÍSICA 96, 10ª Conferência Nacional de Física, Faro, Setembro de 1996, publicada nos resumos da conferência, XI - PO (18), pp.560.

[Basílio] - J. Basílio Simões, João Cardoso, Nuno Cruz e Carlos M. B. A. Correia, "A Windows CE Portable Spectrometer", apresentação no IEEE NSS-

99 (IEEE Nuclear Science Symposium 1999), Seattle, Washington, EUA de 26 a 28 de Outubro de 1999.