



FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Departamento de Engenharia Electrotécnica e de Computadores

Rua dos Bragas, 4099 Porto Codex, PORTUGAL

Telef. 351-2-317105/107/412/457 · Telex 27323 FEUP P · Telefax 351-2-319280

Projecto de um Sistema de Software na área da Supervisão, Controlo e Aquisição de Dados (SProcess)

Eduardo Luís Ribeirinha Cardoso de Carvalho

Porto, Abril de 1992

6.8.1630

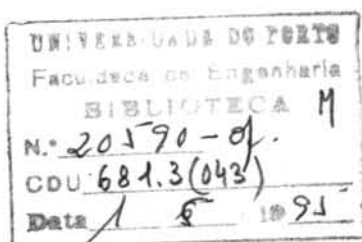
ELV.

Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores

**Projecto de um Sistema de Software na área da
Supervisão, Controlo e Aquisição de Dados
(SProcess)**

Trabalho de Dissertação do Curso de Mestrado em Engenharia
Electrotécnica e de Computadores

Eduardo Luís Ribeirinha Cardoso de Carvalho
Licenciado em Engenharia Química pela Faculdade de Engenharia da
Universidade do Porto



043 M
C 322 p
EX. 32

Porto, Abril de 1992

Agradecimientos

Agradecimentos

Desejo agradecer ao Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto a realização deste curso de mestrado em Engenharia Electrotécnica e de Computadores - Informática Industrial, e à Escola Superior de Biotecnologia da Universidade Católica Portuguesa as condições que me permitiram frequentá-lo.

Agradeço também a todos aqueles, que, de uma forma ou outra, me ajudaram na concretização dos vários objectivos associados a este curso de mestrado:

- Doutor Alberto Sereno, pela motivação e decisão inicial;
- Eng.º Eduardo Beira, pelo estímulo dado na finalização;
- Ao Francisco Silva e Paulo Jorge Rocha Pereira, e em especial ao Pedro Guedes pela ajuda no trabalho de programação;
- Aos meus colegas de curso, com os quais constituí um excelente grupo de estudo;
- Aos meus colegas de trabalho no Centro de Cálculo e Tecnologias da Informação, da Escola Superior de Biotecnologia, pela colaboração que sempre me prestaram;
- Aos meus amigos e minha família, pelo apoio e paciência durante estes anos de mestrado;
- À Beatriz Ribeiro, Pedro Guedes e Carlos Sousa, pelas rotinas de software que me autorizaram a usar.

Seja-me permitido destacar o Prof. Doutor José Marques dos Santos pela orientação dispensada que me permitiu concretizar este trabalho de dissertação do curso de mestrado.

Índice

Índice

Agradecimentos	I
Índice	III
Lista de Figuras	VI
Introdução	IX

Capítulo 1

Sistemas de Supervisão, Controlo e Aquisição de Dados

1.1 — Introdução.....	1/2
1.2 — Características e Funções.....	1/4
1.3 — Os Sistemas de Tempo Real	1/10
1.4 — O Desenvolvimento do Software.....	1/19

Capítulo 2

Projecto de *Software*

2.1 — Especificação de Requisitos.....	2/2
2.1.1 - Especificação de Arquitectura do Sistema	2/2
2.1.2 - Especificação Funcional	2/4
2.2 — Análise do Sistema.....	2/7
2.2.1 - Fluxo de Informação.....	2/8
2.2.2 - Fluxo de Controlo.....	2/15
2.3 — Concepção do Sistema	2/22
2.3.1 - Arquitectura da Aplicação	2/25
2.3.2 - Concepção dos Módulos	2/38
2.3.3 - Concepção de Dados	2/40
2.3.4 - Concepção da Interface	2/45
2.4 — <i>Packaging</i>	2/47
2.4.1 - <i>Packaging</i> de Programas	2/47
2.4.2 - <i>Packaging</i> de Dados.....	2/49
2.4.3 - Estrutura de Directórios	2/50
2.4.4 - <i>Packaging</i> de Procedimentos.....	2/51

Capítulo 3

Implementação: Exemplos

3.1 — Exemplo 1 - Criação da estrutura de dados Tabela.....	3/2
3.2 — Exemplo 2 - Processo em <i>background</i> : Monitorização com o Sistema de Aquisição de Dados e Controlo Keithley 500.....	3/3
3.3 — Documentação do código.	3/11

Capítulo 4

Teste e Validação: Aplicação prática

4.1 — Caso de estudo: Tanque de retenção com nível constante.	4/2
4.2 — Simulação do Processo e da Interface.....	4/16

Capítulo 5

Conclusões.....	5/1
-----------------	-----

Referências e Bibliografia.....	R&B/1
---------------------------------	-------

Apêndices

A	Análise de Produtos Comerciais Tipo SCADA	A/1
B	Dicionário de Dados: Entidades, Processos e Depósitos.....	B/1
C	Concepção dos Módulos: fichas de especificações	C/1
D	Descrição Técnica do SAC Keithley 500	D/1

Lista de Figuras

Lista de Figuras

Figura 2-1:	Diagrama de Blocos do Sistema.....	2/3
Figura 2-2:	Diagrama de Fluxo de Dados (DFD) de Contexto	2/10
Figura 2-3:	DFD de 1ª Ordem - Aplicação	2/11
Figura 2-4:	DFD de 2ª Ordem - Configuração da Aplicação.....	2/12
Figura 2-5:	DFD de 2ª Ordem - Exploração <i>On-line</i>	2/13
Figura 2-6:	DFD de 2ª Ordem - Operação <i>Off-line</i>	2/14
Figura 2-7:	Diagrama de Fluxo de Controlo.....	2/17
Figura 2-8:	Diagrama de Transição de Estados - Aplicação.....	2/19
Figura 2-9:	Estrutura tipo de um processo.....	2/22
Figura 2-10:	Mapa de Estrutura (ME) de Contexto.....	2/26
Figura 2-11:	ME de 1ª Ordem - Aplicação.....	2/27
Figura 2-12:	ME de 1ª Ordem - Exploração <i>On-line</i>	2/28
Figura 2-13:	ME de 2ª Ordem - Segurança.....	2/29
Figura 2-14:	ME de 2ª Ordem - Configurar Sinópticos.....	2/29
Figura 2-15:	ME de 2ª Ordem - Configurar Variáveis/Processo.....	2/30
Figura 2-16:	ME de 2ª Ordem - Configurar Variáveis/Sistema	2/31
Figura 2-17:	ME de 2ª Ordem - Configurar Gráficos	2/31
Figura 2-18:	ME de 2ª Ordem - Configurar Tabelas	2/32
Figura 2-19:	ME de 2ª Ordem - Inicialização.....	2/33
Figura 2-20:	ME de 2ª Ordem - Alarmes	2/34
Figura 2-21:	ME de 2ª Ordem - Análises/Gráficos	2/34
Figura 2-22:	ME de 2ª Ordem - Análises/Tabelas	2/35
Figura 2-23:	ME de 2ª Ordem - Sinópticos.....	2/35
Figura 2-24:	ME de 2ª Ordem - Operação	2/36
Figura 2-25:	ME de 2ª Ordem - Terminação	2/37
Figura 2-26:	ME de 2ª Ordem - Conversão	2/38
Figura 2-27:	Arquitectura do <i>Buffer</i> Tabela.....	2/42
Figura 2-28:	Arquitectura das Filas Circulares.....	2/43

Figura 3-1:	Código retirado do módulo Inicialização - inicia ().....	3/3
Figura 3-2:	Código do módulo Monit_instala - monit_instala ()	3/5
Figura 3-3:	Código do módulo Monitorização - monit_int ()	3/11
Figura 4-1:	Apresentação esquemática do tanque de retenção	4/2
Figura 4-2:	Écran de configuração das variáveis de sistema	4/4
Figura 4-3:	Écran de configuração do canal 0	4/5
Figura 4-4:	Écran de configuração do canal 1	4/6
Figura 4-5:	Configurações para os canais 2 a 8	4/6
Figura 4-6:	Desenho do quadro sinóptico.....	4/7
Figura 4-7:	Definição dos parâmetros do sinóptico.....	4/7
Figura 4-8:	Écran de configuração do gráfico 1.....	4/8
Figura 4-9:	Écran de configuração do gráfico 2.....	4/9
Figura 4-10:	Écran de configuração da tabela 1	4/9
Figura 4-11:	Evolução do nível do tanque (P=1, I=0.6)	4/11
Figura 4-12:	Evolução do nível do tanque (P=1, I=0)	4/13

Introdução

Introdução

O relatório que se apresenta refere-se ao trabalho para dissertação, parte integrante do Curso de Mestrado em Engenharia Electrotécnica e de Computadores, perfil de Informática Industrial, iniciado, em 1988, no Departamento de Engenharia Electrotécnica e de Computadores da Faculdade de Engenharia da Universidade do Porto.

O objectivo proposto foi o de estudar e projectar um sistema de *software* * na área da Supervisão, Controlo e Aquisição de Dados (SCADA - *Supervision, Control and Data Aquisition*), desenvolvendo conhecimento e experiência na produção de *software* interactivo para aplicações em tempo real, configurável a diferentes situações.

O trabalho deverá enquadrar-se e ser um contributo para o desenvolvimento a médio prazo de uma linha de produtos de *software* na área das aplicações SCADA de uso geral.

Os sistemas tipo SCADA foram entendidos como sistemas de *software* mais orientados para funções de supervisão, envolvendo a aquisição de dados necessária mas não assegurando o controlo directo de processo, considerado função do equipamento de controlo local com os quais o sistema poderá ter possibilidade de dialogar.

Pelas características destes sistemas e tipo de utilizações, quer industriais quer laboratoriais, era considerado importante dar ênfase à fase de Projecto do processo de produção de *software*. Foi assim proposto o desenvolvimento da Análise e Concepção de uma aplicação que abordasse as principais características dos sistemas tipo SCADA.

Foi, todavia, proposta a produção de uma aplicação, a encarar como protótipo para uma aplicação profissional, que permitisse identificar metodologias, dificuldades e passos limitantes. Este protótipo devia incorporar, sempre que possível, módulos e programas pré-existentes que permitissem produzir uma versão demonstrável e, portanto, alvo de validação e documentação.

O desenvolvimento da aplicação protótipo devia ser realizado em ambiente MS-DOS, usando a linguagem de programação Turbo C da Borland, para ser executado num microcomputador de uso generalizado tipo IBM AT, com 640 KB de RAM, écran gráfico EGA (mínimo) e disco rígido. Pretendeu-se que a aplicação pudesse ser configurável para operar com diferente equipamento de aquisição de dados e controlo.

* Os termos de origem inglesa, para os quais a tradução não está generalizada, estão colocados em itálico. Nos capítulos de maior aplicação prática prescindimos todavia desta notação.

O trabalho realizado em função dos objectivos propostos, e agora apresentado, inclui, no Capítulo 1, uma caracterização dos denominados sistemas de Supervisão, Controlo e Aquisição de Dados — SCADA, sendo descritas as principais funções assim como as diferentes fases de desenvolvimento deste tipo de *software*.

No Capítulo 2 é apresentado o Projecto da aplicação protótipo SProcess, nas suas diferentes fases, com a Especificação de Requisitos, a Análise Funcional e a Concepção dos diferentes módulos a codificar.

Alguns aspectos da codificação são descritos em pormenor no Capítulo 3.

No Capítulo 4, a aplicação desenvolvida é testada e validada através de uma aplicação prática a um processo laboratorial simulado.

No Capítulo 5, são apresentadas conclusões do trabalho realizado e perspectivam-se eventuais desenvolvimentos.

Em apêndice, incluem-se uma análise comparativa de produtos comerciais tipo SCADA (Apêndice A), os resultados para codificação da Análise e Concepção efectuadas, nomeadamente o Dicionário de Dados (Apêndice B) e a Concepção de Módulos (Apêndice C), e uma Descrição Técnica do Sistema de Aquisição de Dados e Controlo utilizado (Apêndice D).

Os principais produtos do trabalho realizado são a caracterização dos sistemas SCADA, o projecto de *software* e a aplicação protótipo.

Foi então possível conhecer as especificidades dos sistemas SCADA e abordar aspectos do desenvolvimento formal de *software*, do processamento paralelo de tarefas e da utilização de sistemas digitais para aquisição de dados e controlo de processos.

No trabalho que se apresenta incluímos instrumentos que consideramos de utilidade no quadro de futuros desenvolvimentos, como sejam os diagramas de fluxo de dados, os mapas de estrutura, o dicionário de dados e as fichas de especificação para codificação da aplicação projectada.

CAPÍTULO 1

Sistemas de Supervisão, Controlo e Aquisição de Dados

1.1 — Introdução

Os sistemas de informação industriais podem ser apresentados em quatro níveis funcionais de acordo com a seguinte hierarquia: Gestão da Empresa, Gestão da Produção, Supervisão e Operação Directa.

Estes sistemas têm-se desenvolvido no sentido de uma informatização baseada em sistemas digitais com uma estrutura distribuída que possa incluir equipamento analógico pré-existente, e que ofereça a cada nível hierárquico uma integração das funções relevantes. A utilização, cada vez mais generalizada, de sistemas digitais tem como motivação a redução de custos por aumento da automatização e integração de funções, com ganhos de flexibilidade e de qualidade, e permitindo a inclusão de novas funções em sistemas de informação, abrindo novas perspectivas à gestão industrial.

Aos níveis hierárquicos considerados podem corresponder tipicamente os seguintes sistemas digitais: *mainframe*, mini computador, micro computador e controladores. Um sistema de informática industrial pode incluir como principais componentes: sensores que asseguram a interface de medida entre o sistema e o processo; actuadores que convertem instruções de comando em acções efectivas a nível do processo; interfaces de Input/Output que asseguram a ligação entre o sistema informático e os sensores e actuadores; equipamento de comunicações que interligam os vários subsistemas envolvidos; equipamento informático que inclui os computadores utilizados nos diferentes níveis hierárquicos; o *software* de sistema, de desenvolvimento e de aplicação que, conforme o nível considerado, terá de oferecer características específicas.

Os sistemas de *software* de aplicação para Supervisão, Controlo e Aquisição de Dados (SCADA - *Supervision, Control and Data Aquisition*) cobrem funções aos níveis da supervisão de processo e da operação directa de equipamento processual.

As funções implementadas por este tipo de *software* podem incluir: Monitorização do processo e registo da sua evolução; Controlo automático e Operação manual; Análise, Optimização e Supervisão do processo; Comunicações e Integração no sistema de gestão da produção.

Este tipo de *software* de aplicação caracteriza-se fundamentalmente pela necessidade de traduzir o comportamento real de um processo e de possibilitar intervenções na evolução desse processo, de acordo com restrições de tempo que caracterizam a sua dinâmica, de forma a manter o comportamento em conformidade com as especificações existentes.

O *software* tem que ser capaz de gerir, em tempo real, tarefas simultâneas que têm que ser tratadas em tempos pré-definidos e sincronizadas de acordo com prioridades.

O funcionamento em tempo real do *software* de aplicação depende também de todo o sistema de *software/hardware* que o suporta, incluindo o sistema operativo, a linguagem de programação, as características da base de dados eventualmente usada, o sistema de comunicações e a arquitectura de *hardware*.

Uma outra característica dos sistemas SCADA, que representa uma restrição forte, é a exigência de fiabilidade que os processos industriais pressupõem. A fiabilidade, aqui, significa uma baixa probabilidade de paragem do sistema, já que, em geral, pode ser aceitável alguma degradação no seu funcionamento.

Pode assim ser necessário prever no projecto de *software* formas alternativas de funcionamento (equipamento de emergência, operação manual) e procedimentos de recuperação de erros ou falhas.

Por outro lado é necessário ter em conta que a aplicação é para ser usada em ambientes industriais, sujeita a intervenção de técnicos e operadores, e que as funções que executam podem ser críticas pelas consequências humanas, económicas e ecológicas associadas, áreas cada vez mais regulamentadas e fiscalizadas.

Pelas suas características, o *software* SCADA é muito dependente do sistema envolvente e, pela complexidade que pode atingir, pressupõe um processo de desenvolvimento rigoroso.

Neste Capítulo 1 são descritas as funções que em geral se esperam de um *software* de aplicação do tipo SCADA, é caracterizado o sistema envolvente para aplicações de tempo real e o processo de desenvolvimento do projecto de *software*, baseado, fundamentalmente, numa consulta extensiva a artigos em publicações recentes, nomeadas nas referências [1] a [45].

1.2 — Características e Funções

O *software* de aplicação SCADA pode ser proprietário e exigir um determinado sistema de *hardware* ou mesmo ser adaptado a um dado processo. Estamos aqui interessados em produtos SCADA configuráveis, interactivamente, para um dado processo e sistema de *hardware*. No apêndice A é apresentada uma análise de diferentes destes produtos comerciais.

Tipicamente, os produtos deverão incluir dois subsistemas bem distintos, um correspondente a desenvolvimento e outro a exploração.

O subsistema de desenvolvimento oferece todas as facilidades para configurar o funcionamento da aplicação em tempo real. Deverá ser independente, não sendo necessário existir instalado após completada a configuração, libertando recursos e tornando mais económica uma utilização multiposto. As configurações poderão ser especificadas em programas modulares e independentes, de preferência normalizados ou de uso generalizado (ex: editor gráfico, *package* estatístico, gerador de relatórios, etc.), originando uma base de dados de configurações a serem usadas na fase de exploração.

Num sistema aberto deverá ser oferecido um ambiente para acesso à base de dados e para produzir programas de ligação (*drivers*) para comunicação equipamento de aquisição e controlo. As configurações devem prever a possibilidade de integrar programas de utilizador para execução de diferentes funções.

As configurações da aplicação ao processo devem incluir os seguintes aspectos:

- interface com o operador
- aquisição e controlo
- alarmes
- análises
- relatórios
- *hardware* instalado.

O subsistema de exploração implementa as funções do SCADA [46] durante a operação em tempo real, de acordo com o especificado nas várias configurações pré-existentes.

As funções típicas de um sistema SCADA incluem:

- aquisição de dados do processo, com eventual arquivo em disco

- visualização de quadros sinópticos representativos das várias partes do processo
- análise dos dados recolhidos, automática ou manual
- monitorização e tratamento de situações de alarme
- operação manual do processo
- controlo digital directo
- optimização do processo
- supervisão do processo: interacção com os controladores locais
- comunicação com equipamento local no processo e com computadores de níveis hierárquicos iguais ou superiores
- emissão de relatórios sobre a aplicação e o processo
- segurança e protecção na utilização da aplicação
- ajudas e informações ao utilizador
- exportação de dados de exploração para tratamento *off-line* em aplicações de uso comum.

Aquisição de dados do processo

Execução de leituras periódicas de variáveis de processo, sendo os valores guardados temporariamente em memória e opcionalmente gravados em disco. A velocidade de varrimento das variáveis configuradas como de entrada pode ser seleccionada pelo utilizador dentro das opções existentes ou o valor ser livremente especificado pelo utilizador. Poderá haver um modo de funcionamento "tão rápido quanto possível". Cada variável (ou conjunto de variáveis) poderá ter a sua própria velocidade de aquisição de dados.

A cada variável corresponde um canal no sistema de aquisição de dados caracterizado pela frequência de leitura, ganho e filtragem. As variáveis devem poder ser configuradas quanto ao tipo de sinal (digital ou analógico), à formula de cálculo a aplicar, à ligação com outras variáveis, ao protocolo da comunicação, às unidades, ao tipo da grandeza física associada, etc.

Além destas variáveis de entrada, que podemos classificar com reais, poderemos ter variáveis virtuais cujos valores sejam calculados ou gerados, ou cujos valores sejam lidos a partir de ficheiros em disco.

O arquivo de dados em disco, além de ser opcional, pode ser limitado no tempo, produzir ficheiros circulares ou incluir um filtro que, por exemplo, execute o arquivo, apenas, se

houver uma alteração mínima no valor da variável, assegurando assim uma redução de dados em disco. Os dados podem ser guardados numa base de dados para facilitar o seu uso posterior.

O *software* deverá permitir usar diferentes sistemas de aquisição de dados e oferecer um ambiente de desenvolvimento de drivers para sistemas não contemplados.

Estas funções de aquisição e arquivo de dados devem ser executadas em paralelo e em *background*. Se esta função de aquisição for exigente em velocidade de leitura e no número de pontos a serem lidos, a aplicação SCADA poderá ter que ser compatível com um sistema operativo para funções de tempo real e, eventualmente, suportar *hardware* adicional, com capacidades de processamento de comunicações e de aquisição de dados.

Visualização de quadros sinópticos

Fazendo uso de écrans desenhados para representarem o processo, a aplicação deverá oferecer uma actualização numérica, gráfica e animada do quadro sinóptico que tiver sido seleccionado para visualização de acordo com os dados recolhidos do processo. O desenho destes écrans é produzido através de um editor, em modo gráfico ou carácter, que inclua uma biblioteca de símbolos e uma biblioteca de parâmetros actualizáveis em exploração.

Estes parâmetros devem incluir gráficos para representação *on-line* de variáveis do processo, ainda que numa forma simplificada.

Os editores gráficos deverião poder ser de uso generalizado, para economia e facilidade de utilização, e independentes do resto da aplicação já que só seriam usados na fase de configuração.

Esta função, que corresponde aos quadros sinópticos das tradicionais salas de controlo, poderia tirar partido de um sistema com mais que uma consola para visualizar simultaneamente vários écrans. No entanto, a implementação desta função pode permitir que écrans estejam ligadas logicamente ou que possa haver uma visualização rotativa que em determinada cadência percorra os vários écrans. É uma função que pode ter uma execução em *foreground* e apenas quando um écran de visualização estiver seleccionado.

Análise de Dados

Na operação em tempo real devem ser oferecidas análises de dados pré-configuráveis com interesse para áreas como a qualidade ou a gestão da produção, e que são executadas

sobre os últimos valores adquiridos para as variáveis envolvidas com os resultados apresentados numa forma gráfica ou tabular. Pode corresponder à utilização de um *package* de geração de gráficos em que os dados submetidos estão em memória e correspondem a históricos recentes, podendo a execução ser em *foreground*. As análises de dados pretendidas em algumas aplicações podem ser complexas: FFT, integrações, derivações, ajustes, etc.

Ainda em tempo real pode ser oferecida uma análise e visualização de dados históricos, arquivados em disco.

Tratamento de Alarmes

As situações de funcionamento que tenham sido configuradas como anormais devem produzir alarmes visíveis durante a operação *on-line*.

A configuração de situações de alarme deve poder estar associada a cada variável (real ou virtual) do sistema e ser caracterizada por vários limites inferiores ou superiores, por desvio a um ponto estabelecido (*setpoint*), pela taxa de alteração de valor, etc. Determinados alarmes devem poder desencadear acções de forma automática. As mensagens de alarmes devem ser visíveis em qualquer écran da aplicação, e poder ser gravados em disco ou enviados a uma impressora para consultas posteriores. As mensagens devem ser explícitas e indicar o nome, o tipo, a variável associada, a hora, prioridade, etc.

A visualização de situações de alarmes pode estar ligada aos écrans sinópticos com um tipo de representação específica (mudança de cor, sinal sonoro, etc.). Um determinado tipo de alarme poderá provocar a visualização do écran sinóptico associado.

Os alarmes devem ser colocados em fila de atendimento de acordo com a prioridade e a ordem cronológica, sendo eliminados à medida que são validados, individualmente ou globalmente. Esta função de monitorização e tratamento de situações de alarme deverá ser executada em paralelo com a aquisição de dados e considerada como prioritária.

Operação

A operação directa do processo passa pela visualização das variáveis do sistema, relacionadas com as funções a nível de operação, e pela possibilidade de alterando valores disponíveis, actuar sobre o processo.

Esta operação, não automática, pode ser feita em ambiente próprio ou a partir dos écrans sinópticos, aqui com a vantagem de se poder monitorar os efeitos. A possibilidade de agrupar comandos de operação em macros facilitará a tarefa do operador. Esta função deverá ser executada em *foreground*.

Controlo

Deve ser possível definir de forma flexível as estratégias de controlo a partir de algoritmos pré-definidos: PID, ON/OFF, etc. Deverá ser possível sequenciar comandos para controlo em funcionamento *batch*, que agrupados constituam macros de execução calendarizada no tempo ou em função de eventos. Deverá ainda ser possível implementar acções de controlo baseadas em algoritmos executados por programas do utilizador.

Estas funções de controlo digital directo, a realizar pela aplicação, deverão poder ser executadas em *background*, em paralelo com a aquisição de dados, e podem ter que obedecer a restrições de tempo e ser consideradas prioritárias.

Optimização

O funcionamento do processo como um todo pode ser analisado e serem sugeridas alterações às condições de operação.

Como exemplo as estratégias de controlo podem ser optimizadas por um módulo de auto-sintonização (*self-tunning*), ou as acções que o operador deve executar em face de um problema no processo, podem ser sugeridas por um módulo do tipo sistema pericial.

Supervisão

As configurações estabelecidas na aplicação devem ser guardadas numa estrutura de base de dados que permita alterações individuais e de forma fácil. Deve ser possível visualizar, de forma global, a estratégia de aquisição de dados e controlo que estiver a ser executada.

Actuações sobre o processo, a nível de supervisão, deverão ser permitidas de forma a poderem ser alterados parâmetros de controlo, *setpoints*, programas dos controladores locais, ou mesmo aspectos da estratégia de aquisição e controlo.

Algumas das acções de supervisão podem ser indicadas por outras funções da aplicação como a optimização ou por outros computadores do sistema de informação instalado.

Comunicações

A integração num sistema distribuído é potenciada pelo suporte de comunicações em rede industrial que assegure troca de informações e partilha de ficheiros entre e nos diferentes níveis hierárquicos do sistema.

Relatórios

Devem ser oferecidos resumos da informação tratada pela aplicação, com interesse a diferentes níveis. A função pode ser implementada por um gerador de relatórios que pode fazer uso de relatórios tipo pré-formatados sobre dados do processo, configurações, análises, etc.

Os relatórios devem poder ser impressos, em disco ou papel, em tempo real e passíveis de serem calendarizados.

Segurança

Dada a responsabilidade que pode estar associada a algumas das funções deste tipo de aplicação pode ser importante dispor de protecções, por codificação ou identificação dos utilizadores, que hierarquizem e restrinjam o acesso às diferentes opções da aplicação.

Ajuda

Dada a complexidade destas aplicações é importante a ajuda, por um *help* sensível ao contexto, na fase de configuração, mas pela variedade e criticalidade de funções a ajuda na fase de exploração deve ser considerada uma função específica destes produtos. Um bom sistema de mensagens de erro ou aviso, passíveis de serem arquivadas em disco, deve completar a aplicação.

Análises *off-line*

Os dados e as informações tratadas pela aplicação devem poder ser exportados, ou seja, escritos em formatos organizados e compatíveis com folhas de cálculo, bases de dados ou *packages* de análise estatística.

Estas funções devem estar implementadas numa estrutura modular que facilite desenvolvimentos futuros e permita que a aplicação seja utilizada ou vendida

parcialmente, de acordo com as necessidades. Uma arquitectura de *software* aberta deverá permitir a integração de programas do utilizador satisfazendo especificidades e compatibilidades.

Dada a variedade de características e funções dos produtos tipo SCADA será importante caracterizar os produtos em função da sua estratégia de aplicação e de mercado. De facto, e embora possa haver produtos que pretendam cobrir todas as funções especificadas, é natural que outros produtos optem por ser especialistas em determinadas funções.

1.3 — Sistemas de Tempo Real (STR)

Nos sistemas com funções em tempo real, a correcção depende não só do resultado lógico da computação envolvida mas também do tempo em que esses resultados são produzidos. O *software* interactivo de tempo real em funções de gestão de processo — monitorização, controlo e supervisão — podem ser consideradas aplicações críticas ao poderem ter graves consequências económicas, humanas e ecológicas, por deficiente resposta a requisitos como segurança humana, segurança de recursos importantes, privacidade, integridade, tolerância a falhas, tempos de resposta ou disponibilidade.

A boa operação da solução informática, para além do sistema de *software* e do computador de suporte, depende ainda de equipamento adicional, de condições envolventes, de comunicações e da intervenção humana, o que pode potenciar deficiências do sistema.

Estes sistemas tendem a ser cada vez maiores e mais complexos para operação em ambientes dinâmicos e distribuídos, incorporando inteligência e capacidade de adaptação e integrando os desenvolvimentos em *hardware*.

As características de tempo real devem ser satisfeitas ao nível dos vários constituintes da solução informática: sistema operativo, linguagem de programação, base de dados, comunicações e arquitectura.

Sistema Operativo

O Sistema Operativo para uma aplicação de Tempo Real (SOTR) terá que suportar características específicas de função e de performance, integrando a dimensão tempo com uma característica central.

Deve oferecer o suporte básico para garantir restrições de tempo real, suportar tolerância a falhas e processamento distribuído, integrando uma alocação temporizada de recursos como CPU, comunicações, memória e os vários tipos de I/O.

A concepção de um tal sistema operativo deverá ter particular atenção aos seguintes aspectos:

- manuseamento de interrupções, que deverá oferecer respostas determináveis e de acordo com prioridades;
- gestão de memória, que deverá permitir retenção de múltiplos programas em memória (*memory locking*), para diminuir tempos de espera em mudanças de contexto;
- necessidade de processamento concorrente, em que as diferentes tarefas podem ter de obedecer a rigorosas restrições de tempos e a interdependências;
- necessidade de intercomunicação e de sincronização inter-tarefas.

Os componentes do sistema deverão então ser de código reentrante para ser partilhado por diferentes tarefas, e compacto para uma execução rápida e eventualmente poder residir em RAM ou ROM.

Deverão estar previstas facilidades para recuperação de erros ou falhas.

Scheduler

O comportamento dinâmico das aplicações exige um *scheduler* com alocação temporizada de recursos, multi-tarefa, oferecendo tolerância a falhas e com capacidade de adaptação *on-line* à evolução das restrições no tempo de cada tarefa individual. Cada uma poderá ser calendarizada para execução várias vezes, de forma periódica ou assíncrona.

Um *scheduler* configurável pode permitir uma maior adequação do sistema operativo às características específicas de cada processo.

Seleccção de SOTR

A especificação ou a selecção de um SOTR para uma dada aplicação deverá envolver uma análise quantitativa que passa pela avaliação do tempo para mudança de contexto — tempo necessário para o sistema operativo guardar o estado do sistema e o conteúdo dos registos — e o tempo de latência, tempo entre o pedido e o início da mudança de contexto para servir um interrupt, causado por processos não-reentrantes ou críticos que tem de ser terminados.

Pode-se assim determinar o pior tempo de resposta do sistema e comparar com as exigências da aplicação.

Tipos de SOTR

Conforme as características e exigências da solução em que o SOTR se integre deverá ser identificado um tipo de sistema operativo adequado, basicamente entre sistemas dedicados a funções de tempo real e sistemas de uso geral, melhorados para suportar estas aplicações.

Sistemas de uso geral

Algumas formas de implementar o suporte para aplicações de tempo real em sistemas operativos como MSDOS, UNIX, etc., são a seguir descritos:

— Executivos de tempo real (*Real-Time Executives*). É como o *software* aplicativo executado no sistema operativo. Vai realizar de forma mais rápida e eficiente funções do sistema operativo que afectam o seu comportamento em tempo real.

Exemplo é o VRTX da *Ready Systems*, um executivo de tempo real para ambientes DOS, que pode ser complementado por módulos adicionais para funções especiais de I/O, para comunicação e sincronização em sistemas baseados em multiprocessadores. O sistema VRTX está disponível em módulos de PROM's.

— Executar as funções de tempo real num processador adicional que opera em estrita relação com o processador onde é executado o sistema operativo.

O CONCEPS da Bell Lab. é um SOTR para monitorização e controlo baseado no UNIX, que é executado num computador satélite, sob controlo e em comunicação com um computador central UNIX.

— Modificar e estender o sistema operativo para funções de tempo real. Estão disponíveis diferentes sistemas operativos com suporte para tempo real baseados no UNIX e que podem ser soluções para aplicações de média e grande dimensão mas menos exigentes em tempo de resposta.

O UNIX SystemV4.0 da AT&T ou o Real/IX da Modcomp incorporam diferentes adicionais para tempo real. O RTUX da Energy Systems é um núcleo para tempo real que se instala sobre o UNIX e o substitui em algumas funções.

— Desenvolver um SOTR compatível com o sistema operativo em causa.

O *Lynxos* da Lynx Real-Time Systems, é um sistema operativo de tempo real compatível com o *software* desenvolvido para o UNIX.

Sistemas dedicados

Os SOTR dedicados, concebidos para aplicações em tempo real, implementam estritamente as funções necessárias. São de difícil interacção e oferecem ambientes de desenvolvimento limitados.

Pela sua dimensão mais reduzida podem ser disponíveis em meios não voláteis (ROM, PROM, EPROM, etc.), o que pode melhorar a fiabilidade e performance, e facilita a integração do sistema operativo no projecto de *hardware*.

Esta é uma característica particularmente interessante em sistemas integrados em equipamento industrial (*embedded systems*) não interactivos e com funções de I/O reduzidas, bem assim como em sistemas a colocar em ambientes industriais agressivos: ruído, quebras de tensão, etc. A função do sistema operativo neste tipo de aplicações pode mesmo ser substituída por uma linguagem de programação que forneça o suporte para execução em tempo real. É uma solução específica para uma dada aplicação que pode ser mais rápida e ocupa menos memória.

Um exemplo de SOTR dedicado é o IRMX-86, um sistema operativo para tempo real da Intel Corp. disponível em EPROM.

UNIX — Tempo Real

O UNIX poderá vir a ter especificações normalizadas para funções de tempo real no seguimento do trabalho do grupo IEEE 1003, que procura definir o que pode vir a ser um standard ANSI.

Um UNIX tempo real poderá, de qualquer forma, oferecer características interessantes nas aplicações tradicionais do sistema. A utilização de ambientes baseados no UNIX para aplicações de tempo real é potenciada pelos grandes desenvolvimentos em *hardware* e arquitecturas disponíveis para estas aplicações. Outros desenvolvimentos previsíveis para o UNIX, como suporte a processamento distribuído e a modularização do sistema, irão também contribuir para a sua utilização em STR. O sistema operativo modular deverá permitir carregar em memória os módulos estritamente necessários para uma dada aplicação.

Em arquitecturas baseadas em múltiplos processadores, as funções do sistema (I/O, Gráficos, ...) podem estar distribuídas libertando o processador central para executar um sistema operativo de tempo real com uma maior capacidade de resposta.

Linguagens de Programação

As características dos sistemas de tempo real podem justificar a utilização de linguagens de programação específicas que facilitem o desenvolvimento de aplicações eficientes e muito fiáveis, que podem ter que responder a requisitos complexos com restrições fortes em tempo, correcção e segurança.

Uma linguagem de programação deveria então suportar, directamente e em alto nível de abstracção, os conceitos envolvidos no processamento de tempo real.

Algumas características para uma linguagem de tempo real:

- Multi-tarefa, para um atendimento de eventos assíncronos simultaneamente. Suporte à implementação de prioridades.
 - Comunicação e sincronização de tarefas por implementações de *semáforos*, *mailboxes* e *mensagens*, com comandos para activações/desactivações de tarefas.
 - Construções para uma programação mais fiável. Desde estruturas para um desenvolvimento modular e hierárquico até construções que podem restringir o estilo de programação, como não permitir manipular endereços de variáveis ou definir *arrays* dinâmicos.
 - Construções para uma programação tolerante a falhas. Identificação ou previsão de situações de excepção e seu tratamento. Um exemplo será a existência de *handlers* de situações de excepção pré-definidas ou definidas pelo utilizador.
 - Suporte para controlo dos tempos de execução. Restrições de tempo expressas nas construções da linguagem e disponibilidade de primitivas de controlo da utilização de recursos pelos módulos do *software*. Estas características irão permitir uma análise, na compilação, do comportamento dinâmico da aplicação.
- Exemplo são as restrições à alocação dinâmica de memória para dados ou os ciclos lógicos limitados no tempo.
- Reutilização de módulos de *software* de tempo real.
 - Suporte a processamento distribuído. A previsão de tempos de execução será, neste contexto, bem mais complicada.

Extensões à linguagem

A linguagem de programação poderá estar acompanhada por utilitários de desenvolvimento e por extensões específicas para aplicações de tempo real, que podem incluir um analisador do código compilado que avalie a capacidade da aplicação responder em tempo real às tarefas calendarizadas, um simulador da execução em tempo real para avaliação do comportamento dinâmico da aplicação ou um suporte para garantir, na execução, a calendarização de tarefas definida.

Algumas linguagens

São diversas as abordagens que levaram à produção de linguagens de programação caracterizadas por oferecerem, em alguma extensão, facilidades para o desenvolvimento de aplicações para tempo real.

Algumas — como MPP, Concurrent Pascal ou Concurrent Euclid — são desenvolvimentos para processamento multi-tarefa de linguagens de uso geral.

Outras — como Pearl, Modula 2, Edison ou mesmo ADA —, reconhecidas como linguagens de tempo real, são também um compromisso para uso generalizado e muitas construções para tempo real não estão directamente implementadas, antes têm um desenvolvimento facilitado.

De qualquer forma, ADA do U.S. Dept. Defense, impõe-se por uma utilização generalizada em diferentes áreas de aplicação com forte componente de tempo real.

Outras linguagem orientadas de forma restrictiva para aplicações de tempo real são HAL/S, Real-Time Lucid e mais ainda RT-Euclid.

Um exemplo específico

Real-Time Euclid é uma linguagem especialmente concebida para aplicações com restrições críticas no tempo que garante os desempenhos especificados e propicia um desenvolvimento fiável.

Para além de permitir uma definição explícita das restrições no tempo, oferece utilitários para a análise e determinação da resposta às tarefas calendarizadas para os tempos objectivos.

A filosofia do compilador é de encontrar o maior número possível de estrangulamentos, e conhecer todos os tempos que vão ser necessários na execução.

Todas as construções da linguagem são limitadas no tempo — não são permitidas alocações dinâmicas de dados, uso de apontadores e chamadas recursivas a procedimentos, os ciclos têm tempos máximos para execução, as conversões de tipos de variáveis são explícitas e limitadas.

RT-Euclid implementa directamente multiprocessamento de tarefas, comunicações entre tarefas e tratamento hierárquico de excepções, entre outras características.

Ao restringir o estilo de programação e garantir determinado calendário de execução de tarefas, RT-Euclid propicia o desenvolvimento de *software* para aplicações de tempo real, fiável e conforme o especificado.

Que linguagem usar?

A selecção de uma linguagem para desenvolver uma aplicação poderá, no entanto, obedecer a critérios mais pragmáticos em desenvolvimentos pontuais ou em aplicações de pequena dimensão ou não críticas.

Assim, a linguagem melhor conhecida pelos programadores ou a melhor suportada por utilitários de desenvolvimento, poderão ser boas escolhas.

Esforços de desenvolvimento propício a erros de difícil detecção, como codificação em Assembly, programação de interrupts ou escrita de *device drivers* deverão ser, sempre que possível, uma produção automática por utilitários robustos e fiáveis.

Base de Dados Distribuída

Uma aplicação de tempo real, sendo um sistema de processamento de dados, pode requerer um sistema de gestão de bases de dados (SGBD) para armazenamento da informação.

As bases de dados distribuídas são particularmente interessantes nestes sistemas que envolvem processamento paralelo dos dados, com requisitos de velocidade e volume que podem ser críticos.

O controlo do acesso concorrente à base de dados, deverá estar sincronizado com o algoritmo de calendarização de tarefas que impõe objectivos para os tempos de acesso a dados.

Com a informação distribuída, cada tarefa individual pode trabalhar os seus dados com menos estrangulamentos que um sistema centralizado. Uma falha numa parte do sistema

pode ter aqui consequências limitadas, uma característica importante em sistemas que podem ser críticos.

Naturalmente que o uso de SGBD distribuídos implica compromissos, já que os tempos para actualizar e replicar as várias bases afectam o tempo de resposta.

Comunicações

As redes de comunicações são o suporte dos sistemas distribuídos de tempo real. Além da correcção lógica, as comunicações têm que satisfazer as restrições de tempo, impostas pelos processos das aplicações e pelas funções do sistema operativo, para implementar sistemas estáveis e previsíveis.

O sistema de comunicações para STR deve então integrar:

- Encaminhamento (*Routing*) dinâmico em tempos determinados.
- Sincronização de processos em tempo real de acordo com a calendarização executada nos processadores.
- Mecanismos para tolerância a falhas.

O desenvolvimento de comunicações com características para aplicações em tempo real permite utilizações mais abrangentes, integrando processamento distribuído em larga escala.

Arquitectura

A utilização de arquitecturas baseadas em múltiplos processadores oferecem grandes capacidades de computação que permitem alargar a dimensão e a complexidade dos sistemas de tempo real.

As funções do sistema (I/O, gráficos, etc.) podem estar distribuídas, libertando um processador para as funções de tempo real com uma maior capacidade de resposta.

No entanto, uma maior disponibilidade de cálculo pode não ser condição suficiente para satisfazer as restrições de tempo real, já que o processamento associado à distribuição de tarefas e às necessidades de comunicações aumentam de forma significativa com a complexidade da arquitectura.

Por outro lado, a disponibilidade de mais recursos de computação para as soluções de tempo real promove solicitações de uma maior funcionalidade para as aplicações.

As arquitecturas dos STR podem oferecer topologias para interligação de processadores e I/O, para tolerância a falhas e para comunicações mais rápidas e fiáveis, bem assim como para suporte de funções do sistema operativo e de construções das linguagens de programação.

Arquitecturas paralelas para sistemas gráficos

As aplicações interactivas com funções de tempo real na área da gestão de processos, procuram oferecer cada vez mais interfaces gráficas com o utilizador, para uma melhor interacção e compreensão dos múltiplos aspectos relevantes e da sua evolução no tempo. A informação é comunicada ao utilizador por meio de símbolos, gráficos, diagramas e sinópticos animados em tempo real, o que acrescenta um grande esforço de computação gráfica ao STR.

As arquitecturas paralelas vêm permitindo soluções para computação gráfica baseadas em multiprocessadores de baixo custo, e grande capacidade de actualização de écrans, recorrendo a uma divisão das áreas gráficas por diferentes processadores.

1.4 — O Desenvolvimento de Software

A produção de *software* de tempo real (STR), para funções de gestão de processos pode, dependendo da dimensão e área de aplicação, envolver um desenvolvimento exigente em suporte de tecnologias de *software* muitas vezes distintas de outros tipos de produções, com metodologias, princípios, teorias e utilitários específicos.

As características de tempo real devem ser satisfeitas ao nível das várias fases do desenvolvimento do *software* de aplicação: análise, concepção, verificação, codificação, teste e manutenção.

Este é o modelo tradicional para o desenvolvimento de *software* mais orientado para suportar a gestão do processo e que, na realidade da produção, é alterado pela necessidade de esforços iterativos ou pela incerteza nos requisitos, o que vai provocando alterações ao longo de todo o processo, sobrepondo as várias fases definidas.

Na produção de *software* de tempo real, a complexidade dos requisitos e a dificuldade em os validar promovem, em particular, um desenvolvimento não sequencial, dos requisitos e concepção à validação e codificação. Antes, aconselha um desenvolvimento incremental

dos módulos que, incompletos, são refinados por sucessivas decomposições ou composições numa estratégia de desenvolvimento, tanto *top-down* como também *bottom-up*.

De facto, a concepção e verificação em STR, não pode ignorar detalhes da implementação já que as restrições de tempo, em análise, dependem do ambiente de execução e da implementação.

Procura-se assim um modelo tecnicamente mais apropriado à natureza do processo de desenvolvimento.

Análise de Requisitos

A especificação de requisitos descreve as características a que deve obedecer a solução de *software* para satisfazer os seus utilizadores. Define também as restrições do sistema pretendido.

Uma especificação deverá então permitir obter um entendimento da solução através do processo iterativo de análise do problema, documentação dos requisitos encontrados e validação do modelo conseguido.

DFD

Os Diagramas de Fluxo de Dados, DFD, constituem um processo de decomposição de sistemas aplicado na fase de análise. É uma metodologia *top-down* que conduz à implementação de um processo como uma rede de processos mais elementares.

O modelo, uma representação lógica do processo, é assim progressivamente construído (síntese) e validado (análise), para ser completo e consistente com as características esperadas para o sistema.

A metodologia é usualmente aplicada com base em regras informais, heurísticas, experiência e conhecimento do problema e dos requisitos do sistema.

DFD para STR

A extensão dos modelos baseados em DFD para representar sistemas de tempo real, deverá incluir formas de representar os fluxos e as transformações de controlo no sistema, as restrições no tempo para as transformações, os fluxos discretos e contínuos de dados, as múltiplas instanciações da mesma transformação, os estados do sistema e os mecanismos de transição entre estados.

Utilitários

A formalização do processo de decomposição como uma técnica rigorosa com suporte algébrico, permite oferecer uma decomposição automática de DFD por utilitários para análise de sistemas, que podem incluir também uma validação automática dos níveis de DFD construídos.

Um protótipo para decomposição automática de DFD é apresentado por Adler [24]. O formalismo representa o fluxo de dados e prevê extensões para representação do fluxo de controlo e de transformações em tempo real com restrições de tempo, conforme Ward [29].

A integração deste suporte em ambientes gráficos de edição e descrição de DFD, constituirá um utilitário produtivo na engenharia de *software*.

Concepção

A concepção de *software* implica a síntese de representações da estrutura de dados, da estrutura do programa e dos procedimentos do sistema, a partir das especificações dos requisitos.

A concepção pode ser baseada no fluxo de informação com a representação da estrutura do *software* obtida a partir dos DFD produzido na análise. Esta abordagem é adequada para aplicações em áreas técnicas e científicas, como controlo de processos ou análise de dados.

As características de tempo real podem ser tratadas por extensões a esta abordagem, como DARTS [50].

A representação formal do STR deverá incluir uma métrica de tempo.

A concepção de STR's não pode ignorar detalhes da implementação, já que as restrições de tempo dependem do ambiente de execução e da implementação.

Interacção Análise — Concepção

A interacção entre as fases de análise e concepção é formalizada na metodologia proposta por Ward [29], que prevê:

- 1 — Construção do DFD de nível 1 com fluxo de dados e controlo.
- 2 — Construir uma lista das acções principais que ocorrem no sistema. Útil para refinar o DFD 01 e para criar os diagramas de estado do sistema.

- 3 — Detalhar o DFD para os níveis 2 e 3.
- 4 — Construir os diagramas de estado para o sistema.
- 5 — Rever os DFD's e os diagramas de estado para assegurar a consistência de funções e fluxos no sistema.
- 6 — Descrever os dados e os procedimentos das funções do sistema. Usar linguagem estruturada, PDL, árvores ou tabelas de decisão.
- 7 — Obter o diagrama de estrutura para o STR por análise de transformação ou transacção.
- 8 — Rever as várias representações e iterar.

Documentação

Uma metodologia standard para documentação da fase de concepção de *software* é proposta pela IEEE — projecto 1016 [28].

O sistema de *software* é representado por um modelo da informação necessária pelos vários intervenientes no desenvolvimento do *software*.

O modelo é constituído por um conjunto de 15 atributos que caracterizam as propriedades e relações de cada entidade resultante da decomposição do sistema. Cada entidade é distinta em estrutura e função.

A documentação facilita a análise, o planeamento, a gestão e particularmente a codificação e a manutenção do sistema de *software*. Permite identificar os recursos necessários ao sistema, avaliar o impacto de alterações aos requisitos, verificar a satisfação de requisitos e especificar testes ao *software*.

Atributos das Entidades da Concepção de *Software*:

- 1 — Identificação: o nome da entidade.
- 2 — Projectistas: descrição do criador da entidade. Nome, organização e contacto.
- 3 — Origem: origem do *software*. Fornecido, adaptado ou desenvolvido internamente.
- 4 — Tipo: descrição da natureza da entidade. Subsistema, módulo, procedimento, ficheiro.
- 5 — Objectivo: razão da existência da entidade. Requisitos específicos para a entidade ou processo de raciocínio que a originou.
- 6 — Funções: transformações implementadas pela entidade.
- 7 — Subordinados: identificação das entidades usadas por esta.

8 — Dependência: descrição de relações desta entidade com outras. Interações, associação ou duplicação de entidades. Interface por parâmetros.

9 — Recursos: utilização de unidades externas pela entidade e processamento relacionado.

10 — Interfaces: descrição de como as outras entidades interactivam com esta entidade. Métodos e regras. Mecanismos de chamada e de partilha de dados.

11 — Processamento: sequência das operações realizadas pela entidade. Algoritmos, inicializações, excepções.

12 — Dados: dados internos à entidade.

13 — Compromissos: alternativas consideradas e razões de escolha.

14 — Assumpções: premissas do projecto não explicitamente definidas nos requisitos.

15 — Referências: fontes utilizadas na formação da entidade.

Verificação

A verificação deverá ser formal, baseada em procedimentos que avaliem de forma sistemática o comportamento dinâmico do sistema. Trata-se de verificar as especificações do sistema que, incluindo uma métrica no tempo, caracterizam as restrições de tempos do STR, provocadas de facto pela implementação e pelo ambiente de execução.

CAPÍTULO 2

Projecto de Software

2.1 Especificação de Requisitos

A análise de requisitos da aplicação SProcess permite estabelecer as características do sistema de hardware e de software que condicionam o seu desenvolvimento, especificação de arquitectura, e as funções que esta deve desempenhar, especificação funcional.

2.1.1 Especificação de Arquitectura do Sistema

A especificação de arquitectura do sistema, incluindo as componentes de Hardware e Software foram pré-definições ao projecto do software em causa.

- Hardware

A aplicação SProcess deverá ser desenvolvida para execução num microcomputador, tipo IBM-AT compatível, com 640 KB de RAM, écran gráfico EGA, disco rígido, portas série e paralela e *slots* de expansão.

Embora o sistema de interface com o processo (aquisição e actuação) deva poder ser um qualquer, começará por ser assegurado o funcionamento com o sistema KEITHLEY 500, que inclui uma interface para entradas de dados analógicos e uma placa para condicionamento de sinal e assegurada comunicação com o controlador Armfield através da porta série.

A figura 2-1 representa esquematicamente o sistema de hardware especificado.

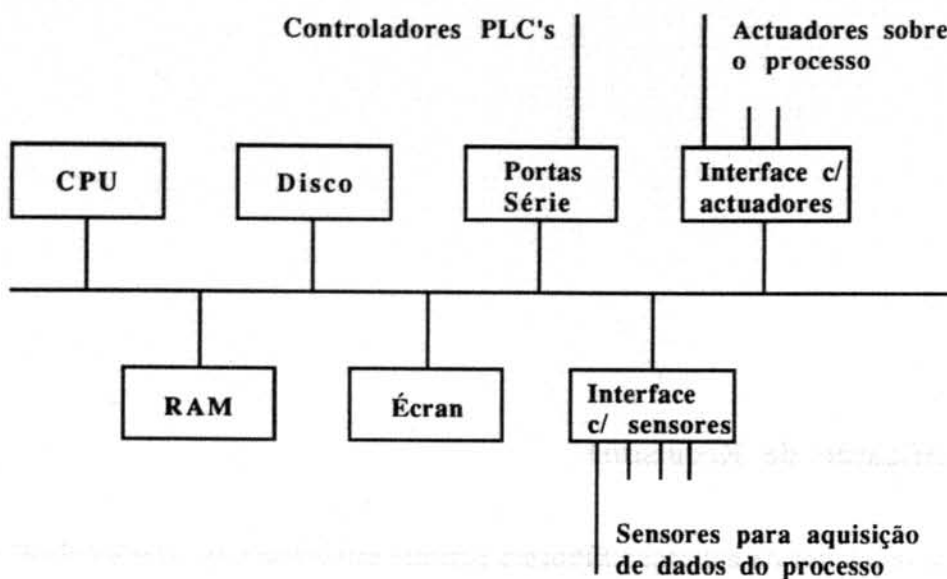


Figura 2-1: Diagrama de Blocos do Sistema

• Software

O sistema operativo para desenvolvimento e utilização da aplicação será o MS-DOS e a linguagem de programação o C.

Trata-se de um ambiente unitarefa e monoposto, quando a aplicação SProcess deverá implementar várias funções com resposta em tempo real e uma operação interactiva.

É assim necessário conceber e desenvolver módulos da aplicação com um funcionamento em paralelo, em background e, eventualmente, temporizáveis.

A aplicação deve poder utilizar módulos de software pré-existentes, nomeadamente, o editor gráfico e o gerador de gráficos multilinhas, pelo que deverão ser desenvolvidas interfaces independentes com estes sub-programas, que permitam a sua substituição.

A concepção da aplicação deverá reflectir preocupações de modularidade e independência necessárias a uma aplicação que se pretende configurável a diferente hardware de aquisição e actuação, e a diferentes processos industriais ou laboratoriais. A aplicação deverá ser facilmente tornada um sistema aberto a desenvolvimentos e expansões.

Serão também separados os sub-programas relacionados com a configuração da aplicação pelo utilizador, que produzirão ficheiros de configurações e os sub-programas relacionados com a exploração em tempo real, que começarão por conhecer esses ficheiros.

2.1.2 Especificação Funcional

Pretende-se que a aplicação, em fase de exploração em tempo real, implemente as seguintes funções:

- **Aquisição de dados**

Deverá poder ser executada uma leitura periódica a variáveis do processo que correspondam a canais de um sistema de aquisição de dados. Cada variável deve poder ser caracterizada pela sua frequência de leitura e acondicionamento de sinal.

O sistema de aquisição a utilizar deverá assegurar a comunicação com os sensores no processo.

Os dados adquiridos deverão ser guardados temporariamente em memória para processamento.

Esta função deverá ser executada em background de forma periódica.

- **Arquivo de dados**

Os dados adquiridos do processo e guardados temporariamente em memória irão sendo arquivados em ficheiros em disco, para posterior utilização. Esta função deve ser executada em background, em paralelo com a aquisição de dados.

- **Visualização de quadros sinópticos**

Durante a operação on-line deverão poder ser visualizáveis écrans que representem quadros sinópticos do processo e que possam incluir uma actualização numérica, gráfica e animada das variáveis do processo associadas.

É uma função que uma vez seleccionada será executada em foreground (no écran) e em paralelo com as funções de background.

- **Análise de dados**

Deve ser oferecida a possibilidade de serem efectuados cálculos sobre os dados adquiridos e dos resultados poderem ser visualizados numa forma gráfica ou tabelar. Só serão usados os dados mais recentes, ainda em memória.

Deve ser possível alterar as configurações tanto dos cálculos como dos gráficos durante a exploração de tempo real.

É uma função que uma vez seleccionada será executada em foreground (no écran) e em paralelo com as funções de background.

- Tratamento de alarmes

Devem ser produzidos avisos ao operador, alarmes, quando os valores das variáveis do sistema excedam dois limites configurados, um inferior e um outro superior. As mensagens de alarme devem ser visíveis em todos os écrans de operação e constar de um número de ordem, da variável, do tipo de alarme e da hora. As mensagens devem ser guardadas numa fila de pendentes, de onde poderão ser retiradas após validação individual. A identificação e visualização de alarmes deve ser executada em paralelo com aquisição de dados e com prioridade sobre as outras funções. A validação é uma função interactiva.

- Operação

Deverá ser oferecido um ambiente aonde os valores das variáveis configuradas como de saída, possam ser alteradas para operação manual do processo.

Deverá ser possível actuar sobre controladores locais, visualizando e alterando os valores dos seus registos.

É uma função que uma vez seleccionada poderá ser executada em foreground (no écran) e em paralelo com as funções de background.

- Segurança

Deverão existir dois níveis de protecção por palavra-chave. Um nível de supervisor do processo, cuja palavra-chave permitirá usar toda a aplicação, e um nível de operador com a utilização limitada à exploração em tempo real.

Estas funções implementadas na fase de exploração em tempo real, pressupõem uma configuração prévia ao sistema de hardware e software e ao processo em causa.

A configuração destas funções deve poder ser feita em um ou vários programas independentes.

Para a produção dos quadros sinóticos deverá ser usado o editor gráfico " Editor ".

A coerência das configurações efectuadas deve poder ser validada, de forma automática, antes de se iniciar a exploração em tempo real.

Terminada a exploração em tempo real, deve ser possível converter os dados arquivados em ficheiros para um formato compatível com aplicações, como uma folha de cálculo, aonde podem ser efectuados relatórios com análises e gráficos sobre a operação do processo.

A aplicação no seu conjunto deverá ter uma utilização simples baseada na selecção de opções em menus de barras. Os nomes das opções e comandos disponíveis devem poder ser alterados pelo utilizador permitindo uma melhor adaptação ao processo em causa.

2.2 Análise do Sistema

Pretendeu-se construir um modelo do sistema de software que satisfizesse as especificações de requisitos enunciados.

A análise foi efectuada considerando que o sistema será configurável a diferentes situações reais, pelo que não foram consideradas restrições ao tamanho dos depósitos e buffers de dados, nem mesmo aos tempos de execução dos processos.

Trata-se, então, de construir um modelo de requisitos do sistema com recursos infinitos. Os depósitos têm capacidade ilimitada e os processos um tempo de execução nulo.

Dadas as características de tempo real de algumas das funções previstas, que implicam paralelização de processos, temporização de acções, fluxos contínuos, fluxos de controlo, etc, é feita uma análise tanto no domínio da informação como no do controlo, sendo adoptada, embora parcialmente, uma nomenclatura específica para as representações gráficas usadas, baseada no proposto por Ward [29].

É feita uma análise orientada ao Fluxo de Dados que, sendo uma metodologia de grande espectro de aplicação, com a inclusão de algumas extensões, pode ser utilizada no projecto de aplicações de tempo real, conduzidas por interrupções. Essas extensões são, resumidamente, as seguintes:

- a) um mecanismo de representação de comunicação e sincronização entre tarefas;
- b) uma notação para representar a dependência de estado, como por exemplo Diagramas de Transição de Estados;
- c) uma abordagem que permita a inclusão nos Diagramas de Fluxo de Dados convencionais de particularidades dos sistemas de tempo real, recorrendo a diagramas de fluxos de dados e de controlo, fluxos de dados contínuos, etc.

Num contexto de Análise Estruturada, pretende-se obter, como produto desta fase do Projecto de Software, as representações gráficas do modelo da aplicação para o fluxo de dados e controlo (DFD, DFC e DTE) e as respectivas especificações de dados (Dicionário de Entidades, Processos e Depósitos) e de controlo.

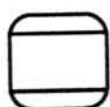
2.2.1 Fluxo de Informação

Apresenta-se o resultado de uma análise do sistema no domínio da informação, representando o processamento e a circulação da informação em Diagramas de Fluxos de Dados — DFD's. São tratados os primeiros três níveis funcionais numa perspectiva de abordagem top-down, ficando definidas as características fundamentais e as interacções entre os processos necessários para implementar as funções pretendidas (figs. 2-2 a 2-6). As entidades, os processos e os depósitos envolvidos nos DFD's são caracterizados no Apêndice B — Dicionário de Dados: Entidades, Processos e Depósitos.

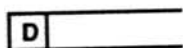
Nomenclatura — DFD's



— Entidade externa ao sistema



— Processo associado ao desempenho de determinada função



— Depósitos de dados



— Fluxo de dados discreto



— Fluxo de dados contínuo

Diagramas de Fluxos de Datos

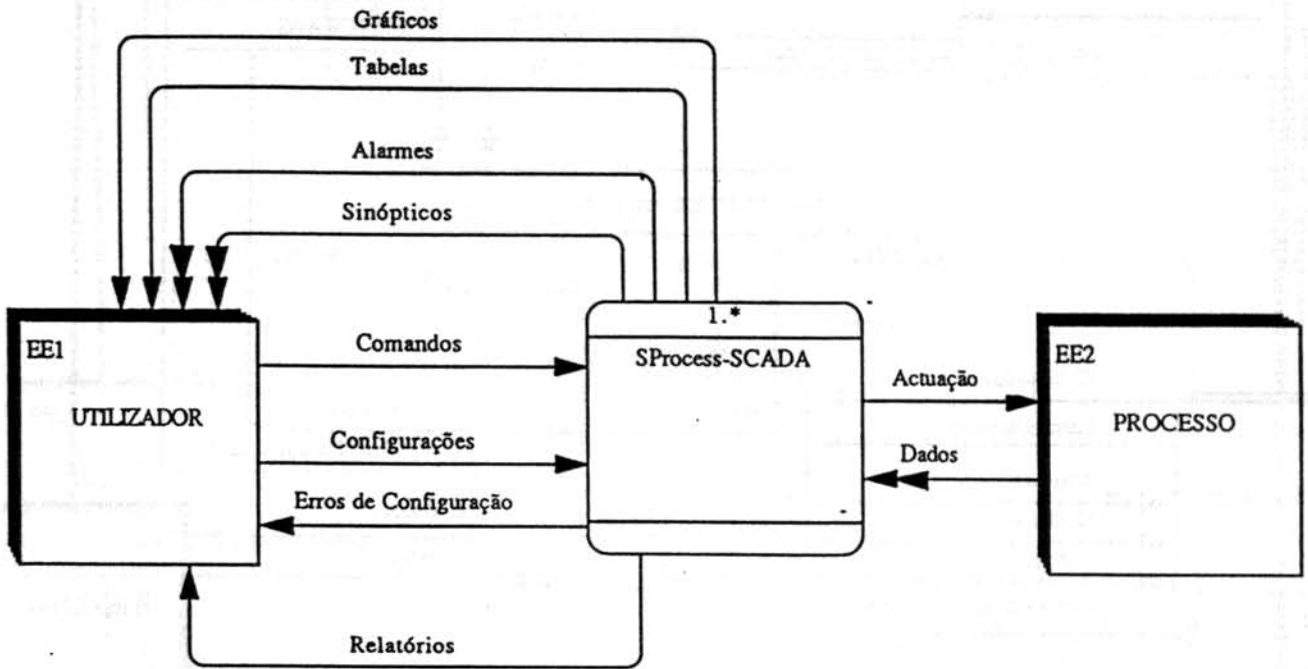


Figura 2-2: DFD de Contexto

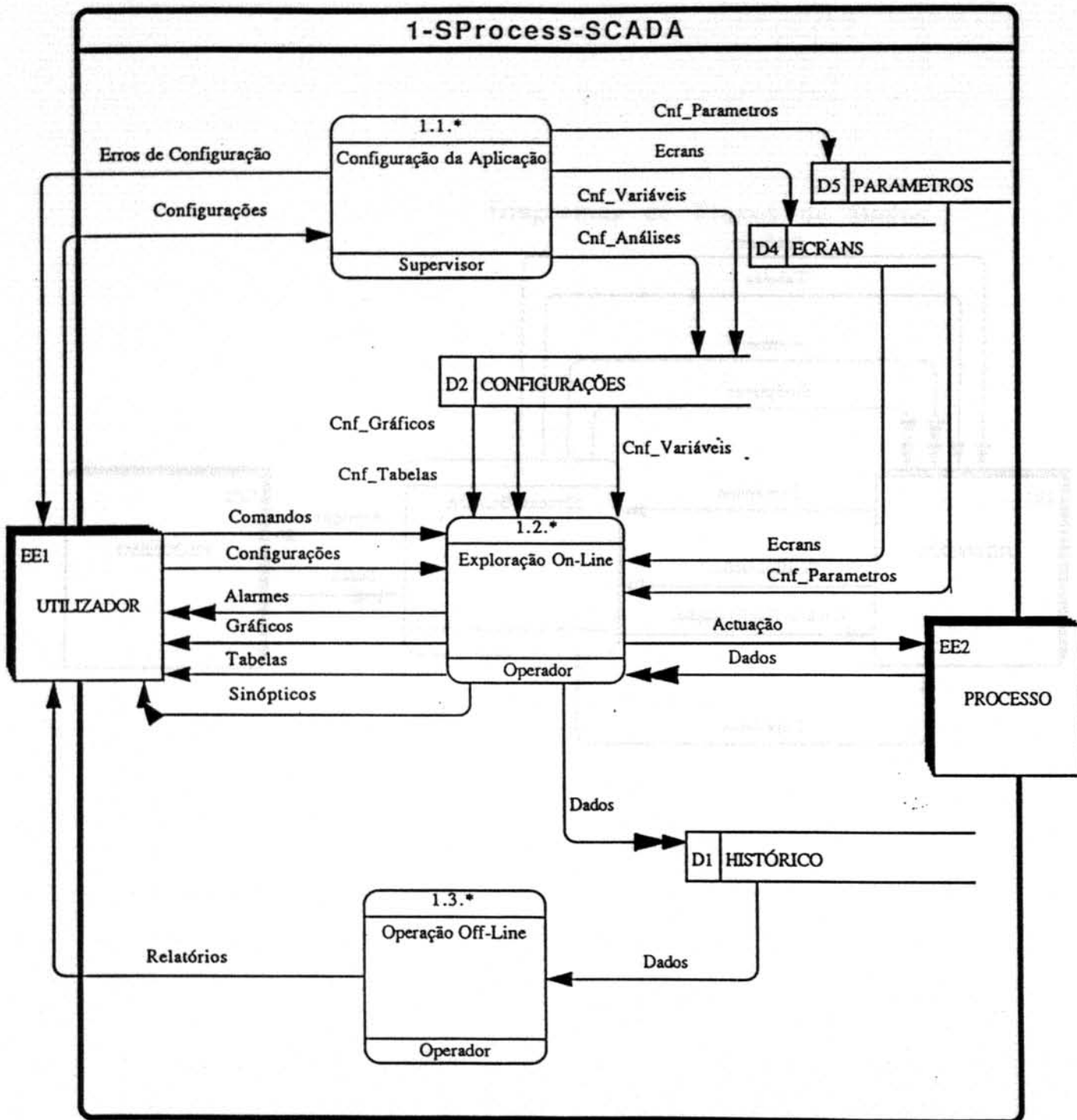


Figura 2-3: DFD de 1ª Ordem - Aplicação

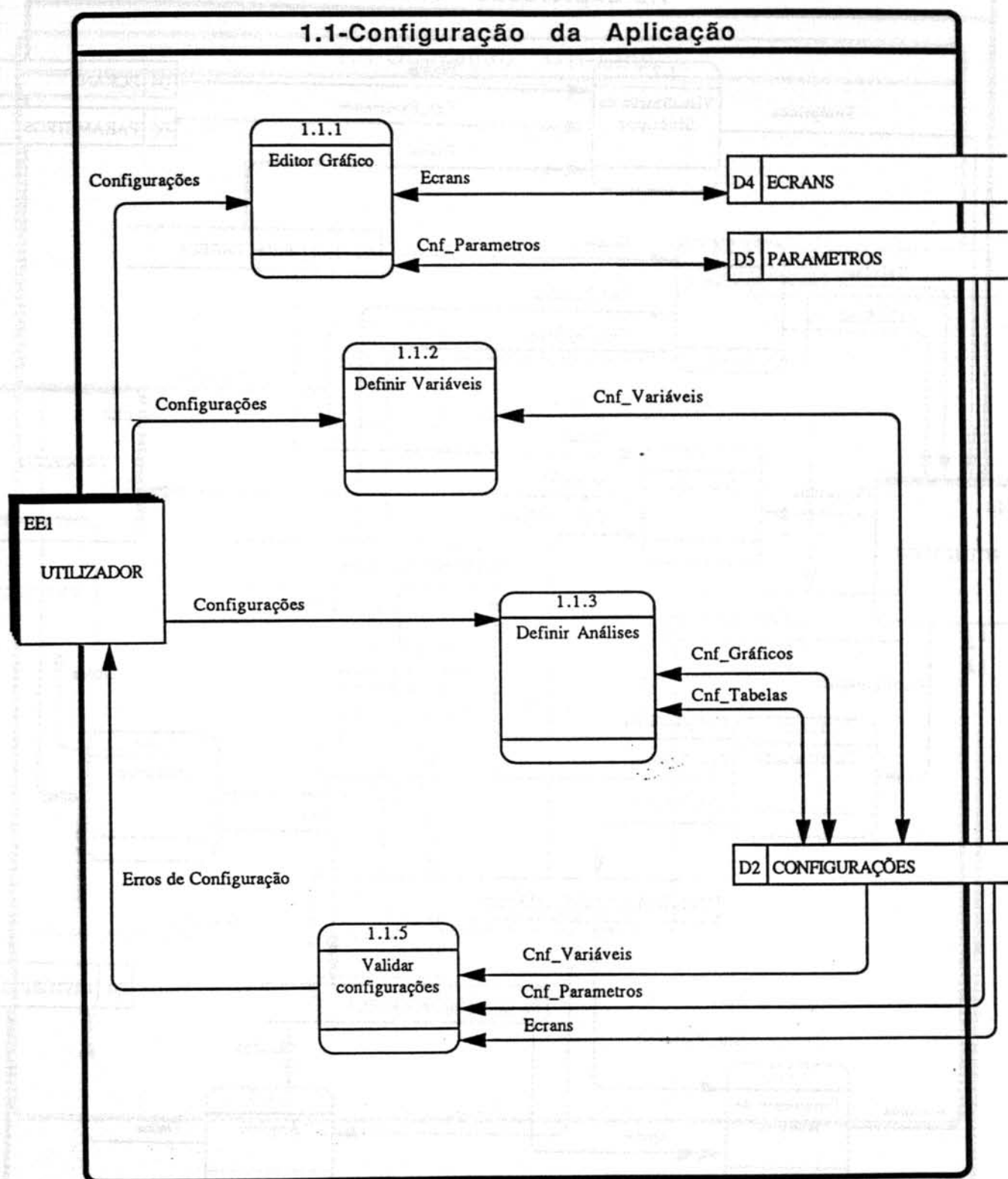


Figura 2-4: DFD de 2ª Ordem - Configuração da Aplicação

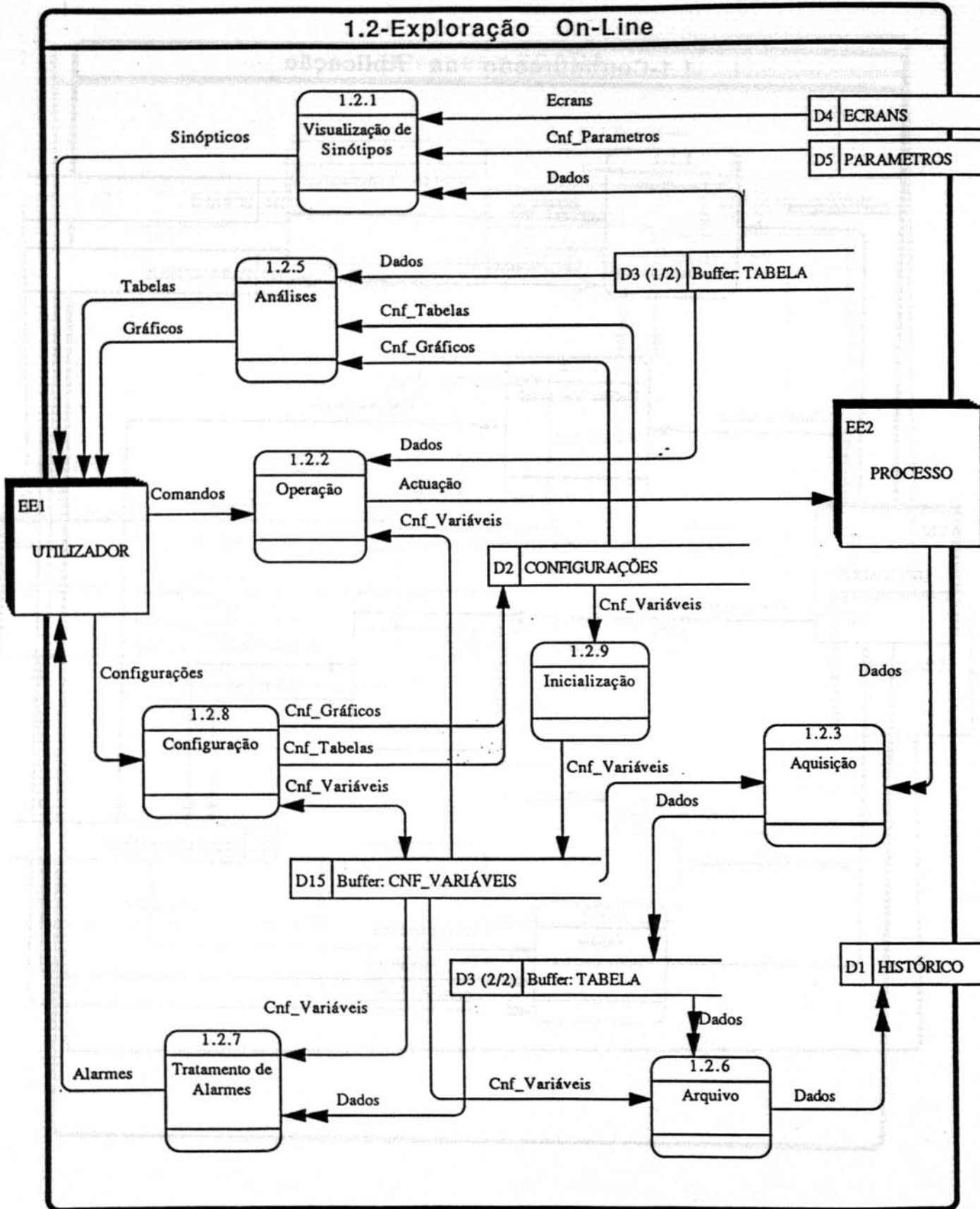


Figura 2-5: DFD de 2ª Ordem - Exploração On-Line

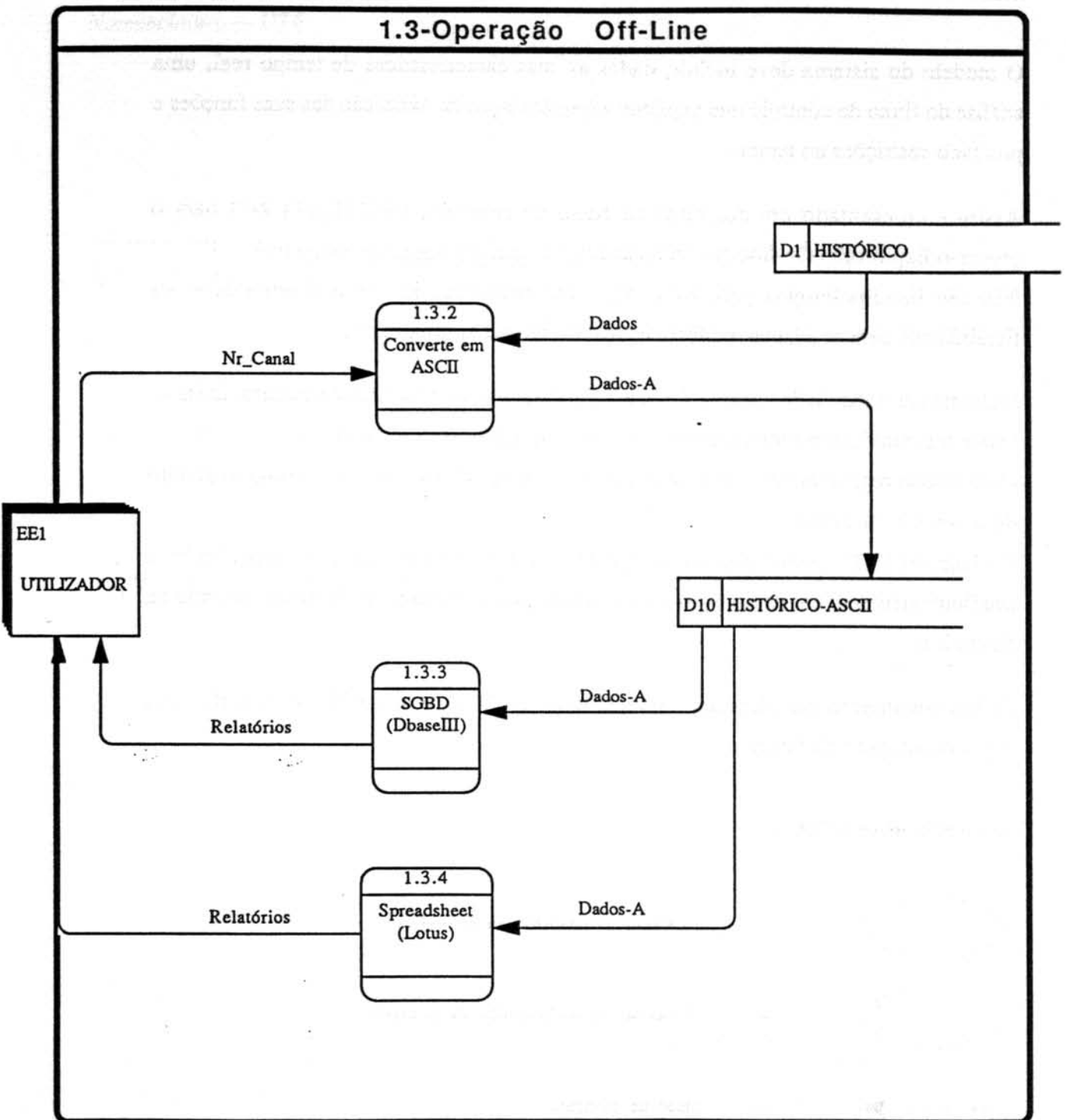


Figura 2-6: DFD de 2ª Ordem - Operação Off-Line

2.2.2 Fluxo de Controlo

O modelo do sistema deve incluir, dadas as suas características de tempo real, uma análise do fluxo de controlo que explicita a sequência da execução das suas funções e possíveis restrições no tempo.

Assim é apresentado um diagrama de fluxo de controlo, DFC (figura 2-7) para o processo Exploração On-line que inclui as funções com resposta em tempo real.

Não são fixados tempos para a execução dos processos já que o sistema deve ter flexibilidade para se adaptar a diferentes realidades, por configuração.

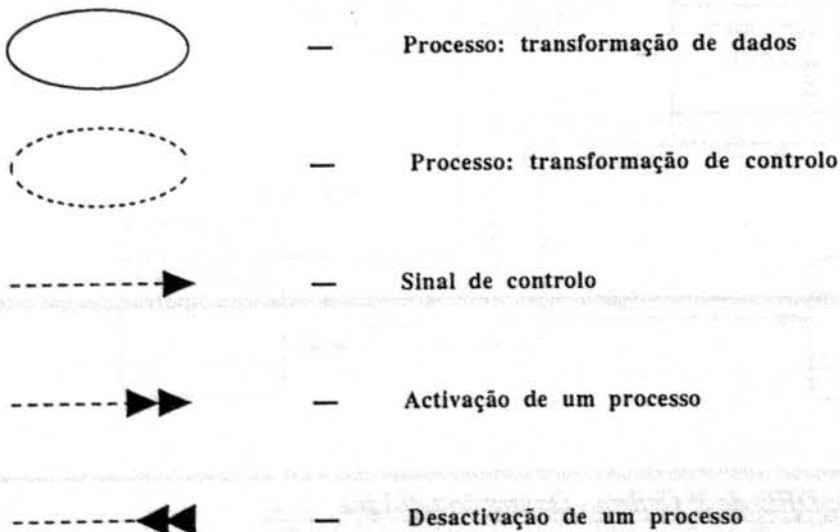
Incluimos no modelo do sistema SProcess um diagrama de transição de estados, DTE — "State Machine", para representação das suas características dinâmicas.

Cada estado representa períodos de tempo durante os quais o padrão de comportamento do sistema é constante.

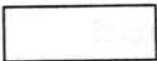
O diagrama DTE apresentado, figura 2-7, modela o sistema SProcess, e em particular o seu funcionamento em exploração on-line, conforme o diagrama de fluxo de controlo da figura 2-6.

O funcionamento do sistema, representado no DFC e no DTE, é descrito nas Especificações de Controlo.

Nomenclatura — DFC's



Nomenclatura — DTE



— Estado do sistema



— Mudança de estado



— A acção de entrada (I) provoca as acções de controlo (O) sobre processos

Diagrama do Fluxo de Controlo

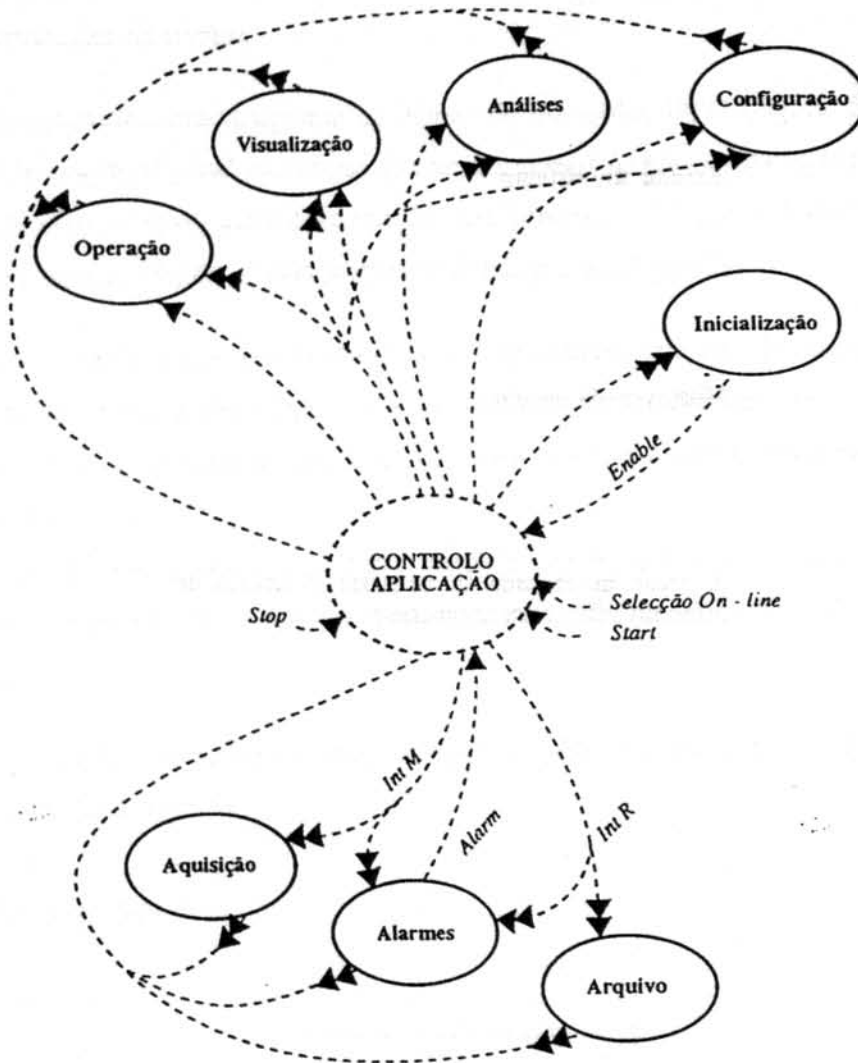
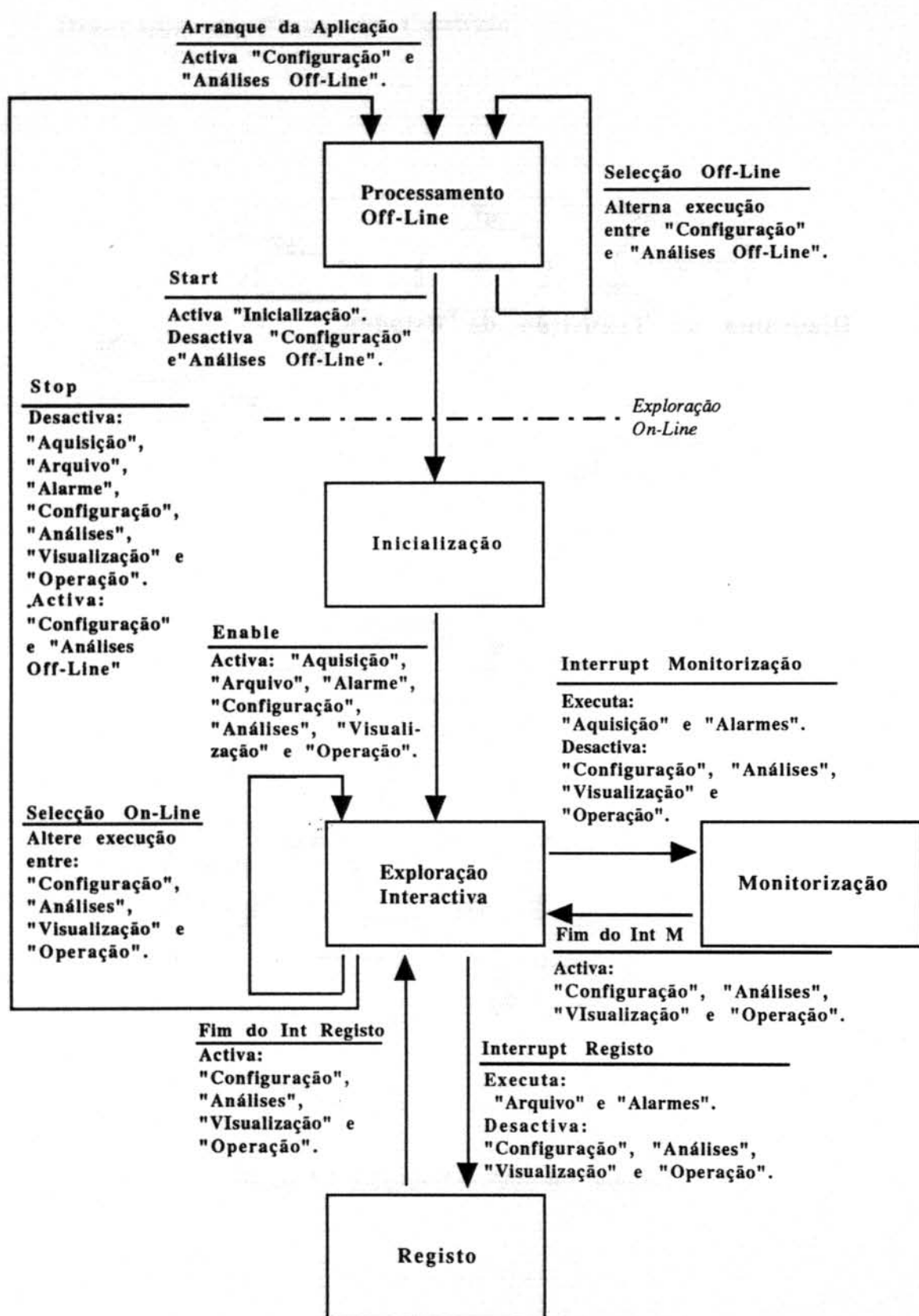


Figura 2-7: Diagrama de Fluxo de Controlo

Diagrama de Transição de Estados



Especificação de Controlo

Descreve-se o modelo dinâmico da Aplicação, funcionamento e interações, conforme as representações gráficas DFC e DTE.

- Arranque da Aplicação

Ao ser executada no DOS, a aplicação começa por oferecer um Processamento Off-Line com dois processos activos.

- Selecção Off-Line

A execução dos processos é seleccionável pelo utilizador, um para Configuração da Aplicação e o outro para Análises Off-Line de dados históricos.

- Start

Para iniciar o Processamento On-Line, o utilizador poderá escolher a opção Start que desactiva os processos off-line e inicia um processo de inicialização que faz uma leitura de configurações em disco e instala as rotinas de serviço às interrupções para aquisição de dados e arquivo em disco. Quando termina emite um sinal de Enable de interrupções.

- Enable

Após o sinal de *Enable* de interrupções, são activados todos os processos da Exploração On-Line: Análises, Visualizar Sinópticos, Operação, Configuração, Aquisição, Arquivo e Alarmes, ficando o sistema no estado de Exploração Interactiva, respondendo a selecções do utilizador.

- Selecções On-Line

Conforme as selecções dos utilizadores, no estado de Exploração Interactiva, são executados os processos: Visualizar Sinópticos, Análises, Operações e Configurações. Visualizar sinópticos mostra no écran quadros sinópticos actualizados com dados do buffer Tabela.

Análise representa gráficos com cálculos ou tabelas de dados com valores do buffer Tabela.

Operação permite produzir actuações sobre o processo.

Configurações permite alterar em tempo real algumas configurações dos vários processos, actualizando os ficheiros Configurações ou o buffer de Configurações de Variáveis.

- Interrupções para Monitorização

Periodicamente o sistema responde a interrupções para aquisição de dados, mudando para um estado de Monitorizações do processo. São desactivados os processos: Visualizar Sinópticos, Análises, Operações e Configurações. São activados os processos de Alarme e de Aquisição. Este assegura um fluxo contínuo de dados que são guardados no buffer Tabela. Os dados são verificados e caso existam situações anormais é emitido um sinal de alarme. Terminando é emitido um sinal de fim da interrupção.

- Fim da interrupção para Monitorização

Com o fim da interrupção o sistema muda para a Exploração Interactiva reactivando os processos: Visualização, Análises, Operações e Configurações, e retomando a sua execução.

- Interrupção para Registo

Periodicamente o sistema responde a interrupções mudando para um estado de registo em que são activados os processos Arquivo e Alarmes.

Havendo dados suficientes no buffer Tabela, estes são arquivados em disco. Havendo sinal de alarme, estes são processados com emissão de mensagens.

No fim da execução é emitido um sinal de fim de interrupção.

- Fim da interrupção para Registo

Com o fim da interrupção o sistema muda para a Exploração Interactiva reactivando os processos: Visualização, Análises, Operações e Configurações, e retomando a sua execução.

- Stop

Com a opção Stop seleccionada o sistema muda do estado de Exploração Interactiva para o de Processamento Off-Line, desactivando os processos de Aquisição, Arquivo, Alarme, Visualização, Análises, Operações e Configurações.

2.3 Concepção do Sistema

O objectivo desta fase do Projecto de Software é produzir, a partir dos resultados da Análise Estruturada efectuada, um modelo físico, para a aplicação em desenvolvimento, e as respectivas especificações para codificação.

A partir das especificações de quais devem ser as funções da aplicação, pretende-se, agora, definir como deve ser construída a aplicação.

Uma representação gráfica do modelo físico, recorrendo a mapas de estrutura, é apresentada na Arquitectura da Aplicação (2.3.1).

As especificações para codificação são apresentadas recorrendo a uma caracterização proposta pela IEEE na norma 1016 [28] e contempla especificações de concepção para os Módulos, os Dados e a Interface com o utilizador.

O modelo do software, conforme o especificado na Análise de Sistemas efectuada, concentra-se sobre as funções principais que se exigem ao sistema. Para a concepção de um modelo físico a implementar, o mapeamento dos DFD's em ME's deve considerar funções adicionais, de interface com o utilizador, de entrada e de saída e de gestão, a aplicar aos vários níveis da especificação da arquitectura da aplicação.

A estrutura tipo, proposta por Hatley e Pirbhai, [49] pode ser representada pelo esquema da figura 2-9 aonde os vários módulos de software se poderão classificar.



Figura 2-9: Estrutura tipo de um processo

O processamento associado à interface com o utilizador deve poder ser individualizado do resto do sistema de software para, por um lado, poder ser testado e validado à parte e, por

outro lado, para estar facilitada a sua actualização ou substituição, já que constitui um problema diferente, com um desenvolvimento bem característico.

As entradas e saídas referem-se a processamento de comunicação com os sistemas físicos envolventes que, podendo ainda não estar definidos, aparecem aqui a requerer tratamento. Pode ser necessário considerar a inclusão de módulos para implementar funções como testes e protecções a erros (ou a falhas) e redundância ou manutenção, para ultrapassar situações anormais.

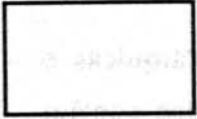
Tendo presente estas orientações, a malha de processos que constitui os vários níveis de DFD's é mapeada ou transformada numa hierarquia de módulos que constitui os vários níveis dos ME's.

São usadas as técnicas da Análise de Transacções que a par da introdução da hierarquia, explicitam transferências assíncronas e síncronas na representação gráfica do sistema.

Os módulos representados correspondem, basicamente, a módulos de software: funções, rotinas, procedimentos, etc.

Nas especificações para codificação, estes módulos serão classificados de acordo com a estrutura-tipo aqui proposta (figura 2-9).



NOMENCLATURA - MAPA DE ESTRUTURA

- Tarefa / módulo



- Módulos específicos ao hardware



- Módulo de I/O



- Centro de Transações



- Buffer



- Selecção síncrona



- Selecção assíncrona

2.3.1 Arquitectura da Aplicação

O modelo físico do sistema de software é, aqui, representado recorrendo a Mapas de Estrutura (ME).

Os ME's integram, numa mesma representação gráfica, relações hierárquicas e transferências de informação e de controlo, em oposição aos DFD's que contêm exclusivamente fluxos de informação e onde não existe hierarquia. O sistema é constituído por módulos independentes, cada um implementando uma tarefa/processo, caracterizado por um conjunto de entradas e saídas e por uma lei de controlo que impõe as condições em que essa transformação se pode efectuar.

O ME de ordem zero abarca todo o sistema e consiste na representação dos módulos, implementando as tarefas principais, as transferências de informação existentes e os eventos que provocam transições entre tarefas/estados. Pode-se representar a comunicação entre tarefas recorrendo à utilização de "buffers" duplos.

Seguidamente, tal como para os DFD's, mas conservando sempre uma estrutura hierarquizada, cada módulo do ME, de ordem zero, é dividido em sub-módulos, dando origem a ME's de ordem um, e, assim, sucessivamente até que se atinja o desejado nível de pormenor.

Os ME's permitem, por um lado, fazer corresponder facilmente a cada tarefa/módulo um módulo de software, i.e. uma rotina, função, procedimento, etc., e, por outro lado, fornecem a estrutura hierárquica que permite articular a interligação entre módulos e implementar o gestor, caso exista, das transições assíncronas entre tarefas/estados.

A representação da comunicação entre tarefas permite, ainda, a identificação de regiões críticas de acesso mutuamente exclusivo a variáveis ou a estruturas de informação, por exemplo áreas de memória onde mais de uma tarefa podem ler e escrever "simultaneamente", alertando para a necessidade de implementação de mecanismos de controlo de acesso.

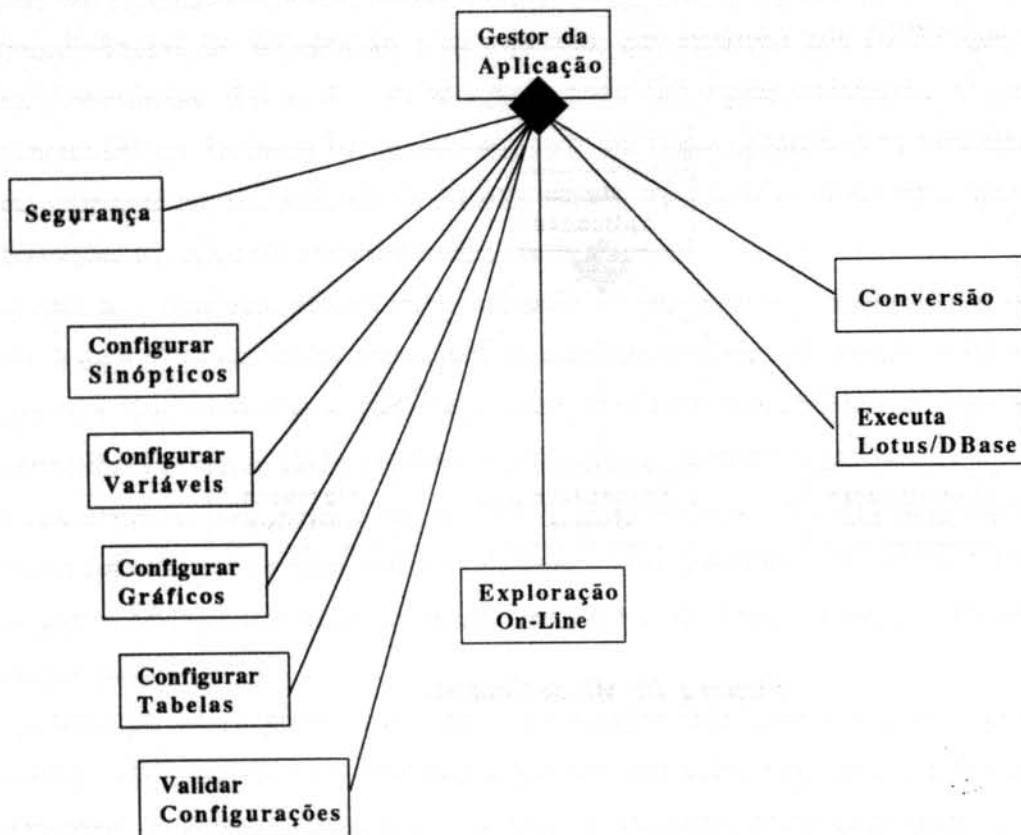
Os ME's aqui apresentados correspondem a uma versão simplificada do formalismo que pode acompanhar estas representações. Pretende-se, fundamentalmente, estabelecer a estrutura de módulos da aplicação, a hierarquia que os liga e destacar a partilha de dados em memória.

Mapas de Estrutura



Figura 2-10: ME de Contexto

O ME de contexto introduz um módulo de Gestão da Aplicação que assegura num primeiro nível a interface com o utilizador e controla a execução das configurações da aplicação, da exploração on-line e da operação off-line.

Módulo 1: Gestor da Aplicação*Figura 2-11: ME de 1ª Ordem - Aplicação*

Com o início da execução da aplicação o módulo Gestor começa por chamar o módulo Segurança que identifica o utilizador e certifica se é autorizado. Em seguida implementa a interface com o utilizador estendida às várias funções da Configuração e Operação Off-line que chama conforme selecção do utilizador. As funções de configuração só estão disponíveis se o utilizador for supervisor.

Módulo 2: Gestor On-Line

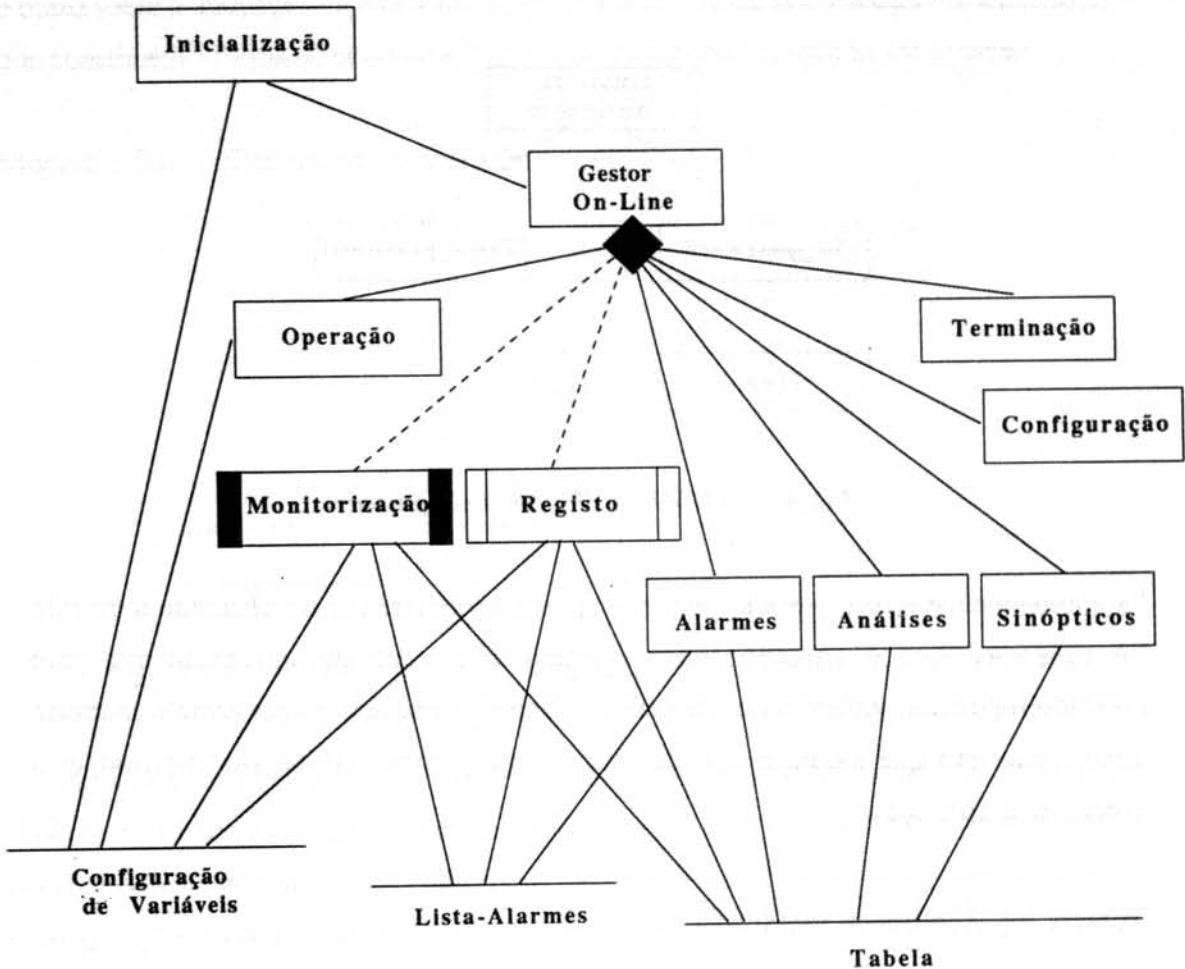


Figura 2-12: ME de 1ª Ordem - Exploração On-Line

Selecionada a Exploração começa por ser executada uma Inicialização antes do controlo passar para um módulo Gestor que assegura a execução dos processos paralelos e a interface com o utilizador ao nível da exploração on-line. A execução das funções de Monitorização e Registo deve ser assegurada em background de forma sincronizada, enquanto a execução das outras funções disponíveis em foreground depende das selecções do utilizador.

Mapas de Estrutura de 2ª Ordem - Configuração

Módulo 1.1.: Segurança

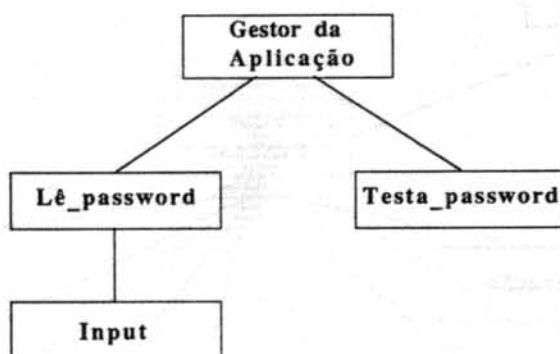


Figura 2-13: ME de 2ª Ordem - Segurança

Para implementar uma protecção simples à aplicação começa por ser chamado o módulo Lê_password que chamando Input formata a entrada de um password pelo utilizador(operador, supervisor ou de defeito). Depois, sempre que necessário é chamado Testa_password que avalia o tipo de utilizador e permite limitar ou interromper a execução da aplicação.

Módulo 1.2.: Configurar Sinópticos



Figura 2-14: ME de 2ª Ordem - Configurar Sinópticos

Este módulo é implementado através de um editor gráfico que deve permitir definir os quadros sinópticos produzindo dois tipos de informações, um sobre a imagem desenhada e outro sobre a animação especificada, a gravar em ficheiros, por selecção do utilizador, que constituem a forma de comunicação com a aplicação SP em que se vai integrar.

Módulo 1.3.1. — Configurar Variáveis/Processo

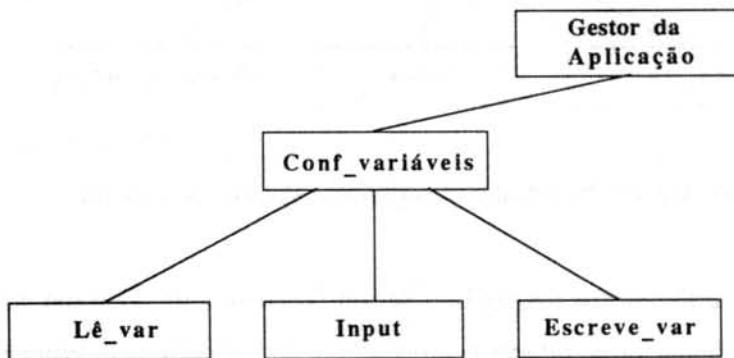


Figura 2-15: ME de 2ª ordem — Configurar Variáveis/Processo

Seleccionada pelo utilizador a opção Configurar Variáveis do Processo é executado o módulo Conf_variáveis que chama o módulo Lê_var para numa leitura a disco das configurações anteriores ou de defeito das variáveis que caracterizam o processo. Estas configurações podem ser alteradas por Conf_variáveis, sendo a entrada de dados formatados assegurada por chamadas a Input. Completado o processo, as configurações são gravadas em disco por uma chamada a Escreve_var e o controlo passa para o Gestor da Aplicação.

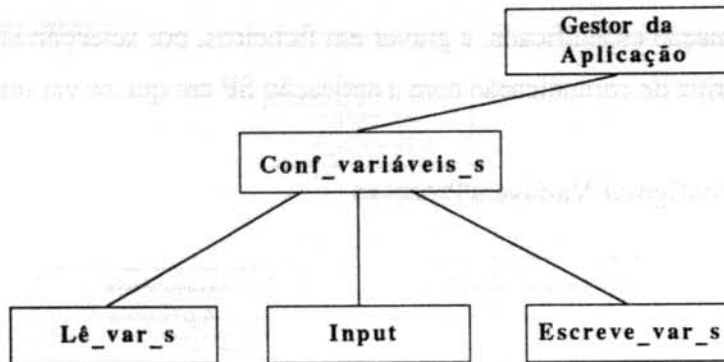
Módulo 1.3.2.: Configurar Variáveis/Sistema

Figura 2-16: ME de 2ª ordem — Configurar Variáveis/Sistema

Seleccionada pelo utilizador a opção Configurar Variáveis do Sistema é executado o módulo Conf_variáveis_s que chama o módulo Lê_var_s para uma leitura a disco das configurações anteriores ou de defeito, das variáveis que caracterizam o sistema. Estas configurações podem ser alteradas por Conf_variáveis_s, sendo a entrada de dados formatados assegurada por chamadas a Input. Completado o processo, as configurações são gravadas em disco por uma chamada a Escreve_Var_s e o controlo passa para o Gestor da Aplicação.

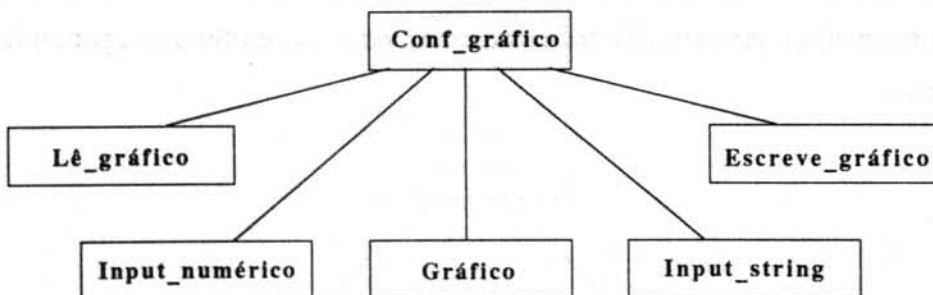
Módulo 1.4.: Configurar Gráficos

Figura 2-17: ME de 2ª ordem - Configurar Gráficos

Seleccionada pelo utilizador a opção Configurar Gráficos (quer off-line quer on-line), é executado o módulo Conf_gráfico que começa por chamar Lê_gráfico para leitura a disco das configurações de um dado gráfico permitindo depois que o utilizador especifique ou

altere estas configurações. As entradas formatadas são asseguradas por chamadas a `Input_numérico` para dados numéricos e a `Input_string` para dados alfanuméricos.

Com base nas configurações especificadas é possível visualizar o aspecto do gráfico a produzir, executando `Gráfico`, o que facilita um ajuste interactivo das configurações.

Quando o processo é completado, as configurações são gravadas em disco por `Escreve_gráfico` e o controlo passa para o Gestor da Aplicação ou para o Gestor On-line conforme a chamada.

Módulo 1.5.: Configurar Tabelas

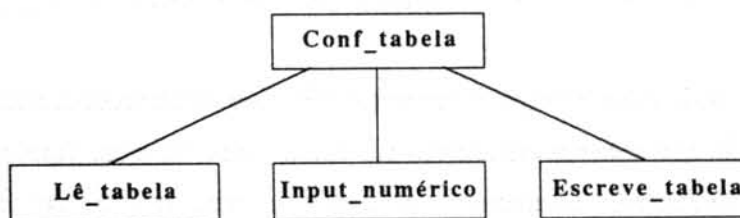


Figura 2-18: ME de 2ª Ordem - Configurar Tabelas

Seleccionada pelo utilizador a opção Configurar Tabelas (quer on-line quer off-line), é executado o módulo `Conf_tabela` que começa por chamar `Lê_tabela` para leitura a disco das configurações de uma dada tabela permitindo depois que o utilizador especifique ou altere estas configurações. As entradas formatadas de dados numéricos são asseguradas por chamadas a `Input_numérico` e quando o processo é completado as configurações são escritas em disco por chamada a `Escreve_tabela`. Quando termina, o controlo é passado ao Gestor da Aplicação ou ao Gestor On-line conforme a chamada.

Mapas de Estrutura de 2ª Ordem - Exploração On-Line

Módulo 2.0: Inicialização



Figura 2-19: ME de 2ª ordem — Inicialização

Seleccionada pelo utilizador a opção Exploração On-line, o controlo é passado para o módulo Inicialização que começa por chamar Lê_canal para leitura das configurações das variáveis a disco. Executadas as alocações dinâmicas e inicializações são chamados os módulos Monit_calibra e Monit_instala para iniciar o processo periódico em background de aquisição de dados e monitorização de alarmes, e a seguir é chamado o módulo Reg-instala para iniciar o processo em background de arquivo de dados e registo de alarmes.

Módulo 2.1.1: Configuração/Gráficos

Idêntico ao módulo 1.4. Configurar Gráficos na Configuração da Aplicação (2ª Ordem)

Quando termina o controlo passa para o Gestor On-Line.

Módulo 2.1.2: Configuraração/Tabelas

Idêntico ao módulo 1.5. Configurar Tabelas na Configuração da Aplicação (2ª Ordem)

Quando termina o controlo passa para o Gestor On-line.

Módulo 2.4: Alarmes

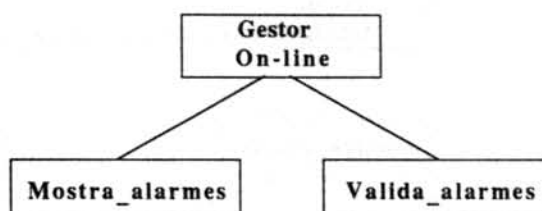


Figura 2-20: ME de 2ª Ordem - Alarmes

O utilizador pode seleccionar duas opções, visualizar ou validar as situações de alarme ocorridas. A estas opções corresponde a execução do módulo correspondente. Terminando o controlo regressa ao Gestor On-line.

Módulo 2.5.1.: Análises/Gráficos

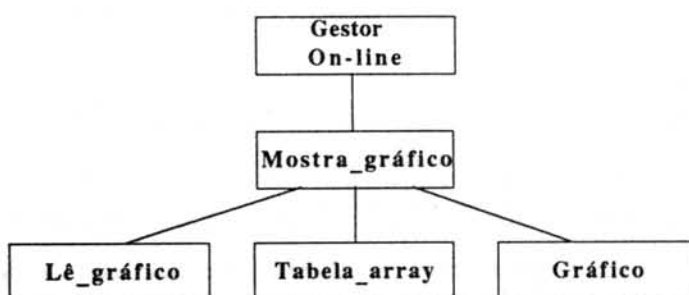


Figura 2-21: ME de 2ª Ordem - Análises/Gráficos

Seleccionada pelo utilizador a opção Gráficos é executado o módulo Mostra_gráfico que chama Lê_gráfico para uma leitura a disco das configurações existentes, chama tabela_array que passa os dados mais recentes do buffer Tabela para as variáveis indexadas usadas pelo gerador de gráficos, Gráfico, que é então chamado para produzir o gráfico pretendido no écran. Terminando o controlo regressa ao Gestor On-line.

Módulo 2.5.2.: Análises/Tabelas



Figura 2-22: ME de 2ª Ordem - Análises/Tabelas

Seleccionada pelo utilizador a opção Análises/Tabelas, é executado **Mostra_tabela** que chama **Lê_Tabela** para uma leitura a disco das configurações existentes chama a **Armação** para desenhar o quadro e vai chamando **Mostra_valor** para a impressão de valores no écran, no quadro.

Quando termina o controlo passa para o **Gestor On-line**.

Módulo 2.6.: Sinópticos

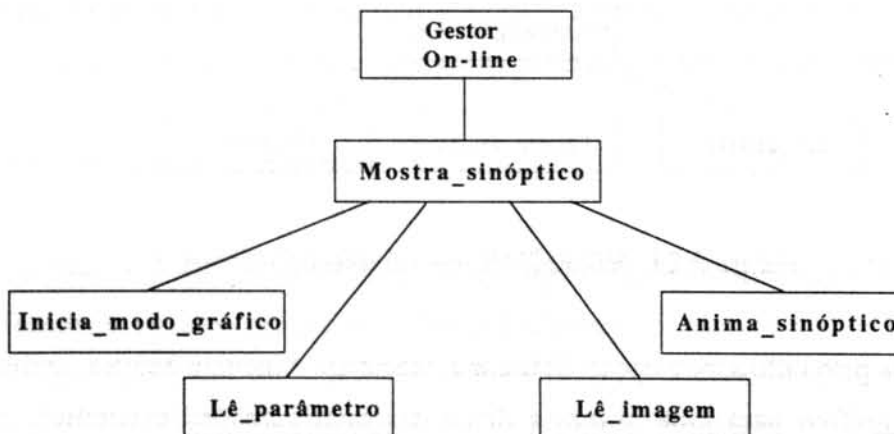


Figura 2-23: ME de 2ª Ordem - Sinópticos

Seleccionada pelo utilizador a opção Sinópticos, para visualização de quadros sinópticos, é executado o módulo **Mostra_sinóptico** que chama **Inicia_modos_gráfico** para inicializar a interacção gráfica e, conforme o sinóptico seleccionado, chama **Lê_parâmetros** e depois **Lê_imagem**, para leitura a disco dos parâmetros e da imagem correspondentes. O quadro sinóptico é mostrado no écran com a actualização de valores garantida por chamadas

sucessivas a Anima_sinótico. Se outro quadro sinótico for entretanto seleccionado estes procedimentos são repetidos.

Terminando o controlo passa para o Gestor On-line.

Módulo 2.7.: Operação

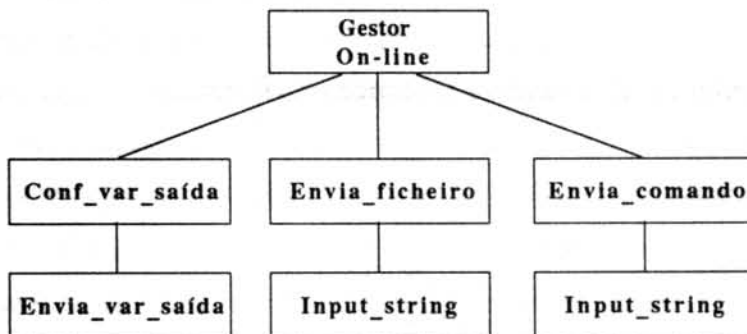


Figura 2-24: ME de 2ª Ordem - Operação

A função Operação, para actuação manual sobre o processo, é seleccionável em três possibilidades diferentes.

Pode ser alterada a configuração de uma dada variável do processo especificada como de saída (Output) e serem enviadas as suas características actuais ao SAC, ou pode ser enviado, pela porta série, um ficheiro de comandos pré-definido, ou ainda ser enviado, também pela porta série, um comando de actuação. Terminando cada uma das opções o controlo passa ao Gestor On-line.

Módulo 2.8: Terminações

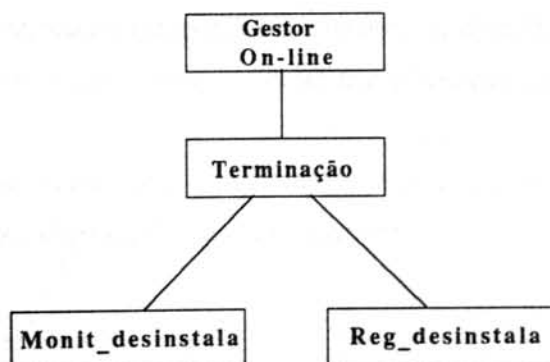


Figura 2-25: ME de 2ª ordem — Terminação

Seleccionado pelo utilizador o fim da exploração on-line, é executado o módulo Terminação que começa por chamar Monit_desinstala que interrompe o processo corrente de monitorização e Reg_desinstala que interrompe o processo corrente de registo. Em seguida é libertado o espaço em memória alocado dinamicamente. O controlo passa ao Gestor da Aplicação.

Mapas de Estrutura de 2ª Ordem - Operação Off-line

Módulo 1.7.: Conversão

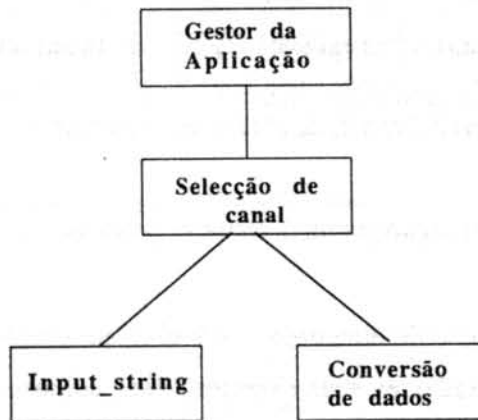


Figura 2-26: ME de 2ª Ordem - Conversão

Seleccionada pelo utilizador a opção Conversão é executada a Seleção de canal que chama Input_string para a entrada formatada do número do canal. É então executada, pelo módulo Conversão de Dados, a escrita em disco dos dados do canal seleccionado num formato exportável (modo texto, ASCII).

2.3.2 Concepção dos Módulos

Os vários módulos identificados na produção dos Mapas de Estrutura são especificados para codificação com base num subconjunto de itens propostos pela IEEE no projecto 1060 [28] para documentação da fase de concepção.

Estes itens constituem fichas de especificações, por cada módulo, apresentadas no Anexo C e aqui exemplificadas para o módulo Gestor da Aplicação.

Estas especificações podiam ser mais detalhadas e formais, nomeadamente nos Dados, Interfaces e Processamento, caso quem tivesse trabalhado na concepção não fosse trabalhar na codificação ou caso fossem ser alvo de uma validação formal, automática.

Os módulos de software cuja concepção é descrita são identificados pelo nome usado nos Mapas de Estrutura seguido do nome proposto para implementação.

O tipo de módulo inclui uma classificação em interface, entrada, saída, processamento e gestão, que caracteriza a actividade principal do módulo.

Ficha de Especificação

Módulo: Gestor da Aplicação

Tipo:

Procedimento principal (interface)

Funções:

Implementa a interface principal com o utilizador usando menus de barras.

Permite seleccionar e chamar os procedimentos que implementam processamento Off-Line: Configuração da Aplicação e Operação.

Permite chamar o procedimento que implementa a Exploração On-Line.

Oferece valores de defeito para as configurações da aplicação.

Subordinados:

Configurar Sinópticos, Configurar Variáveis, Configurar Gráficos, Configurar Tabelas, Validar Configurações, Exploração On-Line, Conversão.

Dependência:

Interfaces:

Os procedimentos chamados são essencialmente independentes, poderiam ser todos programas executáveis. Cada procedimento é então responsável por implementar as suas

entradas de dados e por produzir as suas saídas, sendo a comunicação entre os procedimentos aqui usados assegurada pela partilha de dados em ficheiro.

Processamento:

Chama Segurança

Actualiza menus e conforme a selecção do utilizador, chama:

Configurações

Sinópticos

Variáveis

Gráficos

Tabelas

Validar

Operação Off_line

Conversão

Lotus

DBase

Exploração On-line

Dados:

Configurações globais do sistema.

Restrições:

A interface aqui implementada deve poder estender-se a todo o processamento off-line, mantendo a hierarquia proposta nos Mapas de Estrutura.

Irão poder ser configurados 6 gráficos e 6 tabelas.

Os procedimentos associados ao processamento off-line são essencialmente independentes não tendo necessidade de grandes trocas de informação.

A utilização dos procedimentos deve permitir que estes passem a ser programas executáveis. Os procedimentos associados ao processamento Off-Line, que não sejam necessários, não devem permanecer residentes em RAM na Exploração On-Line, especialmente os de grande dimensão.

Este módulo deve funcionar como um centro de transacções.

2.3.3 Concepção dos Dados

Os vários depósitos de dados definidos na Análise por DFD's são aqui modelizados para implementação. São apresentados depósitos concretizados tanto por ficheiros em disco como por buffers em memória.

Nome: Tabela (buffer).

Dados: São o Valor da leitura e o Tempo em que esta é executada para um canal do sistema de aquisição de dados ao qual corresponde uma variável do processo.

Escrita: Monitorização.

Leitura: Registo, Sinópticos, Análises e Operação.

Função: Implementar a comunicação de dados entre processos. Indirectamente sincronizar o funcionamento dos processos dependentes da disponibilidade de novos dados.

Restrições:

- A colocação dos dados no buffer de ser um procedimento simples para não pesar na operação da Aquisição.
- Os dados devem ter uma organização que facilite o acesso por canal e por ordem cronológica.
- O espaço em RAM reservado para dados é limitado.
- Cada variável é caracterizada por um número máximo de valores que devem existir no buffer para implementar as funções da exploração on-line.
- A dimensão do buffer deve ser configurável em run-time, quanto ao número de canais (variáveis) e quanto ao número máximo de valores (valor e tempo) armazenado por canal.
- O acesso ao buffer deve ser condicionado tendo em conta a concorrência de processos de escrita e leitura de dados.

Arquitectura de Dados: A estrutura elementar é uma leitura constituída por Valor e Tempo em que Valor é um real e Tempo um inteiro positivo.

Um vector de estruturas para cada canal é a base de uma fila circular que retém os dados mais recentemente adquiridos.

Um vector de apontadores para os vectores dos canais agrega o buffer, uniformiza o seu endereçamento e facilita o dimensionamento em run-time.

A estrutura resultante é esquematicamente a representada na figura 2-27.

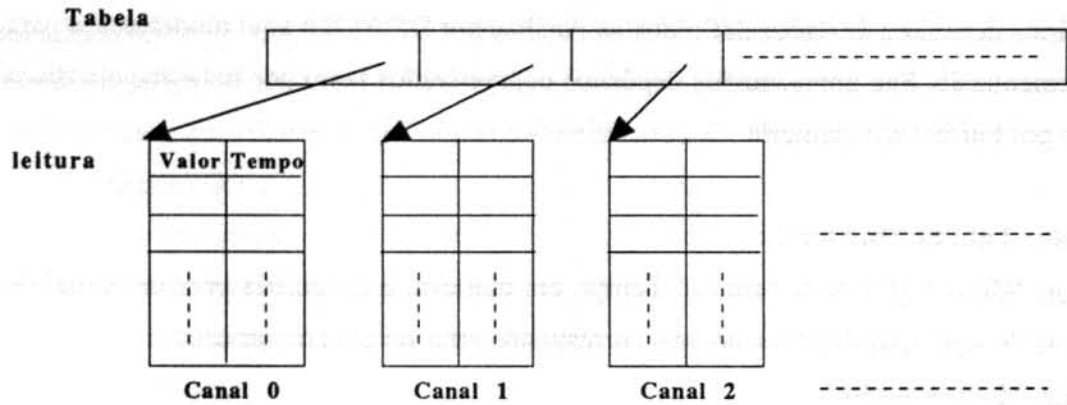


Figura 2-27: Arquitectura do Buffer Tabela

Filas Circulares

O funcionamento em filas circulares dos vectores de leituras dos canais é assegurado por um apontador para o último valor colocado no vector, a Frente da fila.

No início cada vector dos canais está inicializado com zeros. O apontador, Frente da fila, aponta para a primeira posição do vector. Sempre que são colocadas leituras no vector, Frente é incrementado e passa a apontar para a posição seguinte.

Se for atingida a última posição do vector, o apontador Frente, em vez de ser incrementado, é feito apontar para a primeira posição do vector iniciando a sobreposição dos valores mais atingidos pelos mais recentes. Em estado estacionário a leitura mais antiga é a que segue à apontada por Frente.

Um vector de apontadores para a frente dos vectores de dados dos canais (Frente) implementam as filas circulares dos canais existentes. Um outro vector (Tamanho), guarda os valores dos tamanhos das filas circulares da cada canal.

A estrutura resultante é esquematicamente a representada na figura 2-28.

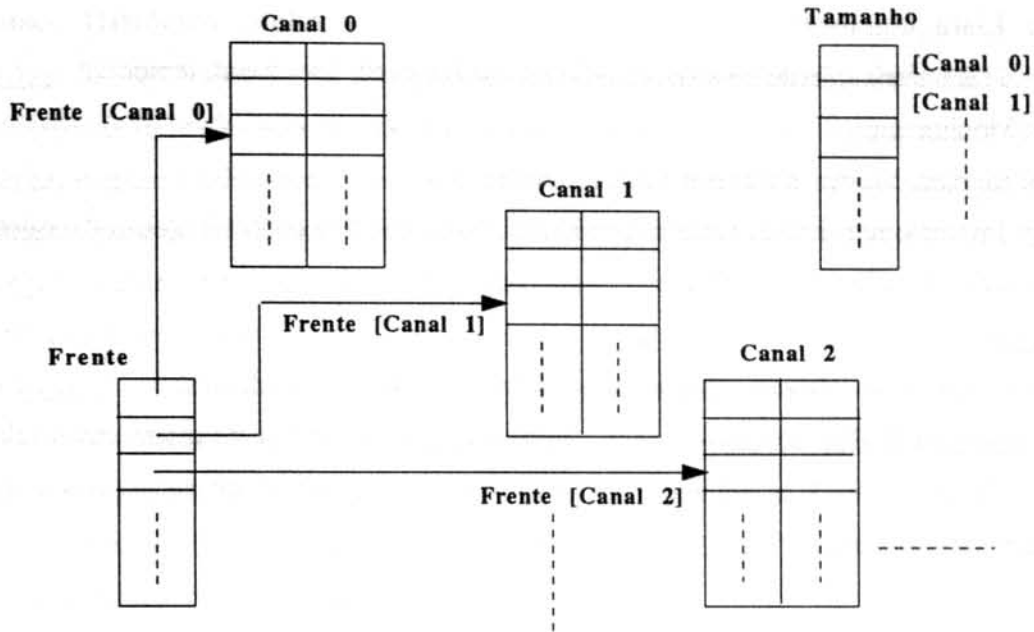


Figura 2-28: Arquitectura das Filas Circulares

Nome: Cnf_Variáveis (buffer)

Dados: As configurações das Variáveis do processo.

Escrita: Inicialização ou Configuração da aplicação.

Leitura: Monitorização, Alarmes, Operações e Configuração.

Função: Disponibilizar em memória as configurações das variáveis do processo especificadas na Configuração da Aplicação.

Restrições:

- As configurações das variáveis devem estar o mais acessíveis possível aos processos de tempo real.
- O acesso é feito por canal.
- Devem poder adicionadas novas características para cada variável.

Arquitectura de Dados: Propõe-se a implementação por um vector de estruturas. Cada posição do vector configura uma variável com uma estrutura que agrupa as suas várias características: nome da variável, tipo (I ou O), endereço no SAC, tamanho da fila circular, frequência de aquisição, arquivo (Sim ou Não), níveis de alarme superior e inferior e valor do set point.

Nome: Lista_ alarmes

Dados: A caracterização de uma situação de alarme (número, tipo, canal, tempo).

Escrita: Monitorização.

Leitura: Registo, Mostra_alarmes e Valida_alarmes.

Função: Implementar lista de alarmes pendentes, assegurando a comunicação entre estes módulos.

Restrições:

— Deve oferecer um número máximo (configurável) de alarmes pendentes.

Arquitectura de Dados: Propõe-se a implementação por um vector de estruturas. Cada posição do vector está associada a um alarme que é caracterizado pelas variáveis agregadas na estrutura.

Nome: Configurações (ficheiros)

Dados: São as configurações das Variáveis (de Processo e de Sistema) e das Análises (de Gráficos e de Tabelas).

Escrita: Na Configuração da Aplicação e no processo Configuração na Exploração On-line.

Leitura: Na Exploração On-line

Função: Guardar com carácter permanente as configurações da aplicação e disponibilizá-las na Exploração On-line.

Restrições:

— As configurações das Variáveis têm que estar disponíveis em RAM para uso facilitado pelos processos de execução em background, pelo que são copiadas para o buffer Cnf_Variáveis antes de serem usadas.

— As configurações dos Gráficos e Tabelas são usadas de forma independente a partir de disco, sempre que o utilizador seleccionar as opções correspondentes. Devem portanto poder ser lidas rapidamente.

Arquitectura de Dados: O depósito Configurações é implementado por um ficheiro para as variáveis de processo (VAR.CNF), outro para as de sistema (VAR SIS.CNF) por um conjunto de ficheiros para os gráficos, um por cada (GRAF#.CNF) e por um conjunto de ficheiros para as tabelas, um por cada (TAB#.CNF).

Nome: Histórico (ficheiro)

Dados: São os Valores e os Tempos das leituras correspondentes a cada variável do processo configurada como de entrada (input).

Escrita: Registo na Exploração On-Line.

Leitura: Operação Off-Line.

Função: Armazenar, com carácter permanente, dados adquiridos do processo para tratamentos posteriores.

Restrições:

- A colocação dos dados em ficheiro em disco deve ser um procedimento simples para não pesar na operação de Arquivo.
- Os dados devem ter uma organização que facilite o acesso por canal.
- Cada canal pode ter um número diferente de valores.

Arquitectura de dados: A cada canal corresponde um ficheiro em disco, binário, em que o nome indica o número do canal e que contém os sucessivos valores do Valor e do Tempo das leituras, (CANAL#.DAD, # - número do canal).

Nome: Histórico - ASCII (ficheiro)

Os ficheiros que implementam este depósito de dados são em tudo idênticos aos anteriores mas num formato ASCII compatível com aplicações comerciais de uso geral (CANAL#.TXT).

Só apresentamos os dados considerados mais relevantes, nomeadamente, por serem utilizados por vários módulos ou pela sua complexidade.

Os outros ficheiros de dados usados são os dos Écrans e Parâmetros dos sinópticos.

Os outros buffers de dados em memória incluem as configurações das variáveis globais de sistema, dos gráficos, das tabelas e dos parâmetros dos sinópticos. E ainda os dados associados ao tratamento das situações de alarme.

2.3.4 Concepção da Interface

A interface com o utilizador será baseada em munus de tipo barra que proporcionam uma interação simples.

O conjunto de funções disponíveis devem ser apresentadas ao utilizador de acordo com a hierarquia proposta nos DFD's e nos ME's. A ordem de apresentação deverá conduzir o utilizador por uma metodologia proposta para a operação com a aplicação.

Assim o sistema de opções a ser apresentado ao utilizador deverá obedecer à seguinte árvore de menus:

- Configurar
 - Variáveis
 - Sinópticos
 - Análises
 - Gráficos
 - Tabelas
 - Hardware
- Validar
- Exploração On-Line
- Operação Off-Line
 - Converter
 - Lotus
 - DBase

Para a Exploração On-Line a árvore de menus será a seguinte:

- Sinópticos
- Análises
 - Gráficos
 - Tabelas
- Alarmes
 - Visualizar
 - Validar

Operar

Configurar

Análises

Gráficos

Tabelas

Variáveis

STOP

Como previsto na Concepção de Processos, as funções da Exploração On-Line devem ser implementadas de forma o mais independente do resto da aplicação. Assim, foram também previstos dois sistemas de menus independentes.

A implementação dos menus deve corresponder a um módulo independente e com uma configuração e actualização simples das opções disponíveis.

Propõe-se a utilização de um sistema de menus disponível e que corresponde bem ao pretendido.

O sistema é baseado nos ficheiros Menus.obj e Menus.h e no ficheiro de definições Menu.def que assegurará a comunicação com a aplicação, impondo uma coesão a nível de dados que assegura a independência desejada. Alterações ao Menu.def são reservadas à manutenção e desenvolvimento e não ao utilizador.

2.4 Packaging

Começamos por definir a forma da aplicação SProcess tal como irá ser vista pelo utilizador, em termos de programas executáveis (.EXE), de ficheiros de informação e da estrutura de directórios que os organiza.

É também definida a forma da aplicação sobre o no ponto de vista de desenvolvimento em termos da estrutura de módulos que agregam os vários procedimentos a codificar.

2.4.1 Packaging de Programas

SP-Supervisão de Processo constitui o programa principal da aplicação e inclui a interface com o utilizador e todos os procedimentos associados à Exploração On-Line. Todos os procedimentos associados à Configuração e à Operação Off-Line só devem ser incluídos se também forem, pelo menos parcialmente, usados pela Exploração On-Line.

A inclusão de procedimentos no programa principal, que vai residir permanentemente em RAM, deve obedecer a considerações de ocupação de RAM. A execução da aplicação pode exigir um espaço em RAM significativo para o armazenamento de dados adquiridos de acordo com o que for configurado. Minimizar o número e a dimensão dos procedimentos residentes na Exploração On-Line é então um objectivo.

Nome: SP.EXE

Ficheiros incluídos:

- main.c
- on-line.c
- gráficos.c
- tabelas.c
- variável.c
- kh 500.c
- sinoptic.c
- password.c
- registo.c
- alarmes.c

Interfaces:

Comunica com a Configuração da Aplicação através dos ficheiros de Écrans, Parâmetros e Configurações, e com a Operação Off-line pelos ficheiros de Histórico.

Nome: EDITOR.EXE

Implementa a configuração de quadros sinóticos.

Constitui um sub-programa de execução independente da restante Aplicação. Terminando liberta a memória RAM ocupada.

Interfaces:

É chamado pelo Gestor da Aplicação.

A comunicação com a aplicação é feita através dos ficheiros de Écrans e de Parâmetros.

Nome: CONV_ASC.EXE

Implementa, na Operação Off-Line, a conversão dos ficheiros de dados para um formato exportável (ASCII).

Constitui um sub-programa de execução independente da Aplicação. Terminando a conversão liberta a memória RAM ocupada.

Ficheiros incluídos:

conv_asc.c

Interfaces:

É chamado pelo Gestor da Aplicação. Comunica com a aplicação pelos ficheiros de dados Histórico e Histórico - ASCII.

Nome: VALIDA.EXE

Implementa validação da coerência das configurações especificadas para a aplicação e guardadas em ficheiros (directório CNF).

Ficheiros incluídos:

valida.c

Interfaces:

É chamado pelo Gestor da Aplicação. Não exige parâmetros de execução. Faz leitura dos ficheiros de configurações e produz um relatório para o écran conforme as análises que efectua.

2.4.2 Packaging de Dados

Os depósitos de dados projectados são implementados como ficheiros ou conjuntos de ficheiros organizados em directórios. As designações a usar são tipificadas como se segue:

Nome: Depósito 02 - Configurações

Directório SP\CNF\

Ficheiros VAR.CNF - Configurações das variáveis do processo.

Ficheiros TAB#.CNF - Configurações da tabela nº#.

Ficheiros GRAF#.CNF - Configurações do gráfico nº#.

Ficheiros VARSIS.CNF - Configurações globais ao sistema.

Nome: Depósito 07 - Histórico

Directório SP\DADOS\

Ficheiros CANAL#.DAD - Dados da variável nº#

Nome: Depósito 10 - Histórico - ASCII

Ficheiros CANAL#.TXT - Dados ASCII da variável nº#

Nome: Depósito 04 - Imagens

Directório SP\PROCESS\IMG\

Ficheiros ECRAN#.IMG - Imagem do écran sinóptico nº#

Nome: Depósito 05 - Parâmetros

Directório SP\PROCESS\PAR\

Ficheiros ECRAN#.PRO - Parâmetros referentes ao écran nº#

Nome: Símbolos usados pelo Editor

Directório SP\SYMBOLS\

Ficheiros *.SIM - Biblioteca de figuras pré-definidas

Nome: Descrição dos écrans usados pelo Editor

Directório SP\PROCESS\EDIT\

Ficheiros *.ED - Descrição vectorial dos écrans.

2.4.3 Estructura de Directórios

SP	• Obj • Exe											
	SYMBOLS		• Sim									
	PROCESS		<table><tr><td>IMG</td><td> </td><td> • Img </td></tr><tr><td>PAR</td><td> </td><td> • Pro </td></tr><tr><td>EDIT</td><td> </td><td> • Ed </td></tr></table>	IMG		• Img	PAR		• Pro	EDIT		• Ed
IMG		• Img										
PAR		• Pro										
EDIT		• Ed										
	CNF		• Graf • Tab • Var • Sis									
	DADOS		• Dad • Txt									
	HEADERS		• H									
	SOURCES		• C • Def									

2.4.4 Packaging de Procedimentos

Procura-se associar procedimentos que correspondem à execução da mesma função da aplicação mas, nas funções em que os procedimentos de configuração não sejam necessários na exploração, deverão constituir-se dois ficheiros separados, um com os procedimentos de configuração e outro com os de exploração. É então proposta uma organização dos vários procedimentos em ficheiros de código (*.c) diferentes que corresponderão a ficheiros compilados (*.obj) diferentes no ponto de vista do desenvolvimento, facilitando a manutenção e as evoluções posteriores.

Nome: kh500.c

Driver para o SAC Keithley 500. Os vários procedimentos específicos do hardware de monitorização do processo usado devem ser agrupados num ficheiro independente.

Outros drivers podem então ser desenvolvidos a partir do existente para permitir a utilização pela aplicação de outros SAC. Só este ficheiro teria que ser compilado e linkado à aplicação.

Procedimentos incluídos:

monit_instala - usado na Inicialização para instalar a rotina de serviço às interrupções Monit_int.

monit_calibra - usado na Inicialização para executar auto-calibração do SAC.

monit_int - implementa a Monitorização

monit_desinstala - usado na Terminação para desinstalar Monit_int

Interfaces:

A comunicação com a aplicação é feita fundamentalmente pelos buffers Cnf_Variáveis e Tabela.

Nome: tabelas.c

A utilização de tabelas de dados envolve uma configuração (On-line e Off-line) e uma representação (On-line). Os procedimentos que implementam estas funções devem constituir um ficheiro independente.

Módulos incluídos:

le_tabela — entrada de configurações de disco

`conf_tabela` — interface com o utilizador e gestão da configuração das tabelas.

`escreve_tabela` — saída de configurações para disco.

`mostra_tabela` — processa a representação de uma tabela na Exploração On-line.

`armacao` — desenha os contornos da tabela.

`mostra_valor` — coloca um novo valor da tabela no écran.

Interfaces:

A comunicação com a aplicação é feita pelos buffer `Tabela` e `Cnf_Variáveis`.

Nome: sinópticos.c

Os módulos associados à representação de sinópticos em Exploração são agregados num ficheiro.

Procedimentos incluídos:

`mostra_sinoptico` - representa no écran o sinóptico especificado

`le_imagem` - entrada por leitura a disco da imagem do écran do sinóptico especificado.

Interfaces:

A comunicação com a aplicação é feita através do buffer `Tabela` e dos ficheiros de configuração dos écrans sinópticos (`*.img` e `*.pro`).

Nome: conv_asc.c

Foi proposto que a conversão do formato dos ficheiros de dados fosse um programa executável independente. Assim propõe-se que os procedimentos relacionados com esta função estejam agregados num ficheiro.

Procedimentos incluídos:

`selecção de canal` - interface com o utilizador para entrada do canal a converter

`conversão de dados` - processa conversão dos dados do canal para o formato

ASCII

Interfaces:

A comunicação com a aplicação é assegurada pelos ficheiros de dados `Histórico` e `Histórico-ASCII`.

Nome: operar.c

Agrupar funções associadas à actuação manual sobre o processo.

Procedimentos incluídos:

conf_var_saída - permite alterar as configurações das variáveis tipo saída (output).

envia_var_saída - envia ao SAC um comando de actuação a partir do configurado para uma variável de saída.

envia_comando - envia pela porta série um comando de actuação.

envia_ficheiro - envia pela porta série um ficheiro de comando previamente definido.

Interfaces:

A comunicação com a aplicação é feita através do buffer Conf_variáveis.

Nome: gráficos.c

A utilização de gráficos sobre dados do processo envolve uma configuração (On-line e Off-line) e uma representação (On-line).

Os procedimentos que implementam estas funções devem constituir um ficheiro independente.

Procedimentos incluídos:

conf_gráfico - interface com o utilizador para caracterização de uma análise baseada em gráficos.

lê_gráfico - entrada por leitura de disco das configurações de um gráfico.

escreve_gráfico - saída por escrita em disco da configuração de um gráfico.

mostra_gráfico - processa a chamada do gerador de gráficos.

tabela_array - processa cópia dos dados necessários do buffer Tabela para as variáveis indexadas sobre o qual o gerador de gráficos actua.

Interfaces:

Comunicação com a aplicação através dos ficheiros de configurações dos gráficos e dos buffers Tabela e Configuração de gráficos.

Nome: registo.c

Agrupar as funções de arquivo de dados em disco e de impressão de mensagens associadas a situações de alarme.

Procedimentos incluídos:

reg_instala - usado na Inicialização para instalar a rotina de serviço a interrupções
Reg_int.

reg_desinstala - usado na Terminação para desinstalar Reg_int.

reg_int - implementa as funções de registo.

Interfaces:

A comunicação com a aplicação é assegurada fundamentalmente pelo buffer Tabela e Lista_alarmes.

Nome: main.c

Implementa a gestão geral da aplicação SProcess.

Procedimentos incluídos:

main - interface com o utilizador e gestão das várias funções de Configuração, da Exploração On-line e das funções da Operação Off-line.

Nome: password.c

Implementa protecção à utilização da aplicação.

Procedimentos incluídos:

lê_pass - interacção com o utilizador para entrada do respectivo password.

pass_test - testa a autorização que o utilizador possui.

Interfaces:

Comunica com a aplicação pelo buffer das configurações globais e pela variável que guarda o password.

Nome: alarme.c

Agrupa os procedimentos associados à interacção com o utilizador para tratamento de situações de alarme.

Procedimentos incluídos:

mostra_alarm - visualização das situações de alarme ocorridas.

valida_alarm - validação de situações de alarme que são então retiradas da lista de alarmes pendentes.

Interfaces:

A comunicação com a aplicação é assegurada pelo buffer Lista_alarmes e variáveis auxiliares.

Nome: variável.c

Agrupa os procedimentos associados à configuração das variáveis que caracterizam o sistema na globalidade e também das variáveis que caracterizam o processo em supervisão.

Procedimentos incluídos:

lê_var - entrada por leitura a disco das configurações das variáveis do processo.

lê_var_sis - entrada por leitura a disco das configurações das variáveis globais do sistema.

cfg_var - interface com o utilizador para especificação das variáveis do processo.

cnf_var_sis - interface com o utilizador para especificação das variáveis globais do processo.

escreve_var - saída por escrita em disco das configurações das variáveis do processo.

escreve_var_sis - saída por escrita em disco das configurações das variáveis globais do sistema.

Interfaces:

A comunicação com a aplicação é assegurada pelo depósito Configuração implementado pelos ficheiros var.cnf e varsis.cnf.

Nome: on-line.c

Agrupa as funções relacionadas com a Exploração On-line.

Procedimentos incluídos:

g_on_line - interface e gestão do processamento associado à Exploração On-line.

inicia - processa a inicialização da Exploração On-line.

termina - processa a terminação da Exploração On-line.

tabela_array - processa a passagem de dados do buffer Tabela para as variáveis do gerador de gráficos (Graf).

Interfaces:

A comunicação com a aplicação é feita através dos ficheiros que implementam o depósito Configurações.

CAPÍTULO 3

Implementação: Exemplos

Com base no projecto de software apresentado no Capítulo 2 foi iniciado o desenvolvimento da aplicação SProcess.

São aqui apresentados dois exemplos de implementações da concepção realizada (dados e processos) que ilustram a forma como alguns dos aspectos mais particulares do projecto foram resolvidos.

A linguagem de programação é o C Turbo C.

3.1 Exemplo 1 - Criação da estrutura de dados Tabela

Na inicialização da exploração on-line é criada em memória a estrutura de dados Tabela que vai armazenar temporariamente os dados adquiridos do processo. Descreve-se aqui a implementação efectuada tanto para o buffer Tabela, de acordo com o concebido no capítulo 2.3, como para outras estruturas que asseguram a manipulação dos dados adquiridos do processo.

Funções implementadas no código apresentado na figura 3 - 1:

— Alocação dinâmica de memória a variáveis de acordo com a configuração da aplicação. As estruturas de dados dependentes do número de canais são dimensionadas dinamicamente o que permite manter optimizada a utilização do recurso memória.

O buffer Tabela é constituído por variáveis indexadas de estruturas do tipo leitura (tempo, valor) previamente definido.

As alocações são efectuadas pelo comando calloc(), que inicializa a zeros as estruturas criadas, nomeadamente:

— tamanho: guarda os tamanhos de cada fila circular;

- tabela: variável indexada (dimensão = número de canais) de apontadores para variáveis indexadas (dimensão = tamanho de cada canal) de estruturas do tipo leitura (tempo, valor).
- frente: implementa filas circulares na Tabela. Guarda apontadores para a frente das filas.

```

.
.
.
tamanho=(unsigned int *)calloc(totalcanais,sizeof(unsigned int));
for(i=0;i<totalcanais;i++)
    tamanho[i]=var_conf[i].tamanho;

tabela=(t_leitura **)calloc(totalcanais,sizeof(t_leitura *));
for(i=0;i<totalcanais;i++)
    tabela[i]=(t_leitura *)calloc(tamanho[i],sizeof(t_leitura));

frente=(t_leitura **)calloc(totalcanais,sizeof(t_leitura *));
for(i=0;i<totalcanais;i++)
    frente[i]=tabela[i];
.
.
.

```

Figura 3-1: Código retirado do módulo Inicialização - inicia ()

3.2 Exemplo 2 - Processo em background: Monitorização com o Sistema de Aquisição de Dados e Controlo Keithley 500.

A aquisição de dados é assegurada por uma rotina de serviço a interrupções, Monit_int, que é instalada no processo de Inicialização do processamento on-line por chamada às rotinas Monit_calibra e Monit_instala, e desactivada no processo de Terminação do processamento on-line por chamada à rotina Monit_desinstala.

São estas rotinas que constituem o ficheiro k500.c e devem ser reescritas para um SAC diferente do agora usado.

As funções destas rotinas são especificadas como se segue:

Módulo: Monit_instala**Tipo:**

Procedimento (processamento).

Funções:

Inicia a execução em background da Monitorização.

Subordinados:**Dependência:**

É chamada na Inicialização da Exploração On-line.

Interfaces:

São usados os dados globais do SAC. Precisa do endereço da rotina Monitorização e da frequência pretendida para programar o relógio do SAC. Devolve o conteúdo da posição 0x0A da tabela na variável endereço OA.

Processamento:

Programa o relógio do SAC Keithley para periodicamente gerar interrupções tipo IRQ2.

Desactiva geração de interrupções

Instala a rotina Monit_int de serviço às interrupções para monitorização, na posição 0x0A da tabela de vectores do DOS.

Activa geração de interrupções

Activa a geração das interrupções pelo SAC.

Dados:

Não usa dados internos.

Restrições:

A interrupção que implementa a Monitorização deve ser de maior prioridade, já que os dados adquiridos são a matéria prima da aplicação.

A geração de interrupções é assegurada por um relógio do SAC.

Este módulo é específico do SAC Keithley 500.

É um procedimento específico do hardware de aquisição de dados -Keithley 500.

PROCEDIMENTO

MONIT_INSTALLA - instala e inicia a interrupção para aquisição de dados e monitorização de alarmes

PARÂMETROS

NOTAS

ULTIMA ALTERAÇÃO

```

void monit_instala()
{
    int i=0;

    /* Programar Relógio - Período : 6*ffff = 393210 tics */

    pokeb(END_IBIN,set_int_lev,0x0e); /* Usa IRQ2 */
    pokeb(END_IBIN,timer_global,0x05); /* Usa o contador 0 como gerador
    pokeb(END_IBIN,count_control,0x34); de interrupções */
    pokeb(END_IBIN,r_w_counter0,(unsigned char)0x06);
    pokeb(END_IBIN,r_w_counter0,(unsigned char)0x00);
    pokeb(END_IBIN,count_control,0x74); /* Usa o contador 1 como contador
    pokeb(END_IBIN,r_w_counter1,(unsigned char)0xff); encadeado */
    pokeb(END_IBIN,r_w_counter1,(unsigned char)0xff);

    /* Instalar a rotina de interrupção na tabela de vectores */
    disable();
    addr0A = (long)getvect(0x0A);
    setvect(0x0A,aq_int);
    enable();

    /* Activa a geração de interrupções pelo SAC */
    pokeb(END_IBIN,clr_int,0x00);
}

```

Figura 3-2: Código do módulo Monit_instala

Módulo: Monit_calibra**Tipo:**

Procedimento (processamento).

Funções:

Promover a auto-calibração do sistema Keithley.

Subordinados:**Dependência:**

É chamada na Inicialização da Exploração On-line.

Interfaces:

São usadas as definições constantes do header kh500.h.

Processamento:

Seleccção do modo de calibração e início do processo.

Aguarda até fim do processo de calibração.

Dados:

Variáveis auxiliares.

Restrições:

A auto-calibração deve ser executada antes de se iniciar a utilização do SAC.

Procedimento específico do SAC Keithley 500.

O SAC Keithley 500 exige este procedimento de auto-calibração pelo que foi individualizado este módulo. Sendo usado um SAC que não o exija, a rotina deve existir, mesmo que vazia.

Módulo: Monit_desinstala

Tipo:

Procedimento (processamento).

Funções:

Terminar o processo em background de Monitorização desinstalando a rotina de serviço às interrupções.

Subordinados:

Dependência:

É chamado pelo módulo Terminação da Exploração On-line.

Interfaces:

Usa a variável endereço OA.

Processamento:

Desactiva interrupções

Repõe o conteúdo na posição 0xOA da tabela de vectores com o valor existente antes da instalação de Monitorização e guardado na variável endereço_OA.

Activa interrupções.

Dados:

Não usa dados internos

Restrições:

Realiza o processamento complementar ao de Monit_instala.

Módulo: Monitorização**Tipo:**

Rotina de serviço a interrupções (processamento/entrada)

Funções:

Implementa as funções de aquisição de dados do processo e de identificação de situações de alarme.

Executa a leitura do tempo e do valor aos canais do SAC configurados como de entrada analógica: selecções do canal, conversão A/D, leitura e inserção na fila circular respectiva.

A selecção do canal inclui a especificação do ganho, da filtragem e da gama de leitura.

As leituras são efectuadas à frequência especificada para cada canal, múltipla da frequência das interrupções.

O valor da leitura do canal pode ser calibrado ($= K1 + K2 \times \text{Valor}$) antes de ser guardado na fila circular do buffer Tabela.

Os valores adquiridos são testados em relação aos limites inferiores e superiores especificados para o canal, gerando sinais de alarme caso sejam excedidos.

Subordinados:**Dependência:**

Uma vez instalada na Inicialização (Monit_instala), é activada por hardware (Int 0x0A-IRQ2), até ser desinstalada na Terminação (Monit_desinstala).

Interfaces:

Associado a um canal do SAC, correspondente a uma variável do processo, existe o seguinte conjunto de vectores que configura a Monitorização e fazem parte da estrutura do buffer cnf_variáveis:

— frequência — contêm a frequência de leitura do canal traduzida num número inteiro que indica o número de varrimentos que reparam cada leitura.

— par_M2A — contêm oito bits que parametrizam o funcionamento do SAC para a leitura do canal. No Keithley é usado para seleccionar o filtro, o ganho local e o modo de aquisição de dados do canal.

— par_M2B — contêm oito bits que parametrizam o funcionamento do SAC para a leitura do canal. No Keithley é usado para seleccionar o módulo de hardware no SAC, ganho global e gama de trabalho.

— end_módulo — endereçamento do canal no SAC.

- nr. canais — guarda a sequência dos canais (menos internos), configurados para leitura.
- contador — vai registando para cada canal o número de varrimentos ao SAC já produzidos desde a última leitura. Quando esse número atingir a frequência, uma nova leitura deve efectuar-se.
- K1 e K2 — são respectivamente os parâmetros de calibração do valor da leitura do canal ($K1 + K2 \times \text{Valor}$).
- alarm_sup e alarm_inf — estabelecem os limites para os valores aceitáveis como normais.

Os dados adquiridos são colocados na buffer Tabela e as situações de alarme eventualmente identificadas são sinalizadas no buffer Lista_alarmes.

Processamento:

Suspende a geração de interrupções pelo SAC. Lê o tempo actual em décimos de segundo e actualiza o canal Tempo (canal 0). Decrementa o contador do canal actual. Se o contador atingir o valor 0:

- Repõe o contador do canal actual na frequência configurada.
- Aguarda até à última conversão (canal 0) ser concluída.
- Prepara leitura do conversor e inserção na fila circular.
- Lê o valor do conversor
- Efectua as operações de calibração e insere na fila circular o valor da leitura do canal 0 e respectivo tempo de leitura.
- Actualiza o canal Tempo de Simulação com o tempo de leitura.
- Inicia a conversão para a leitura do canal 1.
- Efectua as selecções para o canal actual, o canal 2, a partir do conjunto de canais configurados para leitura.
- Actualiza os apontadores para os canais em processo de leitura.

Senão:

- Incrementa o canal actual.
- Se ultrapassar o número total de canais o canal actual volta a ser o primeiro.
- Activa geração de interrupções pelo SAC.
- Executa alguma função que existisse na posição 0x0A da tabela de vectores.

A rotina termina a interrupção passando o processamento para a instrução interrompida.

Dados:

São usadas variáveis auxiliares para identificar os canais em processo de leitura (canal 0, canal 1 e canal 2) e para o número do canal_actual.

Restrições:

O módulo é específico para o sistema de aquisição Keythley 500. Neste SAC a aquisição de dados de um canal é executada em dois tempos, selecção e acondicionamento com até 0.8 μ s de settling time e conversão em 20 μ s. Todas operações sobre o SAC têm settling time que têm que ser respeitados.

Este módulo deve ser muito eficiente, sendo executado no mínimo tempo possível.

Para garantir a melhor utilização do recurso tempo, deve procurar-se paralelizar as operações no SAC, implementando um piping de tarefas realmente simultaneas. Assim enquanto o canal servido no tempo 0, (canal 0) é lido do buffer de saída do conversor, o canal servido no tempo 1 (canal 1) está a ser convertido e o canal que está a ser servido no tempo actual (canal 2), está a ser seleccionado e acondicionado.

Em resposta às interrupções geradas pelo SAC é executado o módulo Monitorização codificado pela rotina Monit_int. Em cada interrupção é atendido o próximo canal configurado para leitura sendo ainda processados, no piping implementado, os últimos dois canais atendidos.

A frequência mínima real é a frequência de base de geração de interrupções a dividir pelo número de canais configurados para leitura. As leituras são necessariamente desfazadas no tempo.

PROCEDIMENTO
 MONIT_INT - interrupção para aquisição periódica de dados e monitorização de alarmes.

PARÂMETROS

NOTAS

ULTIMA ALTERAÇÃO

```
void interrupt monit_int()
{
    unsigned static char    canal0=0,canal1=0,canal2=0,bloco=0;
```

```

/*Inibe geração de interrupções pela placa*/
pokeb(END_IBIN,clr_int,0x00);

/* Actualiza canal Tempo */
if(frente[0] >= (tabela[0]+tamanho[0]-1))
{
    frente[0]=tabela[0];
}
else
{
    (frente[0])++;
}
movedata(0,0x46c,FP_SEG(&tempo_sys),FP_OFF(&tempo_sys),4);
frente[0]->valor = (tempo_sys - tempo_inicial)/(double)1.82;

/* Decrementa e atende aos canais cujos contadores atinjam zero */
if(--(contador[bloco]) <= 0)
{
    /*Repõe o contador do canal actual*/
    contador[bloco]=frequência[bloco];

    /* Testar fim de conversão a/d - canal0
    pokeb(END_IBIN,CMDB_M2,(par_M2B[canal0] & 0xef));
    do{
    }while((peek(END_IBIN,CMDA_M2) & 0x40) == 0x40);

    /*Repôr modo de leitura*/
    pokeb(END_IBIN,CMDB_M2,par_M2B[canal0]);

    /* Prepara inserção da leitura na fila circular do canal actual */
    if(frente[canal0] >= (tabela[canal0]+tamanho[canal0]-1))
        frente[canal0]=tabela[canal0];
    else
        (frente[canal0])++;

    /* Leitura do conversor - canal0 (valor e tempo) e canal tempo */
    movedata(END_IBIN,CMDA_M2,FP_SEG(valor_temp),FP_OFF(valor_temp),2);
    frente[canal0]->valor = k1[canal0] * (double)valor_temp + k2[canal0];
    frente[canal0]->tempo = frente[0]->valor;

    /*----- Monitoriza situações de alarme -----*/
    if ((frente[canal0]->valor > var_conf[canal0].alarm_sup) || (frente[canal0]->valor <
var_conf[canal0].alarm_inf))
    {
        if (alarm_canal[canal0]==0)
        {
            alarm_canal[canal0]=1;
            lista_almes[nr_almes].sobdesc='S';
            lista_almes[nr_almes].canal=canal0;
            lista_almes[nr_almes].tempo= frente[0]->valor;;
            lista_almes[nr_almes].validado='N';
            nr_almes++;
            if (nr_almes==nr_limite_almes)
                nr_almes=0;
        }
    }
    if (alarm_canal[canal0]==1 && frente[canal0]->valor <= var_conf[canal0].alarm_sup
&& frente[canal0]->valor >= var_conf[canal0].alarm_inf)

```

```

    {
        alarm_canal[canal0]=0;
        alarm_flag[canal0]=0;
        lista_almarmes[nr_almarmes].sobdesc='D';
        lista_almarmes[nr_almarmes].canal=canal0;
        lista_almarmes[nr_almarmes].tempo = frente[0]->valor;
        lista_almarmes[nr_almarmes].validado='N';
        nr_almarmes++;
        if (nr_almarmes==nr_limite_almarmes)
            nr_almarmes=0;
    }
    /*-----*/

    /* Inicia a conversão - canal1 */
    pokeb(END_IBIN,CMDD_M2,0xff);

    /* Actualização do canal a Seleccionar - canal2 */
    canal2=nr_canais[bloco];

    /* Selecciona localmente o modo de funcionamento o ganho e a filtragem */
    pokeb(END_IBIN,CMDA_M2,par_M2A[canal2]);

    /* Selecciona o módulo AIM e o canal a usar */
    pokeb(END_IBIN,end_modulo[canal2],canal2);

    /* Selecciona a slot, a gama de trabalho e o ganho global */
    pokeb(END_IBIN,CMDB_M2,par_M2B[canal2]);

    /* Actualização dos canais em processo de conversão */
    canal0=canal1;
    canal1=canal2;
}
bloco++;
if(bloco >= totalcanais)
    bloco=0;

/* Clear PIC - retoma geração de interrupções pela placa */
outportb(0x20,0x20);

/* Asm pushf CALL FAR CS:[&addr0A] executa alguma função existente na tabela de
vectors*/
__emit__(0x2E, 0x1EFF, &addr0A);
}

```

Figura 3-3: Código do módulo Monitorização — monit_Int ()

3.3 Documentação do código

Propõe-se uma documentação das rotinas escritas em linguagem C baseada nos seguintes blocos de comentários de acordo com o proposto por Auslander [48].

A incluir no início de cada ficheiro:

```

/*****
FICHEIRO

ROTINAS PÚBLICAS

ROTINAS PRIVADAS

NOTAS

ALTERAÇÕES

*****/
/*****
DECLARAÇÕES GLOBAIS
    Declarações de outros módulos necessárias neste.
*****/

/*****
DADOS PÚBLICOS
    Variáveis e declarações externas visíveis a todos as rotinas deste módulo.
*****/

/*****
DADOS PRIVADOS
    Variáveis e declarações visíveis a todos as rotinas deste módulo.
    Se forem static não são visíveis fora deste módulo.
*****/

/*****
ROTINAS DE ENTRADA
    Rotinas que podem ser chamadas de outros módulos.
*****/

```

Para cada rotina de entrada:

```

/*****
PROCEDIMENTO

PARÂMETROS

NOTAS

ULTIMA ALTERAÇÃO

*****/
/*****
ROTINAS PRIVADAS
    Rotinas para uso exclusivo neste módulo. Não são visíveis fora.
*****/

```

Para cada rotina privada:

```

/*****
PROCEDIMENTO
PARÂMETROS
NOTAS
ULTIMA ALTERAÇÃO
*****/
    
```

A documentação também incluiu comentários ao longo do código das rotinas.

CAPÍTULO 4

Teste e Validação: Aplicação prática

Foi preparado um problema com resolução proposta numa óptica de caso de estudo que, por um lado, permite uma introdução gradual à aplicação SProcess e, por outro, oferece uma forma de testar, com algum nível de realismo, o funcionamento da aplicação, validando os seus resultados e identificando pontos fracos. O processo físico e a sua interface com o sistema são simulados por software.

4.1 - Caso de Estudo — Tanque de retenção com nível constante.

Descrição

Pretende-se que um tanque de retenção de água (figura 4-1) seja mantido a um nível constante (H_{sp}), independentemente de variações do caudal de entrada (Q_{in}).

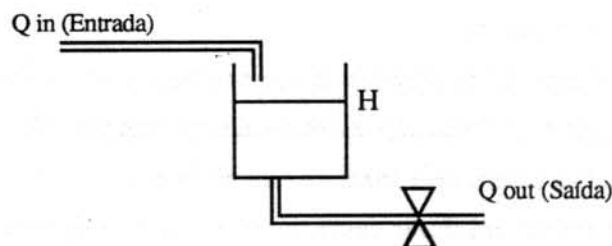


Figura 4-1: Apresentação esquemática do tanque de retenção

Propõe-se uma estratégia de controlo por *feedback*. Neste tipo de estratégia de controlo, a variável do processo a ser controlada é medida e esta medida usada para ajustar outra variável do processo que possa ser manipulada. No presente estudo, a variável medida é o nível do tanque, H , a variável manipulada a abertura da válvula, K_T (directamente relacionada com o caudal de saída, Q_{out}) e a perturbação o caudal de entrada, Q_{in} . O esquema de controlo é implementado por um controlador PI (proporcional-integral) que actua numa válvula na tubagem de descarga do tanque.

O comportamento do processo é dependente dos valores escolhidos para os parâmetros do controlador (constantes proporcional, P , e integral, I).

Várias situações podem ser observadas dada uma perturbação no caudal de entrada:

- i. Se nenhum mecanismo de controlo for utilizado ($P=0$, $I=0$) o processo tenderá para um estado dependente das iniciais e do valor da perturbação.
- ii. Se introduzirmos controlo proporcional ($P \neq 0$, $I=0$) a resposta do sistema é proporcional ao Erro (o nível desejado menos o nível actual), mas mantendo uma diferença designada *offset*.
- iii. A introdução da acção integral ($I \neq 0$) eliminará o *offset*.

Será assim interessante analisar várias situações de controlo:

- 1ª Situação: Arranque da instalação com os parâmetros do controlador: $P=1$ e $I=0.6$
- 2ª Situação: Arranque sem mecanismo de controlo: $P=0$ e $I=0$
- 3ª Situação: Evoluir para controlo proporcional: $P=1$, $I=0$
- 4ª Situação: Evoluir para controlo proporcional-integral: $P=1$, $I=0.6$
- 5ª Situação: Evoluir para controlo proporcional-integral com valores demasiado elevados para os parâmetros de controlo: $P=50$, $I=6$

Objectivos

Pretende-se usar a aplicação SProcess para acompanhar o comportamento do processo físico descrito, tanque de retenção.

Recorre-se a uma simulação do processo e da sua interface com o sistema (aquisição e actuação) por um programa de simulação, especificado no capítulo 4.2.

No arranque do processo o tanque está vazio e a válvula aberta.

A execução do processo deverá durar cerca de 10 minutos e deverá incluir as cinco situações mencionadas anteriormente. A aplicação deve ser configurada, para permitir uma supervisão do comportamento do processo por intermédio de quadros sinópticos, gráficos, tabelas, alarmes ao operador, etc. No fim da execução do processo será feito um relatório que inclua a elaboração de gráficos e a determinação de parâmetros estatísticos, bem como o cálculo da quantidade de água saída durante o tempo de execução.

Metodologia de resolução

1. Configuração das variáveis do sistema

Deve ser configurado um conjunto de variáveis que caracterizam a aplicação na globalidade.

Para tal seleccionar Configurações/Variáveis Sistema e especificar conforme o proposto:

VARIÁVEIS GLOBAIS	
NR. CANAIS	: 9
SIMULAÇÃO	:
PASSWORD 1	:
PASSWORD 2	:
FREQUÊNCIA	: 1

Figura 4-2: Écran de configuração das Variáveis de Sistema

2. Configuração das variáveis do processo

A cada variável deve ser atribuído um canal na aplicação.

Variáveis de saída (output):

Parâmetros do controlador, P (proporcional) e I (integral) com valores fixados de acordo com a situação em simulação:

1ª situação — arranque (2 m): Psp=1, Isp=0.6

2ª situação — arranque (2 min): Psp=0, Isp=0

3ª situação (2 min): Psp=1, Isp=0

4ª situação (2 min): Psp=1, Isp=0.6

5ª situação (2 min): Psp=50, Isp=6

Hsp - valor do set point do controlador estabelecido e alterável pela aplicação.

Valor de set point = 5 m.

Variáveis de entrada (input):

H - nível do tanque

H máximo = 12 m

Hsp = 5 m

Alarme superior = 5.5 m

Alarme inferior = 4.5 m.

Q_{in} - caudal de entrada

Valor aleatório entre 0 e 3 m³/s

$Q_{in\ sp}$ = 2 m³/s

Q_{out} - caudal de saída

Kt - valor proporcional à abertura da válvula

Toma valores entre 2 (fechado) e 0 (aberto).

Erro - diferença entre H e Hsp no comparador.

Toma valores entre Hsp e (12-Hsp) para Hsp=5

Toma valores entre 5 e 7 m.

Seleccionando a opção Configurações/Variáveis, configurar o canal correspondente a cada variável conforme o proposto a seguir:

CANAL 0

NOME	: Tempo
TIPO	: I
ENDEREÇO	:
TAMANHO FILA	: 10
FREQUÊNCIA	: 0
ARQUIVAR (S/N)	: N
ALARME SUP.	: 0
ALARME INF.	: 0
SET POINT	: 0

Figura 4-3: Écran de configuração do canal 0

CANAL 1	
NOME	: H (nível)
TIPO	: I
ENDEREÇO	:
TAMANHO FILA	: 500
FREQUÊNCIA	: 10
ARQUIVAR (S/N)	: S
ALARME SUP.	: 550
ALARME INF.	: 450
SET POINT	: 500

Figura 4-4: Écran de configuração do canal 1

Os restantes canais devem ser configurados nos écrans respectivos de acordo com o proposto:

CANAL	2	3	4	5	6	7	8
NOME:	Qout	Kt (válvula)	Erro (H-Hsp)	Hsp	P	I	Qin
TIPO	I	I	I	O	O	O	I
ENDEREÇO							
TAMANHO FILA	100	20	500	1	1	1	20
FREQUÊNCIA	60	80	100	1000	1000	1000	80
ARQUIVAR (S/N)	S	N	N	N	N	N	N
ALARME SUP.	30	50	50	5.5	2	10	3
ALARME INF.	0	20	- 50	4.5	0	0	1
SET POINT	2	30	0	5	1	0,6	3

Figura 4-5: Configurações para os canais 2 a 8

3. Configuração de um quadro sinóptico

Seleccionando a opção Configurações/Sinópticos efectuar o desenho do écran (ficheiro Écran 1.scr) e a definição de parâmetros (ficheiro Écran 1. par), conforme o proposto nas figuras 4-6 e 4-7, respectivamente.

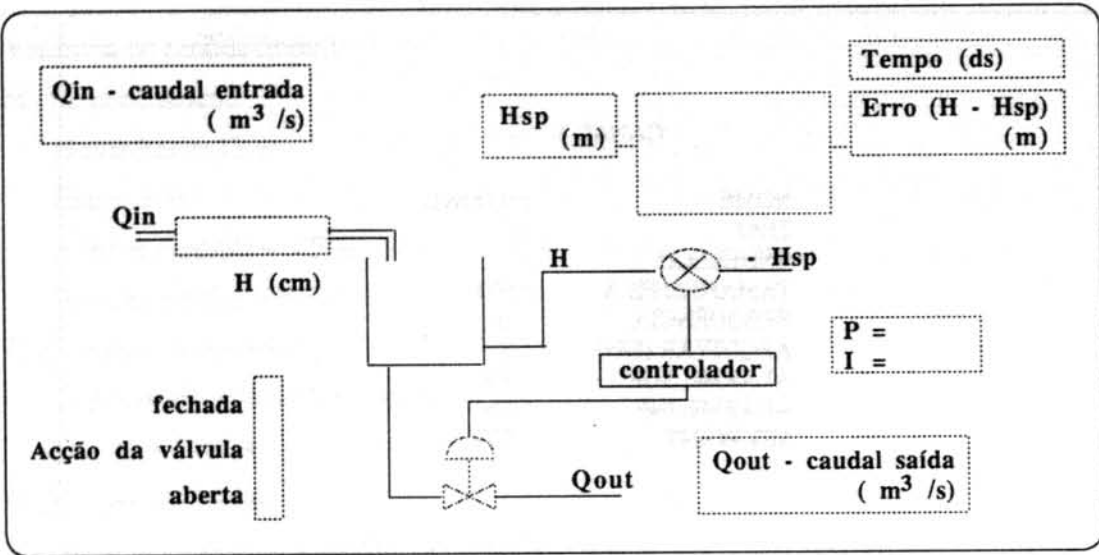


Figura 4-6. Desenho do quadro sinóptico.

```
Ficheiro C:\SP\PROCESS\PAR\ecran1.PRO
Num=1 Posx=570 Posy=10 Color=63 Type=r Format=n Op_Format=22;
Num=2 Posx=185 Posy=140 Color=59 Type=r Format=t Op_Format=42;
Graph=b Type=n Style=2 Rect=(211,116,291,187)
Extreme_Values(0.000000,0.000000,1.000000,1000.000000)
Num=3 Posx=427 Posy=246 Color=63 Type=r Format=n Op_Format=22;
Num=4 Posx=199 Posy=255 Color=63 Type=r Format=t Op_Format=0;
Graph=b Type=n Style=1 Rect=(154,192,167,259)
Extreme_Values(0.000000,0.000000,1.000000,2.000000)
Num=5 Posx=543 Posy=47 Color=63 Type=r Format=t Op_Format=0;
Graph=n Type=n Style=0 Rect=(366,27,467,88)
Extreme_Values(0.000000,-1000.000000,15.000000,1000.000000)
Num=6 Posx=300 Posy=44 Color=63 Type=r Format=n Op_Format=0;
Num=7 Posx=502 Posy=166 Color=63 Type=r Format=n Op_Format=22;
Num=8 Posx=502 Posy=183 Color=63 Type=r Format=n Op_Format=22;
Num=9 Posx=74 Posy=40 Color=63 Type=r Format=t Op_Format=0.
Graph=b Type=n Style=1 Rect=(123,89,199,109)
Extreme_Values(0.000000,0.000000,20.000000,10.000000)
```

Figura 4-7. Definição dos parâmetros do sinóptico.

O conjunto de parâmetros a definir e colocar no écran desenhado corresponde a algumas das variáveis já especificadas:

Parâmetro 1 - canal Ø (Tempo actual)

Parâmetro 2 - canal 1 (H)

Parâmetro 3 - canal 2 (Q_{out})

Parâmetro 4 - canal 3 (Kt)

Parâmetro 5 - canal 4 (Erro)

Parâmetro 6 - canal 5 (Hsp)

Parâmetro 7 - canal 1 (P)

Parâmetro 8 - canal 1 (I)

Parâmetro 9 - canal 1 (Q_{in})

4. Configuração de gráficos para análise do processo.

Pretende-se configurar dois gráficos de acompanhamento da evolução do processo, a serem chamados durante a exploração on-line.

Seleccionando a opção Configurar/Gráficos:

1º gráfico: Evolução do nível do tanque (H) com o tempo.

X1	0	COR FRAME	8	LINHA Nº: 1	
Y2	0	COR BORDER	7	LEGENDA	Canal 1
X2	639	COR JANELA	0	COR	9
Y2	479	COR TÍTULO	5	SÍMBOLO	-1
NTICKX	10	COR SUBTÍTULO	4	PADRÃO	-1
NTICKY	12	COR ABCISSAS	2	NR. DE PONTOS	500
Nº LINHAS	1	COR LEGENDA	1	CANAL	1
GRID X	0	COR ORDENADAS	2		
GRID Y	1	COR VALORES	15		
FLAG LEGENDA	1	COR GRID	2		
X MIN	0				
X MAX	100	TÍTULO		Nível do Tanque	
Y MIN	0	SUBTÍTULO			
Y MAX	1200	ABCISSAS		Tempo (dec. seg.)	
Nº CARAC.LEG.	8	ORDENADAS		H (cm)	
Nº CARAC.TIT.	30	FORMULA			
Nº CAR.SUBT.	30				
Nº CAR.ABCIS.	20				
Nº CAR.ORDEN.	24				
FONTE	1				

Figura 4-8: Écran de configuração do gráfico 1

2º gráfico: Evolução do Erro ($E = H - H_{sp}$) com o tempo.

X1	0	COR FRAME	8	LINHA Nº: 1	
Y2	0	COR BORDER	7	LEGENDA	Canal 4
X2	639	COR JANELA	0	COR	11
Y2	429	COR TÍTULO	5	SÍMBOLO	-1
NTICKX	10	COR SUBTÍTULO	4	PADRÃO	-1
NTICKY	10	COR ABCISSAS	2	NR. DE PONTOS	500
Nº LINHAS	1	COR LEGENDA	1	CANAL	4
GRID X	1	COR ORDENADAS	2		
GRID Y	1	COR VALORES	15		
FLAG LEGENDA	1	COR GRID	2		
X MIN	0				
X MAX	100	TÍTULO		Desvio do Nível ao Set Point	
Y MIN	-600	SUBTÍTULO		Erro = $H - H_{sp}$	
Y MAX	600	ABCISSAS		Tempo (dec. seg.)	
Nº CARAC.LEG.	8	ORDENADAS		Erro (cm)	
Nº CARAC.TIT.	30	FORMULA			
Nº CAR.SUBT.	30				
Nº CAR.ABCIS.	20				
Nº CAR.ORDEN.	24				
FONTE	1				

Figura 4-9: Écran de configuração do gráfico 2

5. Configuração de tabelas de dados do processo.

Construção de uma tabela, a ser chamada durante a exploração on-line, que inclua as seguintes variáveis: Q_{in} , Q_{out} , H e Erro.

Seleccionar a opção Configurar/Tabelas e especificar conforme o proposto:

NR. DE PONTOS	: 10
ACTUALIZAÇÃO (0-1)	: 0
1ª COLUNA	: 8
2ª COLUNA	: 3
3ª COLUNA	: 1
4ª COLUNA	: 5

Figura 4-10: Écran de configuração da tabela 1

6. Validação e confirmação das configurações efectuadas.

Seleccionar a opção Configurações/Validar. Caso seja detectada alguma incorrecção, actuar em conformidade.

7. Execução On-Line

Selecione a opção Hardware/Tanque que define a Monitorização por simulação do processo tanque de retenção.

Selecione a opção On-Line para iniciar o acompanhamento do processo.

8. Ajustar Configurações

Dispondo dos primeiros dados recolhidos verifica-se o funcionamento das análises, gráficos e tabelas e do comportamento das variáveis. Caso seja necessário efectuar ajustes, usar as opções de configuração disponíveis no processamento *on-line*.

9. Acompanhar a evolução do processo durante cerca de 10 minutos.

9.1. Arranque da instalação com os parâmetros do controlador estabelecidos em $P=1$ e $I=0.6$.

Tempo entre 0 e 1200 dec.sec. (décimos de segundo).

Visualizar o Sinóptico geral do processo, Écran 1, acompanhando a instabilidade associada ao arranque do processo.

Chamar o Gráfico 1 para ver em detalhe os últimos valores do nível do tanque (figura 4-11) e o Gráfico 2, para visualizar a evolução do erro (H-Hsp).

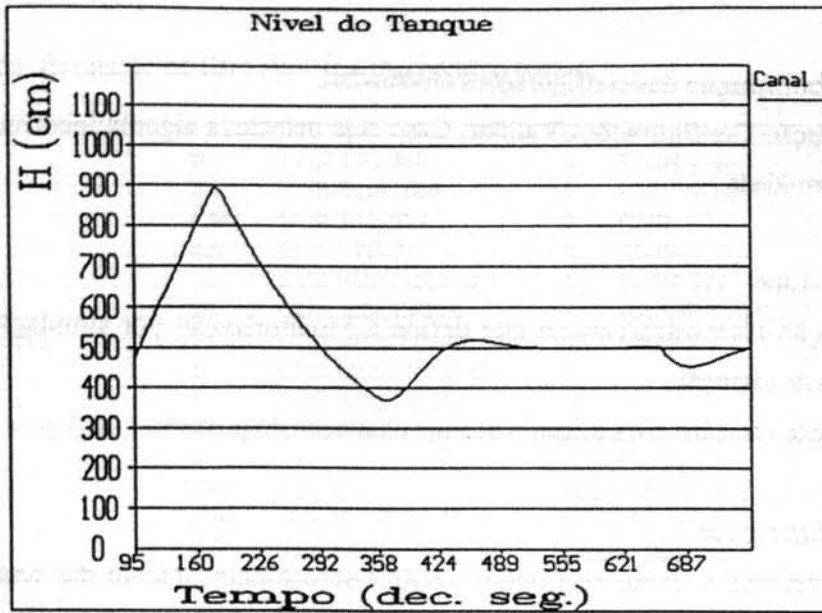


Figura 4-11: Evolução do Nível do Tanque ($P=1, I=0.6$)

Quando o valor do Erro estabilizar próximo do zero (± 50 cm.), alterar as escalas (seleccionar opção Configurações/Gráficos) de modo a poder ver em pormenor as alterações existentes. Fazer as seguintes alterações:

Gráfico 1 (H vs. Tempo)

Y max = 600 cm

Gráfico 2 (Erro vs. Tempo)

X max = -125 cm

Y max = +125 cm

Visualizar os alarmes ocorridos, verificando que estão associados à instabilidade transitória do arranque do processo. Assim sendo, proceder à validação dos alarmes que os removerá da lista de situações de alarmes pendentes.

Repare que a instabilidade verificada se deveu à acção do controlador PI que sendo função do Erro ($H-H_{sp}$) reflecte os valores que este toma, e como no início do processo, o tanque está vazio, $H=0$, a diferença $H-H_{sp}$ é em valor absoluto elevada.

Assim iremos repetir o arranque do processo sem acção de controlo ($P=0$ e $I=0$), evoluindo depois para diferentes situações de controlo automático.

Selecione a opção Stop para terminar a exploração on-line.

Reponha os valores das escalas dos gráficos seleccionando Configurar/Variáveis e alterando os seguintes valores:

Gráfico 1 (H vs. Tempo)

Y max = 1200 cm

Gráfico 2 (Erro vs. Tempo)

X max = - 600 cm

Y max = 600 cm

Antes de reiniciar o processo proceda à conversão dos ficheiros de dados dos canais 1 e 2, seleccionando a opção Off-Line/Conversão e indicando sucessivamente o número dos canais a converter. Os ficheiros resultantes da conversão serão canal 1.txt e canal 2.txt.

Abandonar a aplicação e no DOS alterar os nomes para, por exemplo canal 1-1.txt e canal 1-2.txt, respectivamente, já que se trata dos valores dos canais 1 e 2 para a primeira situação em análise.

9.2. Arranque da instalação sem mecanismos de controlo (P=0, I=0)

Tempo entre 0 e 1200 ds

Visualizar o sinóptico geral do processo, Écran 1, acompanhando a evolução do nível do tanque que se fixa em valores apenas função do caudal de entrada.

Repare nas situações de alarme que deverão ocorrer já que não há nenhuma acção de controlo do nível do tanque.

Revisualize a evolução recente do Erro (H-Hsp), chamando o Gráfico 2, até este atingir valores relativamente próximos do zero (± 100 cm).

Vamos então introduzir um primeiro tipo de acção de controlo sobre o nível do tanque. Mas antes vamos validar os alarmes já ocorridos.

9.3. Evoluir para controlo proporcional : P=1, I=0

Tempo entre 1200 e 2400 ds

Modificar o valor da variável de saída, P, seleccionando Configurar/Variáveis, e atribuir um (1) ao valor de setpoint para P.

Visualizar no sinóptico Écran 1, a forma como o nível do tanque (H) se aproxima do seu setpoint (Hsp).

Reparar no comportamento do nível quando há perturbações no caudal de estrada (Q in). O nível aproxima-se do seu valor de setpoint, mas sem se fixar no valor exacto. Esta diferença, offset, é uma característica deste tipo de acção de controlo (Proporcional).

Verifique os sucessivos valores de offset chamando os Gráficos 1 e 2 e a Tabela 1 que mostram com mais rigor os valores em causa (figura 4-12).

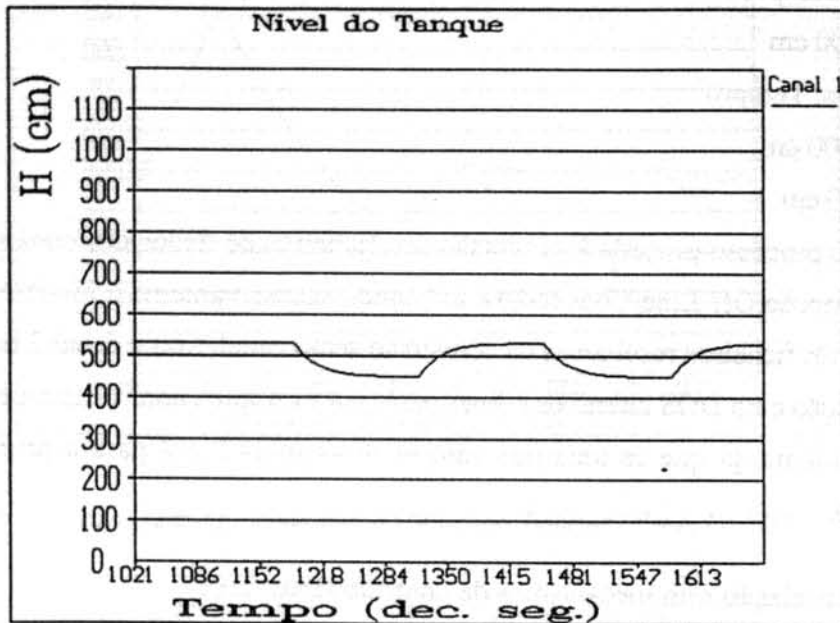


Figura: 4-12: Evolução do Nível do Tanque ($P=1$, $I=0$)

9.4. Evoluir para controlo proporcional-integral: $P=1$, $I=0.6$

Tempo entre 2400 e 3600 ds

Modificar o valor da variável de saída, I, seleccionando Configurar/Variáveis, e atribuir (0.6) ao valor de setpoint para I.

Visualizar no sinóptico Écran 1, a forma como o nível do tanque (H) se aproxima do seu setpoint (H_{sp}).

Reparar no comportamento do nível quando há perturbações no caudal de entrada (Q_{in}).

O nível do tanque atinge o seu setpoint estabilizando nesse valor (± 50 cm).

Estamos numa situação em que acção de controlo assegura uma resposta rápida às perturbações (acção proporcional) e essa resposta faz a variável controlada atingir de facto o nível desejado, setpoint (acção integral).

Esta é uma boa especificação dos valores dos parâmetros P e I do controlador, mas outros valores podem provocar situações diversas.

9.5. Controlo proporcional-integral com valores demasiado elevados para os parâmetros do controlador: $P=50$, $I=6$

Tempo entre 3600 e 4800 ds

Modificar o valor das variáveis de saída, P e I, seleccionando Configurar/Variáveis, e atribuir cinquenta(50) ao valor de setpoint para P e seis(6) para o valor de setpoint para I. Visualizar no sinóptico Écran 1, a forma como o nível do tanque (H) se aproxima do seu setpoint (Hsp).

Reparar no comportamento do nível quando há perturbações no caudal de estrada (Q in). A acção de controlo força um abrir e fechar da válvula constante que cria instabilidade no nível do tanque (H) e também no caudal de saída (Q out).

Verificar os valores destas variáveis chamando os Gráficos 1 e 2 e a Tabela 1 que permitem observar os valores tomados com mais rigor.

Visualizar eventuais alarmes que tenham ocorrido.

10. Fim do processamento *on-line*.

Terminar a exploração *on-line*, seleccionando a opção Stop.

11. Análises Off-Line

As análises *off-line* incluem a exportação de dados para o Lotus e a produção de um relatório de desenvolvimento do processo.

11.1. Exportar os dados recolhidos das variáveis H e Erro para ficheiros em formato compatível com outras aplicações (ex: Lotus).

Seleccionar a opção Off-Line/Conversão e especificar o número do canal a converter.

À variável H corresponde o canal 1.

À variável Q out corresponde o canal 2.

Aguardar o fim do processo de conversão que produz o primeiro ficheiro ASCII canal 1.txt. Executar a conversão da segunda variável que produz o ficheiro canal 2.txt.

11.2. Análise da capacidade do mecanismo de controlo (parâmetros do controlador) pela observação da evolução do nível do tanque.

Seleccionar a opção Lotus que executa a aplicação.

i. Carregar o ficheiro de dados canal 1.txt (importar como ASCII, ficheiro de texto) e dividi-lo em quatro partes correspondentes a cada situação de controlo simulada. Os dados

importados para a folha de cálculo têm que ser convertidos para um formato numérico. Carregar também o ficheiro canal 1-1.txt.

ii. Elaborar gráfico H versus tempo, especificando os valores de *set point* e alarmes inferior e superior para cada uma das situações simuladas.

iii. Cálculo de parâmetros estatísticos que reflitam a capacidade do mecanismo de controlo: análise dos desvios médios do nível do tanque em relação ao valor de *set point* (Média das diferenças absolutas entre o valor da grandeza e o seu valor de *set point*) durante o tempo de execução. Observar diferentes comportamentos entre as cinco situações simuladas comparando os valores obtidos.

11.3. Cálculo do volume de água saído do tanque durante o tempo de execução:

i. Carregar o ficheiro de dados canal 2.txt que corresponde ao Q_{out} .

ii. A partir dos valores de Tempo (t) e Q_{out} e calcular para cada dois valores consecutivos (i-1, i) as seguintes grandezas:

$$\Delta t = t_i - t_{i-1} \quad \text{e} \quad Q_{out \text{ médio}} = (Q_{out, i} + Q_{out, i-1}) / 2,$$

iii. Calcular o volume de água ($V_{água}$) saído do tanque durante o tempo de execução, de acordo com a seguinte fórmula:

$$V_{água} = \sum (Q_{out \text{ médio}} * \Delta t)$$

4.2 Simulação do processo e da interface.

A necessidade de validar o desenvolvimento deste tipo de software dependente de hardware externo e mesmo do funcionamento de um processo real, motivou a concepção de um módulo de simulação do processo e da interface.

A função de Monitorização de dados do processo deve poder ser substituída por uma função de simulação do processo e do sistema de aquisição de dados, assegurando a colocação periódica de dados na Tabela e a monitorização de situações de alarme.

O processo simulação deverá ser executado em background e em paralelo com o Registo.

A simulação é importante não só como instrumento de teste e validação do software em desenvolvimento mas também como forma de facilitar a utilização da aplicação em ambiente de formação e treino em software tipo SCADA.

A função pode ser desenvolvida no sentido de a dotar de um maior nível de realismo baseando a simulação em modelos matemáticos de processos físico-químicos reais.

Pretendeu-se desenvolver duas versões, uma que gere valores aleatórios para as leituras de cada canal, e outra, mais realista, irá gerar valores para um processo de nivelamento de um tanque de retenção de água (ver figura 4-1) com base num modelo de 1ª ordem e com controlador PI à válvula da tubagem de saída.

Apresenta-se, aqui uma especificação geral do módulo de simulação que embora não fizesse parte dos objectivos deste trabalho se revelou de grande utilidade.

Módulo: Simulação

Tipo:

Procedimento de serviço a interrupções.

Funções:

Substituir a função Monitorização. Os dados do processo são gerados de acordo com modelos matemáticos. Simula, portanto, o funcionamento do Processo e da interface de aquisição e controlo.

Subordinados:

Aleatório - simulação aleatória ou Tanque - simulação baseada num modelo.

Dependência:

Com a aplicação configurada para simulação, o procedimento é iniciado por Sim_instala na Inicialização e termina com Sim_desinstala na Terminação da Exploração On-line.

É executado em resposta a interrupções (0x1c), gerados pelo relógio do computador.

Interfaces:

São usados os buffers Tabela, Cnf_Variáveis e Lista_alarmes, como forma de comunicar com os outros processos da aplicação.

Processamento:

Configura a simulação de acordo com as variáveis do sistema e processo
Percorre os vários canais configurados.

Se o contador do canal estiver em zero:

Inicializa o contador com a frequência da variável.

Calcula um novo valor para o canal (Chama Aleatório ou Tanque).

Insere a leitura (Valor e Tempo) na fila circular correspondente.

Se o Valor da leitura estiver fora dos limites dos alarmes.

Sinaliza um alarme.

Senão decrementa a frequência de leitura da variável.

Dados:

Restrições:

Executa todas as funções do processo de Monitorização.

CAPÍTULO 5

Conclusões

Objectivos propostos

Pretendeu-se, com o trabalho realizado, estudar e projectar um sistema de *software* interactivo e configurável a diferentes situações de aplicação, na área da Supervisão, Controlo e Aquisição de Dados (tipo SCADA).

O trabalho devia incluir a análise e concepção de uma aplicação que contemplasse as principais funções dos sistemas tipo SCADA. Foi também proposta a produção de uma aplicação a encarar como protótipo para uma aplicação profissional.

Estes objectivos deverão ser um contributo para o desenvolvimento a médio prazo de uma linha de produtos de *software* na área das aplicações SCADA, de uso geral.

O trabalho desenvolvido e agora apresentado foi realizado, aproximadamente, em 10 meses úteis.

Opções adoptadas

Para o desenvolvimento do projecto de *software* foram adoptadas algumas opções estratégicas baseadas na utilização futura do projecto e protótipo, nas restrições dos objectivos propostos e na adequação do sistema em projecto ao que se julgou ser as necessidades da supervisão de processos.

Das opções tomadas as julgadas mais relevantes foram:

- aplicar uma metodologia de análise e projecto estruturado com introdução de algumas técnicas para modelização de sistemas de tempo real;
- procurar conceber uma estrutura para a aplicação que incorpore, tanto quanto possível, módulos independentes, assegurando, por um lado, uma utilização parcial da aplicação e, por outro, uma substituição facilitada de módulos;

- procurar projectar um sistema aberto a desenvolvimentos;
- basear o processamento paralelo de tarefas em processamento por interrupções;
- utilizar um editor gráfico independente (Editor) e um gerador de gráficos (Graf), pré-existent, ainda que modificando-os;
- basear a utilização concorrente dos dados do processo numa estrutura de filas circulares residente em RAM;

Resultados

Os principais objectivos propostos terão sido atingidos através do trabalho realizado, nomeadamente:

- estudo de sistemas de *software*, tipo SCADA (Cap. 1), que incluiu um estudo comparativo de vários produtos disponíveis no mercado (Anexo A) e uma consulta a publicações recentes na área do *software* para aplicações em tempo real;
- especificação e análise de um protótipo de uma aplicação tipo SCADA (Cap. 2.1 e 2.3) que incluiu as principais funções que caracterizam estes sistemas. A análise do sistema foi feita tanto no domínio do fluxo de informação como no de controlo, introduzindo algumas técnicas orientadas para a modelização de sistemas de tempo real. Como resultado, foi produzido um dicionário de dados: entidades, processos e depósitos (Anexo B);
- concepção do modelo físico da aplicação em projecto (Cap. 2.3) que incluiu a definição da sua estrutura, dos módulos que a constituem, da organização de dados e do tipo de interface com o utilizador. Para cada módulo foi produzida uma ficha de especificações para codificação (Anexo C). A forma como os vários módulos são agrupados em ficheiros e estes em programas executáveis, bem como a forma de armazenar os dados, são definidos no *packaging* da aplicação (Cap. 2.4);
- codificação da aplicação protótipo, de acordo com a concepção efectuada e procurando integrar *software* existente ou *software* escrito para outros fins. Documentação do código;
- teste e validação da aplicação. Recorrendo a uma opção de simulação do processo da função de monitorização, foi possível uma experimentação prática, com um bom nível de realismo, que permitiu a validação com uma aplicação a um processo constituído por um tanque de retenção;

— iteração no ciclo de produção do *software* com revisão e ajuste do projecto e consequente impacto na codificação.

Aplicação protótipo

De acordo com o proposto, o protótipo final é um sistema de *software* escrito em Turbo C para MS-DOS, a ser executado num microcomputador tipo IBM-AT com 640 KB de RAM, disco rígido e écran gráfico EGA (mínimo).

A aplicação inclui funções de configuração *off-line* ao processo, algumas das quais de execução independente (ex. Editor Gráfico), e funções de exploração *on-line*. Estas são constituídas por dois processos de execução em paralelo e em *background*, recorrendo a processamento por interrupções que implementam a Monitorização de dados do processo com armazenamento temporário num *buffer*, em memória, e o Registo, que inclui arquivo de dados em disco a partir desse *buffer*.

Em *foreground*, no écran, podem ser executadas outras funções (Visualização de Sinópticos, Análises, Tratamento de Alarmes, Operação e Configurações), de acordo com as selecções do utilizador-operador.

Terminada a exploração *on-line*, é oferecida uma conversão dos dados para ficheiros exportáveis para aplicações de uso comum (folhas de cálculo, bases de dados, etc.).

Dificuldades

A selecção das funções consideradas no projecto corresponde a um esforço de identificação das características mais singulares dos produtos SCADA, precedida por uma abordagem exaustiva das funções que tipificam estes produtos (ver Cap. 1).

Ainda assim, as características pretendidas de interacção com o utilizador, configuração a diferentes situações, oferta de múltiplas funções, processamento em tempo real e a estrutura de dados associada, originam uma aplicação protótipo de dimensão e complexidade consideráveis que condicionam o seu projecto e desenvolvimento.

As principais dificuldades encontradas no projecto e implementação relacionam-se com o processamento concorrente à estrutura de dados, com a integração e interface com

programas que pudessem ser independentes (ex: Editor e Gerador gráfico), a concepção de uma estrutura de dados ajustável à flexibilidade de configuração pretendida para a aplicação, a estruturação da aplicação a partir do trabalho da análise funcional, as restrições impostas pelo sistema envolvente de *hardware* e *software*, não vocacionado para suportar este tipo de aplicações, a interactividade e modularidade pretendidas e a necessidade de produzir uma aplicação compacta ocupando o mínimo de RAM.

Perspectivas de Evolução

O trabalho realizado permitiu identificar áreas a exigir futuros desenvolvimentos. Assim, e dentro do objectivo global de acumular conhecimento e experiência na produção de *software* tipo SCADA, destacam-se as seguintes áreas:

- processamento de processos concorrentes;
- processamento em tempo real;
- interfaces gráficas;
- sistemas distribuídos.

No quadro dos objectivos de médio prazo estabelecidos, propõe-se que a evolução do presente trabalho envolva:

1. Utilização da aplicação protótipo em processos laboratoriais e em processos simulados, nomeadamente em ambientes de formação, promovendo alguns desenvolvimentos adicionais que essa utilização motive, refazendo as várias fases da produção da aplicação da Análise à Codificação e Teste.
2. Estudos que suportem decisões estratégicas sobre opções a tomar, nomeadamente sobre equipamento informático, sistema operativo (núcleos de tempo real e/ou ambientes multitarefa), interface gráfica, base de dados e ligação em rede, editor gráfico e gerador de gráficos.
3. Desenvolvimento, com base no trabalho já realizado, e tendo em conta as opções tomadas, de um núcleo de um sistema SCADA que contemple as principais funções, de forma a poder constituir um produto aplicável ou comerciável.

Referências e Bibliografia

Referências

- [1] Small, C. H. "Real-Time UNIX & UNIX look-alikes", EDN, Jun. 7, 1990.
- [2] Small, C. H. "Real-time Kernels scarce for single-chip μ Ps", EDN, Set. 1, 1989.
- [3] Small, C. H. "Real-Time Kernels: Sprout into Full-Grown software environments", EDN, Jul. 6, 1989.
- [4] Egas, A.P.V. e Castro, J.A.A.M. "A GKS-Based Interface for Real-Time Monitoring and Control of Chemical Processes", DEQ/Coimbra, 1989.
- [5] Dertouzos, M. L. e Mok, A. "Multiprocessor On-Line Scheduling of Hard-Real-Time Tasks", IEEE Trans. on Software Eng., Vol.15, N.12, Dez. 1989.
- [6] Dasarathy, B. "Timing Constraints of Real-Time Systems", IEEE Trans. on Software Eng., Vol.11, N.1, Jan. 1985.
- [7] Hindin, H. e Rauch-Hindin, W. "Real-Time Systems", Electronic Design, Jan. 6, 1983.
- [8] Sanden, B. "The Case for Electric Design of Real-Time Software", IEEE Trans. on Software Eng., Vol.15, N.3, Mar. 1989.
- [9] Luqi, Berzins, V. e Yeh, R.T. "A Prototyping Language for Real-Time Software", IEEE Trans. on Software Eng., Vol.14, N.10, Out. 1988.
- [10] Kligerman, E. e Stoyenko, A. "Real-Time Euclid: A Language for Reliable Real-Time Systems", IEEE Trans. on Software Eng., Vol.12, N.9, Set. 1986.
- [11] Bologna, S. e Leveson, N. "Foreword Reliability and Safety in Real-Time Systems", IEEE Trans. on Software Eng., Vol.12, N.9, Set. 1986.
- [12] Leinbaugh, D. W. e Yamini, M. R. "Guaranteed Response Time in a Distributed Hard-Real-Time Environment", IEEE Trans. on Software Eng., Vol.12, N.12, Dez. 1986.
- [13] Switzer, C. "Display Technologies for Control Applications", I & CS, Fev. 1985.
- [14] Kligerman, E. e Stoyenko, A. D. "Real-Time Euclid: A Language for Reliable Real-Time Systems", IEEE trans. on Software Eng., Vol.12, N. 9, Set. 1986.
- [15] Putnam, F. A. "Graphics for Control: Looking better all the time", I & CS, Sep 1988.

- [16] Cohen, H. N. e Hanel, J. "A Timing Independent BIOS", BYTE, 1987.
- [17] Stankovic, J. A. "Misconceptions About Real-Time Computing", IEEE Computer, Out. 1988.
- [18] Lauber, R. J. "Software for Industrial Process Control", IEEE Computer, Fev. 1984.
- [19] Mills, H. D. e Dyson, P. B. "Using Metrics to Quantify Development", IEEE Software, Mar. 1990.
- [20] Rombach, H. "Design Measurement: Some Lessons Learned", IEEE Software, Mar. 1990.
- [21] Grady, R. "Work-Product Analysis: The Philosopher's Stone of Software", IEEE Software, Mar. 1990.
- [22] Berzins, V. e Luqi "An Introduction to the Specification Language Spec", IEEE Software, Mar. 1990.
- [23] Akima, N. e Ooi, F. "Industrializing Software Development: A Japanese Approach", IEEE Software, Mar. 1989.
- [24] Adler, M. "An Algebra for Data Flow Diagram Process Decomposition", IEEE Trans. on Software Eng., Vol.14, N.2, Fev. 1988.
- [25] Deutsh, M. "Focusing Real-Time Systems Analysis on User Operations", IEEE Software, Set. 1988.
- [26] Luqi e Berzins, V. "Rapidly Prototyping Real-Time Systems", IEEE Software, Set. 1988.
- [27] Faustini, A. e Lewis, E. "Toward Real-Time Dataflow Language", IEEE Software, Jan. 1986.
- [28] Barnard, H., Metz, R. e Price, A. "A Recommended Practice for Describing Software Design: IEEE Standards Project 1016", IEEE Trans. on Software Eng., Vol.12, N.2, Fev. 1986.
- [29] Ward, P. "The Transformation Schema: An Extension of the Data Flow Diagram to Represent Control and Timing", IEEE Trans. on Software Eng., Vol.12, N.2 Fev. 1986.
- [30] Weber, H. e Ehrig, H. "Specification of Modular Systems", IEEE Trans. on Software Eng., Vol.12, N.7, Jul. 1986.
- [31] Neumann, P. "On Hierarchical Design of Computer Systems for Critical Applications", IEEE Trans. on Software Eng., Vol.12, N.9, Set. 1986.

- [32] Jahanian, F. e Mok, A. "Safety Analysis of Timing Properties in Real-Time Systems", IEEE Trans. on Software Eng., Vol.12, N.9, Set. 1986.
- [33] Auernheimer, B. e Kemmerer, R. "RT-Aslan: A Specification Language for Real-Time Systems", IEEE Trans. on Software Eng., Vol.12, N.9, Set. 1986.
- [34] Boehm, B. W. "Improving Software Productivity", IEEE Computer, Set. 1987.
- [35] Weide, B. e Brown, M. "Process Control: Intergration and Design Methodology Support", IEEE Computer, Fev. 1984.
- [36] Hoare, C. A. R. "An Overview of Some Formal Methods for Program Design", IEEE Computer, Sep 1987.
- [37] Walter, C. "Control Software Specification and Design: An Overview", IEEE Computer, Fev. 1984.
- [38] Gruia-Catalin Roman, "A Toxonomy of Current Issues in Requirements Engineering", IEEE Computer, Abr. 1985.
- [39] Curtis, J. N. e Schwieder, D. "Data Analysis through a generalized interactives computer animation method (DATICAN)" Computer & Graphics Vol.9 N. 2 pp. 153-157, 1985.
- [40] Liu, L. Y. e Shyamasundar, R. K. "Static Analysis of Real-Time Distributed Systems", IEEE Trans. on Software Eng., Vol.16, N. 4, Abr. 1990.
- [41] Chandrasekharan, M., Dasarathy, B. e Kishimoto, Z. "Requirements-Based Testing of Real-Time Systems: Modeling for Testability", IEEE Computer, Abr. 1985.
- [42] Schindler, M. "Real-time software-design tool combines three different views", Jul. 23, 1987.
- [43] Aoyama, M. e Miyamoto, K. "Design Specification in Japan: Tree-Structured Charts", IEEE Software, Mar. 1989.
- [44] Pardikar, S. e Vaidya, M. " GKS Implementation: A Case Study", Comput & Graphics, Vol.11, N. 12, pp. 75-78, 1987.
- [45] White, S. M. e Lavi, J. Z. "Embedded Computer System Requirements Workshop", IEEE Computer, Abr. 1985.
- [46] S.E. Consultants, "PC Control Software Review", Control Engineering, Out. 89, Nov. 89, Jan. 90, Fev. 90, Mar. 90, Abr. 90, Maio 90, Jun. 90, Jul. 90.
- [47] "Manual da Interface 500-Series Measurement and Control Systems", Keithley, 1989.

- [48] Auslander, D. e THAM, C. "Real-Time Software for Control", Prentice-Hall, 1989.
- [49] Hatley, D. e Pirbhai, I. "Strategies for Real-Time Systems Specification", Dorset H.P., 1987.
- [50] Pressman, R. "Software Engineering - A Practitioner's Approach", McGraw-Hill, 2ª ed., 1987.
- [51] "Manual da Aplicação LT-Control", Laboratory Technologies Corp., 1989

Bibliografia

- Tsai, Lane, Lin "Modern Control Techniques for the Processing Industries", Marcel Dekker, Inc 1986.
- Gane, C. e Sarson, T. "Análise estruturada de sistemas", Livros Técnicos e Científicos Editoras S.A., 1985.
- "The Handbook of Personal Computer Instrumentation", Burr-Brown, 3ª ed, 1988.
- Stephanopoulos, G. "Chemical Process Control, An Introduction to Theory and Practice", Prentice-Hall, 1984.
- Mendonça, J. M. "Microcomputer On-Line Control of WECS", Imperial College, Abr 1986.
- Putman, F.A. "Graphics for Control: Looking Better all the Time", I&CS, Set., pp.63-66, 1988.
- Standridge, C.R. "Animating Simulations using TESS", Computers & Indus. Engng., Vol.10, No.2, pp. 121-134, 1986.
- Switzer, C. "Display Technologies for Control Applications", I&CS, Fev.ruary, pp.49-53, 1985.
- Patterson, D. "The Computer Documentation Kit", Reston Publishing Company, 1984.
- Caldeira, J.C. "Sistema Distribuído de Recolha de Dados para Controlo de Produção em Ambientes Industriais".

**Projecto de um Sistema de Software na área da
Supervisão, Controlo e Aquisição de Dados
(SProcess)**

APÊNDICES

- A - Análise de Produtos Comerciais tipo SCADA.
- B - Dicionário de Dados: Entidades, Processos e Depósitos.
- C - Concepção dos Módulos: fichas de especificação.
- D - Descrição Técnica do sistema de Aquisição de Dados e Controlo Keithley - Series 500.

APÊNDICE A

Análise de Produtos Comerciais tipo SCADA

ÍNDICE

	Pag.
1. Introdução.....	3
2. Produtos Comerciais.....	4
FIX ACE.....	4
FACTORYLINK.....	6
GENESIS CONTROL SERIES.....	8
SCADIX.....	10
CIM-PAC.....	12
VIEWpoint.....	14
PARAGON 500.....	16
MULTI-VIEW.....	18
ASISTANT GPIB 1.01.....	20
LABTECH CONTROL.....	22
AC Workbench 2.0 e 3.0.....	24
3. Conclusões.....	27

1. Introdução

Na comparação entre produtos de software de aplicações tipo SCADA (Sistemas para Supervisão, Controlo e Aquisição de Dados), que se apresenta, procurou-se identificar os aspectos mais relevantes ou distintos no que respeita às características, à forma de configurar e às funções de cada produto. Desejou-se individualizar os aspectos relacionados com a configuração do produto ao sistema em causa, já que representa um problema determinante no sucesso de um produto SCADA de uso geral.

É forçada numa classificação baseada nos ítems que podem ser encontrados numa aplicação tipo, embora, nos produtos em causa, esses aspectos possam não ter uma implementação relevante e, portanto, não registada, já que não se pretendeu uma descrição exhaustiva.

A análise de produtos foi baseada em publicações [46], em manuais das aplicações [51] e na experimentação com algumas versões de demonstração do software.

2. Produtos Comerciais

Produto: FIX ACE.

Características

- 1- Fabricante: Intellution, Inc. Norwood, MA.
- 2- Preço: \$6000 a 10.000.
- 3- Requisitos: PC/XT/AT, 640KB, 20MB HD, drive, portas série e paralela, EGA, placa Artic, DOS 3.xx.
- 4- Linguagem de Programação: Lattice C.
- 5- Expansões: Suporta EMS para a base de dados.
- 6- Drives Disponíveis: 54. Principalmente GE.
- 7- Utilização: Por menus. O editor da base de dados é gráfico. Exige conhecimentos de DOS.
- 8- Operação: Pode incluir uma shell multitarefa que permite executar, sobre o DOS, várias tarefas em background enquanto é executada uma em foreground (no écran). Algumas tarefas são executadas pela placa Artic libertando o CPU.
- 9- Desenvolvimento: Toolkit para produzir drivers e a aceder à base de dados.
- 10- Outras Versões: Fix DMACS para OS/2 , VMS e DAE e suporte de redes locais.
- 11- Documentação: Bons manuais de referência. Exigem formação.
- 12- Observações: Adquire-se por módulos de acordo com as funções pretendidas.

Configurações

- 1- Interface: Usa editores compatíveis: em modo gráfico ou caracter, este com biblioteca de símbolos pré-definidos. Usa cor e pode incluir animação.
- 2- Variáveis: As variáveis podem ser classificadas como entradas ou saídas, digitais ou analógicas, como ciclos PID ou resultados de cálculos. A caracterização é guardada numa base de dados acessível em exploração. A base de dados estabelece as estratégias de controlo e monitorização e pode ser vista globalmente numa forma gráfica.
- 3- Alarmes: Para cada variável, são especificadas as possíveis situações de alarme que passam a actualizar um campo de estado para essa variável.

- 4- Análises: Usa um gerador de gráficos para apresentar valores históricos de variáveis.
- 5- Relatórios: São especificadas as variáveis cujos valores podem ser impressos ou exportados em formato ASCII ou LOTUS (.PRN).
- 6- Hardware:
- 7- Validação: Inclui função de validação da coerência da base de dados configurações.

Funções

- 1- Aquisição: Tempo de varrimento = 0.1s. O arquivo em disco é realizado todos os 2 segundos. Pode incluir um filtro aos valores arquivados para reduzir os dados em disco.
- 2- Visualização: Em quadros sinópticos.
- 3- Análises: Módulo para visualização gráfica de valores históricos. Os gráficos são de aspecto pouco configurável mas de utilização muito fácil.
- 4- Alarmes: Podem ser ligados aos quadros sinópticos onde serão visualizados. Também podem ser registados no disco ou na impressora. São tratados de acordo com prioridades e estão disponíveis numa única fila de validação.
- 5- Operação: A operação directa é realizada a partir dos quadros sinópticos permitindo visualizar os efeitos.
- 6- Controlo: Permite modificações e ajustes dos parâmetros durante a operação on-line.
- 7- Optimização:
- 8- Supervisão: Podem ser enviadas receitas de procedimentos e calendarizações aos controladores.
- 9- Comunicações: A versão DMACS é compatível com a generalidades das LAN até 32 nós e não exige server.
- 10- Relatórios: Incluem os valores registados para as variáveis e respectivos alarmes.
- 11- Help:
- 12- Análises Off-Line: Os valores das variáveis podem ser exportados para o LOTUS ou em modo ASCII.
- 13- Segurança:

Produto: FACTORYLINK (DOS/DESQview)**Características**

- 1- Fabricante: United States Data Corp.
- 2- Preço: \$4,995 - \$10,690
- 3- Requisitos: IBM PC/ AT ou compatível, 649KB RAM, 10 MB HD, drive, portas série e paralela, CGA, DOS 3.1 e DESQview 2.01
- 4- Linguagem de Programação: Microsoft C
- 5- Expansões:
- 6- Drives Disponíveis: 60
- 7- Utilização: Não é imediata. Exige desenvolvimento. A operação está facilitada e inclui suporte ao mouse.
- 8- Operação: A aplicação é composta por quatro programas que podem ser executados em paralelo no DesqView, uma shell multitarefa para o DOS.
- 9- Desenvolvimento: Toolkits para customizar o produto ao problema em causa. É um sistema aberto.
- 10- Outras Versões: OS/2, DAE, VMS, UNIX e versões rede.
- 11- Documentação: Boa, mas de leitura difícil.
- 12- Observações: O módulo de desenvolvimento pode ser usado e vendido em separado do da exploração.

Configurações

- 1- Interface: O editor gráfico é um programa independente, standard, com uma biblioteca de símbolos e onde se produz écrans estáticos, imagens. As características animadas são geradas por um editor de animação e incluem: variáveis, gráficos, movimento de objectos e alterações de cor e atributos.
- 2- Variáveis: A configuração da base de dados é feita por um programa independente e pode incorporar tarefas para cálculos, lógica, arquivo de dados, animação, etc. Linearização, correcções de escala e filtragem não estão pré-definidas nos drives disponíveis e têm que ser configuradas.
- 3- Alarmes: No programa de configuração são especificadas as condições de alarme das variáveis.

4- Análises: As análises de históricos serão feitas off-line num software de uso geral. As análises e os gráficos on-line são definidos na configuração e no editor de animação.

5- Relatórios: Um gerador de relatórios que é também um programa independente, pode arquivar os dados históricos, em ASCII, com vários formatos pré-definidos.

6- Hardware:

7- Validação:

Funções

1- Aquisição:

2- Visualização: Em quadros sinópticos.

3- Análises: As análises são definidas na configuração das variáveis e os gráficos on-line são visualizados nos quadros sinópticos. As análises de valores históricos são feitas off-line.

4- Alarmes: Os alarmes podem ter diferentes atributos como: registo, cor e forma de validação.

Os alarmes podem ser representados nos quadros sinópticos.

5- Operação:

6- Controlo:

7- Optimização: Interface com o sistema pericial Nexpert.

8- Supervisão: Uma tarefa, configurável, permite seleccionar receitas tipo de procedimentos a enviar a PLC's que podem ser desencadeadas por situações do processo.

9- Comunicações:

10- Relatórios: Corresponde a fazer imprimir os ficheiros de arquivo configurados.

11- Help:

12- Análises Off-Line: Os ficheiros de dados podem ser ASCII ou compatíveis com o Dbase compatível.

13- Segurança:

Produto: GENESIS CONTROL SERIES**Características**

- 1- Fabricante: Iconics, Inc.
- 2- Preço: \$4,000 - \$6,000
- 3- Requisitos: IBM, PC/XT/AT ou compatível, coprocessador matemático, 640KB RAM, HD, drive, portas série e paralela, mouse, EGA, DOS 3.X
- 4- Linguagem de Programação: Microsoft C
- 5- Expansões: Suporta EMS e XMS
- 6- Drives Disponíveis: Mais de 100
- 7- Utilização: Intuitiva, baseada em ícons com selecção por mouse, trackball ou touchscreen. Não exige conhecimentos de DOS.
- 8- Desenvolvimento: Possível a criação de macros que condensam uma sequência de comandos do operador. Inclui também um módulo geração de receitas de procedimentos. Inclui ainda Tool-Kits para criação de drivers e acesso à base de dados.
- 9- Outras Versões: GENBASICS para pequenas aplicações e GEN-NET para operação em rede, até 32 nós e sem exigir server.
- 10- Documentação: Muito boa.
- 11- Observações: Um produto de qualidade, com o trabalho de configuração muito facilitado mas a precisar de melhoramentos na fase de operação.

Configurações

As configurações são efectuadas utilizando um programa independente executado sobre o DOS.

1- Interface: O editor gráfico é de fácil utilização e dispõe de uma boa biblioteca de símbolos. Pode incorporar animação. O editor está incluído no programa de configuração e pode importar ficheiros DXF.

2- Variáveis: A base de dados é criada à medida que o utilizador define graficamente a estratégia de monitorização, controlo ou supervisão a implementar. O ambiente é baseado em ícons e utiliza o mouse. As configurações podem ser apreciadas graficamente na globalidade.

Muitos dos parâmetros fixados podem depois ser alterados em operação em tempo real.

3- Alarmes: Podem ser individuais às variáveis ou globais. Os alarmes podem ser encadeados oferecendo os níveis de alarme que se pretender. Os alarmes podem ser de limite baixo ou alto, de taxa de variação ou de desvio ao setpoint.

4- Análises: Existe um gerador de gráficos on-line configurável extensivamente.

5- Relatórios:

6- Hardware:

7- Validação:

Funções

As várias funções disponíveis em exploração constituem um programa independente executado sobre um sistema operativo de tempo real, multitarefa e compatível com o DOS.

1- Aquisição: O tempo de varrimento é de cerca de 1 segundo. O arquivo em disco pode ser feito a taxas diferentes para cada conjunto de variáveis.

2- Visualização: Os écrans sinópticos podem ser ligados entre si. O número de écrans não é limitado. Os écrans podem ser chamados por acontecimentos no processo. Os écrans podem incluir gráficos on-line.

3- Análises: O gerador de gráficos oferece gráficos on-line com uma actualização em base de tempo diferentes (1min, 6 min e entre 30 min e 48h). As análises de dados históricos podem ser analisadas em forma tabular ou gráfica.

4- Alarmes: As mensagens associadas aos alarmes aparecem numa área reservada independente do écran corrente. Os alarmes tem uma representação colorida. No entanto as mensagens mostradas não são guardadas e a mensagem seguinte apaga a anterior. Os alarmes são validados individualmente, mudando de cor na fila de alarmes.

5- Operação: A operação pode ser conduzida a partir dos écrans sinópticos.

6- Controlo:

7- Optimização:

8- Supervisão: Em operação on-line podem ser alteradas receitas de procedimentos e submetidas ao processo.

9- Comunicações:

10- Relatórios: Podem incluir operações e cálculos. São impressos por printscreen o que suspende a aplicação.

11- Help: Menus de ajuda assim como mensagens não existem no módulo de configurações e são escassos em run-time.

12- Análises Off-Line: Os ficheiros de valores históricos podem ser compatíveis com o LOTUS.

13- Segurança:

Produto: SCADIX

Características

1- Fabricante: Pioneering Controls Technologies, Inc.

2- Preço: \$6000 a \$10.000

3- Requisitos: 80386 AT ou MC, 4MB RAM, 40 MB HD, drive, portas série e paralela, EGA, coprocessador, SCO Xenix 2.3.2

4- Linguagem de Programação: MS-C

5- Expansões:

6- Drives Disponíveis: 9

7- Utilização: Em modo character, por comandos e pelo preenchimento de tabelas. São necessários conhecimentos de Unix.

8- Operação: Além de tirar partido das características multitarefa do sistema operativo. Scadix está disponível em configuração multiutilizador podendo ser usado em vários terminais do mesmo computador executando funções diferentes. Claro que o S.O. degrada a performance relativa da aplicação nas funções de aquisição de dados.

9- Desenvolvimento: São assegurados por um módulo que implementa a linguagem GA (General Actions) e pela possibilidade de linkar módulos em C.

10- Outras Versões: Versão em rede baseada em Ethernet com TCP/IP sem exigir server. Versões RISC.

11- Documentação: Muito completa. Inclui um tutorial sobre a utilização do produto.

12- Observações: É um produto mais orientado para Supervisão e Aquisição de dados do que para Controlo Directo.

Configurações

As configurações em geral, podem ser alteradas em fase de exploração.

1- Interface: Pode utilizar um editor em modo carácter ou um editor gráfico compatível com diferentes standards desde VGA até X-Windows.

2- Variáveis: A caracterização das variáveis permite associar ao seu processo de cálculo rotinas pré-programadas que implementem cálculos ou lógica.

3- Alarmes: Caracterizam cada variável com muita flexibilidade.

4- Análises: A configuração dos gráficos é limitada às características pré-existentes. Podem ser representados no mesmo écran, gráficos on-line com diferentes taxas de actualização.

5- Hardware:

6- Relatórios: A configuração de relatórios é feita com o editor gráfico. Podem ser calendarizados para impressão automática.

7- Validação:

Funções

1- Aquisição: O tempo de varrimento é lento (3 segundos para 16 variáveis), embora os écrans sejam actualizados todos os segundos.

2- Visualização: Os gráficos on-line podem ser representados no mesmo écran com diferentes taxas de actualização.

3- Análises: Não incluem dados históricos em disco.

4- Alarmes: As situações de alarme são caracterizadas por três níveis de limite superior e três inferior ou pela taxa de variação. Os alarmes podem desencadear rotinas de serviço, com cálculo e lógica. Os alarmes podem ser inibidos globalmente.

5- Operação: As configurações das variáveis podem ser alteradas em exploração.

6- Controlo: Não dispõe de algoritmos pré-definidos: PID, ON/OFF, etc.

7- Optimização:

8- Supervisão: Não permite envio de receitas ou procedimentos para controlo batch. Permite apenas o envio de parâmetros de forma individualizada.

9- Comunicações: Esta solução oferecendo ambiente multiutilizador é fundamentalmente uma alternativa a sistemas em rede.

10- Relatórios: Não podem incluir valores históricos arquivados em disco.

11- Help: Os manuais estão disponíveis em help on-line.

12- Análises Off-Line: A exportação de dados não está implementada, terá que ser o utilizador a desenvolver software.

13- Segurança:

Produto: CIM - PAC

Características

1- Fabricante: Action Instruments, Inc.

2- Preço: \$2,950 - \$3,450

3- Requisitos: IBM PC/XT/AT ou compatível, coprocessador matemático, 640KB RAM, HD, drive, EGA, DOS 3.3

4- Linguagem de Programação: FORTH, C e Assembler

5- Expansões:

6- Drives Disponíveis: 15

7- Utilização: Fácil, simples e directa. Baseada em menus.

8- Operação: Inclui uma shell para funções de tempo-real e multitarefa que substitui o DOS. Assegura funcionamento em rede de 8 postos CIM-PAC.

9- Desenvolvimento: Oferece programação de macros para comandos de operação. Inclui toolkit para desenvolvimento de drivers e acesso a ficheiros.

10- Outras Versões:

11- Documentação: Manuais limitados com falta de exemplos de aplicação. O help é extensivo, sensível ao contexto e fácil de usar.

12- Observações: Só funciona com um tipo de sistema de aquisição e controlo por computador. Adequado para sistemas simples, dada a sua facilidade de configuração.

Configurações

Não há separação entre os módulos de configuração e de exploração. Cada variável pode começar a ser adquirida e processada enquanto se prossegue a configuração.

1- Interface: Embora o editor gráfico seja muito simples e não suporte o mouse, é fácil produzir écrans sinópticos.

2- Variáveis: Cada variável pode ter uma denominação lógica de até 16 caracteres. Podem existir variáveis reais ou virtuais (calculadas). A cada variável pode ser associado um bloco de funções várias: matemáticas, lógica, de arquivo, estatísticas, de contagem e temporização. A cada variável pode também estar associado um bloco programável, fácil de usar.

3- Alarmes: Todas as variáveis podem ter associado dois alarmes de limite superior e dois de limite inferior e deadbands.

4- Análises: Dispõe de representação gráfica em tempo real e de valores históricos, de fácil utilização mas de flexibilidade limitada.

5- Relatórios: Podem ser configurados relatórios de dados do processo.

6- Hardware:

7- Validação:

Funções

1- Aquisição: Taxa de varrimento de cerca de 1000 pontos por segundo, não fixável pelo utilizador. A taxa de arquivo em disco tem valores fixos que podem ser seleccionados.

2- Visualização: Em quadros sinópticos. Pode ser feito uso de gráficos em tempo real com 3 variáveis por gráfico.

3- Análises: Os gráficos representados podem ser gravados para futuras utilizações. Tanto os dados em tempo real como históricos podem ser vistos em forma gráfica ou tabular.

4- Alarmes: As mensagens de alarme podem ser visualizadas nos vários écrans, ter efeito sonoro, ser guardado em disco, impresso ou desencadear uma acção externa. Só podem ser validados globalmente.

5- Operação: As mensagens no operador podem ser arquivadas em disco. A partir dos écrans sinópticos é possível alterar valores de saída ou setpoints. Os menus de operação são também uma interface configurável que incluem programação de macros para sequências de comandos.

6- Controlo: As funções de controlo pré-definidas são limitadas a ciclos PID e a controlo ON/OFF.

7- Optimização:

8- Supervisão: Assegura funções específicas do processamento batch incluindo a operação de fases em paralelo. Oferece comunicação on-line com outros computadores no mesmo nível hierárquico.

9- Comunicações: Funções como transferência de dados e partilha de ficheiros são asseguradas entre computadores executando a aplicação e ligados por Ethernet. Até 8 estações e não suporta server.

10- Relatórios: Podem ser impressos relatórios com os dados adquiridos.

11- Help: De fácil utilização e muito completo.

12- Análises Off-Line: Um utilitário converte os ficheiros de dados para formato DIF.

13- Segurança: A níveis de segurança com uma password em cada nível. Todos os utilizadores de um dado nível têm o mesmo código de acesso.

Produto: VIEWpoint

Características

1- Fabricante: Tele-Denken Resources, Inc.

2- Preço: \$2,500 - \$5,000

3- Requisitos: IBM PC/AT ou compatível, 640KB RAM, HD, Drive, portas série e paralela, EGA, DOS 3.1

4- Linguagem de Programação: Microsoft C

5- Expansões:

6- Drives Disponíveis: 15

7- Utilização: Sistema de menus com suporte de mouse, teclado, trackball e touch screen. Exige ToolKit para acesso à base de dados.

8- Operação: Um módulo de manuseamento de programas concorrentes assegura funcionamento multitarefa. Mesmo para programas desenvolvidos pelo utilizador.

9- Desenvolvimento: Ambiente de programação lógica para uso geral.

10- Outras Versões: Produtos adicionais para comunicação em redes industriais e para programação de PLC's.

11- Documentação: Bons manuais.

12- Observações:

Configurações

As configurações são efectuadas numa estrutura de menus que permitem uma visão global das especificações. As configurações não são alteráveis na fase de exploração.

1- Interface: É usado um editor gráfico proprietário, orientado por objectos e concebido para uma actualização rápida em tempo real. Dispõe de uma biblioteca de símbolos. É possível definir animação.

2- Variáveis: Podem ser configurados 15 conjuntos de variáveis por cada estratégia de monitorização e controlo. Inclui função de linearização de entradas e saídas.

3- Alarmes: Configuração baseada em layouts standard disponíveis para serem usados nos ambientes de programação.

4- Análises: Podem ser configurados gráficos e tabelas para representação de dados em tempo-real. Pode ser necessária alguma programação para implementar alguns pormenores. A análise e visualização de valores históricos não está definida, exige programação.

5- Relatórios:

6- Hardware:

7- Validação:

Funções

A maior parte das funções têm que ser implementadas em programação lógica ou em C, fazendo uso de uma biblioteca de rotinas orientadas para a monitorização e controlo de processos.

1- Aquisição: Cada um dos 15 conjuntos de variáveis pode ter uma frequência de aquisição diferente.

Os dados podem ser arquivados a uma taxa mínima de 1 segundo ou por triggering.

2- Visualização: Os écrans sinópticos podem ser ligados entre si para facilitar a operação. Os gráficos on-line mostram uma variável por gráfico.

3- Análises:

4- Alarmes: Não estão disponíveis. Todo o tipo de alarme terá de ser construído pelo utilizador em programação lógica ou em C.

5- Operação: Pode ser realizada a partir dos ecrans sinópticos.

6- Controlo:

7- Optimização:

8- Supervisão:

9- Comunicações: Funciona em rede, exigindo server dedicado.

10- Relatórios:

11- Help: Disponível em todos os ecrans. As mensagens de erro ou aviso são visíveis por 20 segundos e depois não são guardadas.

12- Análises Off-Line: Os ficheiros podem ser convertidos para formato PRN.

13- Segurança:

Produto: PARAGON 500

Características

1- Fabricante: Intec Controls Corp.

2- Preço: \$2,500 - \$8,500

3- Requisitos: IBM PC/AT ou compatível, 640KB RAM, HD, drive, portas série e paralela, EGA, coprocessador matemático, DOS 3.0

4- Linguagem de Programação: Microsoft C

5- Expansões: EMS

6- Drives Disponíveis: 28

7- Utilização: Fácil, baseada em interface gráfica comandada por mouse.

8- Operação: Usa para além do DOS um sistema operativo proprietário que assegura velocidade e multitarefa nos procedimentos críticos.

9- Desenvolvimento: ToolKit para desenvolvimento de módulos em C que podem ser incluídos nas configurações e executados em paralelo com a aplicação.

10- Outras Versões: Versão rede.

11- Documentação: Bom manual. Inclui exemplos.

12- Observações:

Configurações

As configurações são realizadas em programas independentes do programa que assegura a exploração.

1- Interface: A interface para operação on-line inclui quadros sinópticos construídos num editor gráfico. Este editor gráfico é um programa independente que integra o desenho dos ecrãs e a sua animação.

2- Variáveis: A configuração das variáveis do sistema é conseguida pelo uso de dois programas, um configura o ambiente de operação (drivers, comunicações, I/O, etc.) e outro configura graficamente a estratégia de monitorização e controlo. Os blocos de cálculo que caracterizam as variáveis podem ser agrupados logicamente de forma a constituírem estruturas mais fáceis de trabalhar.

3- Alarmes: A especificação do tratamento de situações de alarme é feita como um bloco de configuração das variáveis.

4- Análises: Para representação on-line de informação podem ser configurados vários tipos de gráficos em ecrãs acessíveis pelo utilizador. Estes ecrãs podem ser configurados para incluir gráficos de valores históricos que podem assim ser vistos durante a operação on-line.

5- Relatórios: Um gerador de relatórios permite emitir relatórios pré-formatados sobre informação vária.

6- Hardware:

7- Validação: Possui Expert checker que testa a coerência da configuração e fornece sugestões de alojamento e optimização.

Funções

1- Aquisição: Suporta até seis velocidades de aquisição em simultâneo (entre 0.1 segundos e 3276 segundos). O arquivo de dados é feito com uma periodicidade de 1 segundo.

2- Visualização: Em operação on-line é possível visualizar graficamente a estratégia de monitorização e controlo e observar o estado de vários parâmetros.

3- Análises: As análises têm os procedimentos de cálculo definidos na configuração das variáveis.

4- Alarmes:

5- Operação: A operação pode ser efectuada a partir dos quadros sinópticos. Não é possível a criação de macros de comandos. O mouse não é usado na operação on-line.

6- Controlo: Existem disponíveis ciclos PID e sequências de cálculos para suporte a controlo digital e analógico. Não existe, no entanto uma programação livre de estratégias de controlo. Não suporta controlo ON/OFF.

7- Optimização:

8- Supervisão: Alguns parâmetros da estratégia configurada podem ser alterados em tempo-real. É também possível interactuar com os controladores locais.

9- Comunicações:

10- Relatórios: A emissão de relatórios pode ser paralela à operação on-line e ser automatizada. Os relatórios podem incluir informação sobre a estratégia da configuração, sobre variáveis, dados ou sistema.

11- Help: O help é sensível ao contexto e de fácil utilização. Inclui um tutorial on-line.

12- Análises Off-Line: Os dados históricos podem ser visualizados off-line na própria aplicação. Os dados podem também ser exportados para o LOTUS. No entanto o LOTUS ou outra qualquer aplicação não pode ser usada em paralelo com o Paragon.

13- Segurança:

Produto: MULTI-VIEW

Características

1- Fabricante: Indelec, Inc.

2- Preço: \$7,495

3- Requisitos: IBM PC/XT ou compatível, 512KB RAM, HD, drive, porta série e paralela, EGA, DOS 2.1

4- Linguagem de Programação: Assembler, Basic e C

5- Expansões:

6- Drives Disponíveis: 45

7- Utilização: Nem todas as funções são de fácil utilização. Só o editor gráfico suporta mouse, as outras funções permitem criação de macros de comandos.

8- Operação: Os programas que implementam as várias funções, são do tipo TSR, pelo que são executadas em paralelo com o DOS, assegurado um funcionamento multitarefa.

9- Desenvolvimento: ToolKit para programação adicional em C.

10- Outras Versões: OS/2, DAE, Windows 3.0, VMS e versão rede.

11- Documentação: Bons manuais, embora com poucos exemplos.

12- Observações:

Configurações

1- Interface: O editor gráfico embora baseado num produto standard, não é muito intuitivo e tem limitações funcionais. A animação é também limitada não sendo possível alterar atributos ou mudar posições baseadas em alarmes.

2- Variáveis: A instalação e configuração dos drivers para as placas de interface está muito facilitada. Os cálculos matemáticos e lógicos e a sequenciação de procedimentos associados a cada variável são implementados em pequenas folhas de cálculo de uso geral.

3- Alarmes: A configuração de alarmes associados a variáveis é feita de forma diferente conforme estas sejam discretas ou analógicas. Se forem analógicas a configuração é complicada. Os alarmes nunca são especificados directamente nas folhas de cálculo.

4- Análises: Os gráficos on-line podem ser representados nos écrans sinópticos ou em écrans específicos. É possível configurar ainda que com pouca flexibilidade gráficos de valores históricos guardados em disco. Os cálculos associados às variáveis são especificados nas folhas de cálculo.

5- Relatórios: É possível configurar relatórios e temporizar a sua impressão.

6- Hardware:

7- Validação:

Funções

1- Aquisição: O arquivo de dados pode ser limitada pelo tempo ou dimensão e ser circular.

2- Visualização:

3- Análises: A representação gráfica, on-line de, variáveis nos écrans sinópticos é apenas para dar uma ideia pouco rigorosa da sua evolução. A partir do menu de operação pode

ser representado um gráfico multivariável on-line. Em operação podem também ser executados gráficos históricos.

4- Alarmes: As mensagens de alarme são mostradas em qualquer tipo de écran e incluem: data, tempo, prioridade, tipo, nome, valor, etc. Podem ser gravados em disco. A validação é apenas individual.

5- Operação: Um programa utilitário executa em tempo real leitura e escrita de variáveis do sistema.

6- Controlo: Não existem pré-definidos algoritmos de controlo como PID, ON/OFF, etc. Têm que ser implementados usando as folhas de cálculo ou fazer uso de controladores locais para executar o controlo directo.

7- Optimização:

8- Supervisão: É suportada uma interacção com algoritmos de controlo implementados em PLC's. É também possível enviar para os PLC's receitas de parâmetros de forma automática, baseada no tempo, em eventos, ou de forma manual.

9- Comunicações: Um programa tipo utilitário, assegura comunicações e integração em rede.

10- Relatórios:

11- Help: Não existe em todos os écrans. As mensagens de erro são mostradas em qualquer écran e podem ser gravadas em disco.

12- Análises Off-Line: Os dados podem ser exportados em formatos DIF, PRN e ASCII.

13- Segurança:

Produto: ASYSTANT GPIB 1.01

Características

1- Fabricante: Asyst Software Technologies, Inc.

2- Preço:

3- Requisitos: IBM PC/XT/AT ou compatíveis, DOS 2.0 ou superior, coprocessador matemático, 640 KB RAM, HD, drive.

4- Linguagem de Programação:

5- Expansões:**6- Drives Disponíveis:****7- Utilização:** Facilitada pelo uso de menus de funções em modo interactivo.

8- Operação: A aplicação é composta por três módulos: dois módulos que contêm sistema base, análises, estatística e gráficos e outro que acrescenta suporte para interface GPIB/IEEE-488.

9- Desenvolvimento: Elaboração de rotinas que englobam sequências de comandos do operador, permitindo automatizar a aquisição de dados, análise e visualização gráfica. Possível a construção de rotinas em Microsoft C (ver 5.0) Fortran (ver 4.1).

10- Outras Versões: O ASYSTANT GPIB faz parte da família de produtos de software ASYSTANT que complementam o sistema ASYST. O produto ASYSTANT é um software integrado para análise de dados, estatística e gráficos. O produto ASYSTANT PLUS acrescenta conversão A/D, D/A e unidades I/O digitais. O produto ASYSTANT GPIB acrescenta suporte para interface GPIB/IEEE-488, possibilitando o controlo de instrumentos.

11- Documentação: Disponível tutor com exemplos práticos.

12- Observações:**Configurações**

1- Interface: Editor em modo gráfico, por menus.

2- Variáveis: A configuração da base de dados é feita no ecrã principal que inclui uma biblioteca de operações matemáticas, operações com vectores, funções especiais e operações matriciais.

3- Alarmes:

4- Análises: Visão gráfica e análise integrada on-line. Alternativamente, os dados adquiridos podem ser tratados off-line numa biblioteca de tratamento estatístico e de correlação.

5- Relatórios: Disponível editor de texto para documentação dos resultados.

6- Hardware: Possui menu de configuração de periféricos: monitor, plotter, impressora, placa GPIB e drives. Configuração e controlo simultâneo até 12 instrumentos.

7- Validação: Permite teste de comunicação entre computador e instrumento antes da programação ou operação.

Funções

- 1- Aquisição:
- 2- Visualização: Em gráficos on-line, incluindo análise integrada. Disponíveis vários tipos de gráficos.
- 3- Análises: Análises estatísticas, correlação e FFT. Possui rotinas de resolução de equações diferenciais e polinomiais.
- 4- Alarmes:
- 5- Operação: A operação pode ser conduzida a partir de menus de funções pré-definidas. Possibilidade de programação de rotinas.
- 6- Controlo:
- 7- Optimização:
- 8- Supervisão:
- 9- Comunicações:
- 10- Relatórios: Os relatórios construídos podem ser impressos via impressora ou plotter.
- 11- Help: Disponível help on-line.
- 12- Análises Off-Line: Os dados podem ser exportados em formatos comuns (ex: para Lotus 1-2-3) e em ASCII.
- 13- Segurança:

Produto: LABTECH CONTROL

Características

- 1- Fabricante: Laboratory Technologies Corp.
- 2- Preço: \$3,995
- 3- Requisitos: IBM PC ou compatível, 640 KB RAM, HD, drive, portas série e paralela, EGA, DOS 2.0
- 4- Linguagem de Programação: Lattice C
- 5- Expansões: EMS

- 6- Drives Disponíveis: 70, inclui um drive universal.
- 7- Utilização: Não é fácil já que dispõe de muitas funções extensivamente configuráveis. A utilização pelo operador na fase de exploração está muito facilitada.
- 8- Operação: Baseia-se numa shell proprietária que implementa multitarefa sobre o DOS.
- 9- Desenvolvimento: Dispõe de uma linguagem de programação própria para automatizar procedimentos.
- 10- Outras Versões: OS/2, Unix, VMS.
- 11- Documentação: Bons manuais. Incluem tutoriais.
- 12- Observações: Especialmente adequado para aplicações em laboratório.

Configurações

- 1- Interface: O editor gráfico usado é standard (Paintbrush) mas a ligação ao LT/Control é complicada. Além da dificuldade, as características animadas dos sinópticos são limitadas.
- 2- Variáveis: As caracterizações associadas a um dada função nem sempre estão agrupadas no mesmo écran. A estratégia de monitorização e controlo é baseada na configuração de variáveis e blocos que podem ser ligados entre si. Estes blocos incluem funções pré-definidas para controlo, análises, arquivo, comunicações, etc. As funções de análise de dados e de arquivo em ficheiro são poderosas e flexíveis. A estratégia configurada pode ser vista de forma global. Muitos dos parâmetros podem ser alterados on-line no painel de controlo.
- 3- Alarmes: O tratamento de alarmes, por variável, pode ser extensivamente configurado: quatro limites, deadband, ligação a sinótipos, emissão de mensagens ou forma de validação.
- 4- Análises: A representação gráfica pode ser configurada extensivamente para diferentes formatos. Disponível em gráficos on-line ou históricos.
- 5- Relatórios:
- 6- Hardware:
- 7- Validação:

Funções

- 1- **Aquisição:** Em dois modos: normal, com acesso às diferentes funções do produto e rápida, dedicada apenas à função de aquisição de dados. Pode ser configurada para a frequência de leitura aumentar num determinado intervalo de tempo de execução.
- 2- **Visualização:** Em quadros sinópticos, chamados pela tecla de número correspondente.
- 3- **Análises:** Os gráficos de valores históricos podem ser usados em conjunto com gráficos on-line e podem ser desenhados como se os dados estivessem a ser adquiridos. Os diferentes cálculos estatísticos e analíticos podem ser efectuados e representados gráficamente em tempo real. Disponíveis análises avançadas como FFT, derivação, integração e ajuste polinomial.
- 4- **Alarmes:** Os alarmes podem ser visualizados no menu principal e uma mensagem de alarme pode ser vista em todos os écrans. Podem desencadear a selecção de um quadro sinóptico particular. A validação pode ser individual ou global.
- 5- **Operação:**
- 6- **Controlo:** Configurável com diferentes algoritmos pré-definidos. Permite controlo em ciclo fechado ou aberto com os dados retirados de um ficheiro do utilizador. Podem ser desenvolvidos algoritmos próprios.
- 7- **Optimização:** Está disponível um sistema pericial para ajuste automático dos parâmetros dos ciclos de controlo.
- 8- **Supervisão:** Pode incluir um módulo de tolerância a falhas.
- 9- **Comunicações:** Suporta comunicações em rede e partilha de ficheiros.
- 10- **Relatórios:**
- 11- **Help:** Não estão disponíveis écrans de help. As mensagens de erro ou aviso aparecem no ecran mas não são guardadas em disco ou impressas.
- 12- **Análises Off-Line:** Os ficheiros de dados são facilmente exportados para o Lotus 1-2-3.
- 13- **Segurança:**

Produto: AC Workbench 2.0 e 3.0

Características

- 1- **Fabricante:** Strawberry Tree Computers, Inc.

2- Preço:

3- Requisitos: Macintosh II ou Macintosh SE. Placa de conexão analógica: qualquer modelo para computador Macintosh.

4- Linguagem de Programação:**5- Expansões:****6- Drives Disponíveis:**

7- Utilização: Fácil, sem necessidade de programação. Orientada por ícons e com suporte de rato.

8- Operação:

9- Desenvolvimento: Possibilita a elaboração de programas por parte do utilizador em qualquer linguagem compilada.

10- Outras Versões:**11- Documentação:****12- Observações:****Configurações**

1- Interface: Editor gráfico de fácil utilização, por menus e com rato disponível.

2- Variáveis: A estratégia de aquisição de dados e controlo é baseada em símbolos gráficos (ícons). Estão disponíveis 14 ícons diferentes que podem ser ligados entre si de forma a representar simbolicamente o que se irá passar a nível de operação.

3- Alarmes: A configuração do alarme é feita através de um ícon disponível em menu.

4- Análises: Disponível ícon de inserção de gráficos.

5- Relatórios:**6- Hardware:**

7- Validação: A coerência da configuração da estratégia de aquisição e controlo é testada pela aplicação.

Funções

- 1- Aquisição:** Velocidade máxima de aquisição de 300 Hz (Mac II) ou 100 Hz (Mac SE) (igual velocidade de actualização gráfica).
- 2- Visualização:** Visualização no écran do esquema de aquisição de dados ou controlo e visualização dos resultados em gráficos on-line.
- 3- Análises:** Gráficos on-line com escalas definidas pelo utilizador e número ilimitado de variáveis visualizadas. Cor disponível no Mac II. Análises de correlação e FFT não estão disponíveis.
- 4- Alarmes:** Alarme definido para qualquer estado ou limite analógico. As mensagens associadas aos alarmes são definidas pelo utilizador. Disponíveis alarmes sonoros e cor no Mac II.
- 5- Operação:** Operação directa a partir dos menus do écran.
- 6- Controlo:** Permite modificações e ajustes dos parâmetros durante a operação on-line. , Inclui algoritmos pré-definidos de controlo ON/OFF e PID..
- 7- Optimização:**
- 8- Supervisão:**
- 9- Comunicações:**
- 10- Relatórios:**
- 11- Help:** Definição dos vários ícons disponível em help on-line.
- 12- Análises Off-Line:** Os ficheiros de valores históricos são compatíveis com folhas de cálculo, base de dados e software de análise estatística (ex: Excel ou Parameter Manager Plus).
- 13- Segurança:**

3. Conclusões

O estudo comparativo realizado permitiu classificar as características e funções fundamentais dos produtos SCADA. Este tipo de produtos de software estão em permanente processo de desenvolvimento, pelo que as suas características são periodicamente revistas e actualizadas.

Este desenvolvimento é marcado pelo aumento de complexidade já que, por um lado, as características e funções vão sendo mais abrangentes e variadas, e por outro, a configuração deve ser extensiva.

Os grandes vectores de desenvolvimento incluem a utilização em rede, assegurando uma exploração em ambiente distribuído, e a integração de programas do utilizador na estrutura da aplicação, para satisfazer especificidades.

Estes desenvolvimentos justificam, por si só, a atenção que também tem sido dedicada à interface com o utilizador, procurando facilitar tanto o trabalho de configuração (engenharia) como o de exploração (operação ou supervisão) de aplicações cada vez mais complexas. Os melhoramentos são no sentido das interfaces gráficas orientadas por objectos, intuitivas, com as funções identificadas por ícons e com a utilização baseada no mouse.

Os produtos comparados são todos baseados em microcomputadores, mas apresentam estratégias bem diferentes a nível das soluções adoptadas para o sistema operativo e para a arquitectura do software.

No entanto, talvez se possa apresentar como grandes tendências a adopção de soluções standard para implementar as funções de multitarefa, a nível de sistema operativo (Windows 3.0, OS/2 e Unix) e de arquitecturas modulares, baseadas em programas standards que podem ser usados e vendidos em separado e que implementam as funções da aplicação (configurações, editor gráfico, gerador de relatórios, gerador de gráficos, sistemas periciais, etc...).

A nível funcional, podem ser identificadas algumas estratégias diferentes de abordagem aos produtos SCADA. Assim, podemos ter produtos mais de supervisão, com limitações e baixas performances a nível de controlo directo (ex: SCADIX), e outros mais de aquisição e controlo (uns mais industriais outros mais laboratoriais).

APÊNDICE B

Dicionário de Dados: Entidades, Processos e Depósitos

ÍNDICE

	Pag.
1. Dicionário de Entidades.....	3
2. Dicionário de Processos.....	7
3. Dicionário de Depósitos.....	27

1. Dicionário de Entidades

Identifiant : EE2
 Nom : PROCESSO
 Alias :

Les libellés

Libellé	Diagramme
---------	-----------

Notice :

É um processo laboratorial ou industrial, com a instrumentação necessária para aquisição de dados e actuação nas variáveis consideradas relevantes e com a comunicação com o microcomputador suportada numa interface adequada.

A interface pode ser uma qualquer, podendo existir módulos de software (drivers) específicos para cada uma.

Tipicamente, as funções de controlo do processo são asseguradas por controladores locais.

Flux intrants

Diagramme	Identifiant	Nom du flux
Contexte	1-SProcess-SCADA	Actuação
1-SProcess-SCADA	1.2-Exploração On-Li...	Actuação
1.2-Exploração On-Li...	1.2.2-Operação	Actuação

Flux extrants

Diagramme	Identifiant	Nom du flux
Contexte	1-SProcess-SCADA	Dados
1-SProcess-SCADA	1.2-Exploração On-Li...	Dados
1.2-Exploração On-Li...	1.2.3-Aquisição	Dados

Identifiant : EE1
 Nom : UTILIZADOR
 Alias :

Les libellés

Libellé	Diagramme
---------	-----------

Notice :

O utilizador poderá ter dois perfis diferentes:

- Supervisor

Configura o software às características do processo e ao tipo de resposta e interacção que pretender ver implementadas.

Após a operação de um processo pode querer fazer análises mais elaboradas com dados históricos do processo.

- Operador

Durante a operação do processo vai querer interactivar: acompanhar a evolução de variáveis, ser avisado em caso de problemas, receber relatórios e poder actuar manualmente no processo.

Flux intrants

Diagramme	Identifiant	Nom du flux
Contexte	1-SProcess-SCADA	Sinópticos
Contexte	1-SProcess-SCADA	Alarmes
Contexte	1-SProcess-SCADA	Gráficos
Contexte	1-SProcess-SCADA	Erros de Configuraçã...
Contexte	1-SProcess-SCADA	Relatórios
Contexte	1-SProcess-SCADA	Tabelas
1-SProcess-SCADA	1.3-Operação Off-Lin...	Relatórios
1-SProcess-SCADA	1.1-Configuração da ...	Erros de Configuraçã...
1-SProcess-SCADA	1.2-Exploração On-Li...	Alarmes
1-SProcess-SCADA	1.2-Exploração On-Li...	Gráficos
1-SProcess-SCADA	1.2-Exploração On-Li...	Sinópticos
1-SProcess-SCADA	1.2-Exploração On-Li...	Tabelas
1.2-Exploração On-Li...	1.2.1-Visualização de...	Sinópticos
1.2-Exploração On-Li...	1.2.5-Análises	Gráficos
1.2-Exploração On-Li...	1.2.7-Tratamento de ...	Alarmes
1.2-Exploração On-Li...	1.2.5-Análises	Tabelas
1.1-Configuração da ...	1.1.5-Validar configu...	Erros de Configuraçã...
1.3-Operação Off-Lin...	1.3.3-SGBD (DbaseI...	Relatórios
1.3-Operação Off-Lin...	1.3.4-Spreadsheet (L...	Relatórios

Flux extrants

Diagramme	Identifiant	Nom du flux
Contexte	1-SProcess-SCADA	Comandos

Contexte	1-SProcess-SCADA	Configurações
1-SProcess-SCADA	1.1-Configuração da ...	Configurações
1-SProcess-SCADA	1.2-Exploração On-Li...	Comandos
1-SProcess-SCADA	1.2-Exploração On-Li...	Configurações
1.2-Exploração On-Li...	1.2.2-Operação	Comandos
1.2-Exploração On-Li...	1.2.8-Configuração	Configurações
1.1-Configuração da ...	1.1.1-Editor Gráfico	Configurações
1.1-Configuração da ...	1.1.2-Definir Variáve...	Configurações
1.1-Configuração da ...	1.1.3-Definir Análise...	Configurações
1.3-Operação Off-Lin...	1.3.2-Converte em A...	Nr_Canal

2. Dicionário de Processos

Identifiant : 1
 Nom : SProcess-SCADA
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Sistema de software que oferece facilidades para a operação ou gestão de um processo laboratorial ou industrial.

É um sistema interactivo configurável a um dado processo que implementa funções em tempo real de supervisão.

Flux intrants

Diagramme	Identifiant	Nom du flux
Contexte	EE1-UTILIZADOR	Comandos
Contexte	EE2-PROCESSO	Dados
Contexte	EE1-UTILIZADOR	Configurações

Flux extrants

Diagramme	Identifiant	Nom du flux
Contexte	EE2-PROCESSO	Actuação
Contexte	EE1-UTILIZADOR	Sinópticos
Contexte	EE1-UTILIZADOR	Alarmes
Contexte	EE1-UTILIZADOR	Gráficos
Contexte	EE1-UTILIZADOR	Erros de Configuraçã...
Contexte	EE1-UTILIZADOR	Relatórios
Contexte	EE1-UTILIZADOR	Tabelas

Identifiant : 1.1
 Nom : Configuração da Aplicação
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
Supervisor		

Notice :

Permite a adaptação do sistema às características do processo em causa.
 É possível definir:

- A interface para representação sinótica e respectivas actualizações e animações. Utiliza o editor gráfico "EDITOR".
- Caracterizar as variáveis relevantes do processo, quanto à aquisição, arquivo e alarmes.
- Caracterizar variáveis do sistema.
- Definir as análises e representações gráficas disponíveis on-line.
- Testar a coerência das configurações efectuadas.

As configurações são guardadas em disco para utilização na exploração on-line e para posteriores alterações.

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	EE1-UTILIZADOR	Configurações

Flux extrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	D2-CONFIGURAÇÃO...	Cnf_Variáveis
1-SProcess-SCADA	D4-ECRANS	Ecrans
1-SProcess-SCADA	D5-PARAMETROS	Cnf_Parametros
1-SProcess-SCADA	EE1-UTILIZADOR	Erros de Configuraçã...
1-SProcess-SCADA	D2-CONFIGURAÇÃO...	Cnf_Análises

Identifiant : 1.1.1
 Nom : Editor Gráfico
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Permite o desenho do layout dos écrans que vão representar um dado processo e guardá-los em discos como ficheiros imagem (ECRAN*.IMG), e permite especificar o tipo de representação de variáveis do sistema: numérica, gráfica, alfanumérica ou animada; que é guardada em disco como parâmetros dos écrans (ECRAN#.PRO).

Ao editor está associado um interpretador dos ficheiros IMG e-PRO, responsável pela actualização on-line dos écrans do processos.

Estas funções são implementadas por um editor gráfico independente da aplicação SProcess. Para o editor adoptado desenvolve-se um interpretador dos ficheiros produzidos que integra SProcess. Esta filosofia assegura que se possa utilizar um editor gráfico diferente desde que se desenvolva o interpretador correspondente.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	EE1-UTILIZADOR	Configurações
1.1-Configuração da ...	D5-PARAMETROS	Cnf_Parametros
1.1-Configuração da ...	D4-ECRANS	Ecrans

Flux extrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	D5-PARAMETROS	Cnf_Parametros
1.1-Configuração da ...	D4-ECRANS	Ecrans

Identifiant : 1.1.2
 Nom : Definir Variáveis
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Permite a configuração das variáveis do sistema:

- número total de variáveis
- numero da variável
- nome da variável
- endereço do canal
- tipo de variável (E/S)
- tamanho da fila circular
- frequência de aquisição
- factores de calibração K1 e K2
- alarme superior
- alarme inferior
- arquivo(sim/não)
- correspondência com os parâmetros dos écrans

Estas configurações devem ser guardadas em ficheiros.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	EE1-UTILIZADOR	Configurações
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Variáveis

Identifiant : 1.1.3
 Nom : Definir Análises
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Permite definir os cálculos e o tipo de saída dos resultados (tabelas ou gráficos) que vão constituir análises disponíveis na exploração on-line do sistema.

São produzidos ficheiros de configurações para um gerador de gráficos e tabelas.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	EE1-UTILIZADOR	Configurações
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Gráficos
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Tabelas

Flux extrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Gráficos
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Tabelas

Identifiant : 1.1.5
 Nom : Validar configurações
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Analisa o conjunto das configurações efectuadas e detecta eventuais incoerências e limitações de recursos:

- Nº de canais configurado compatível com os disponíveis no SAC
- Variáveis compatíveis com as Análises configuradas
- Validar a existência de ficheiros: IMG, PRO

São evitadas assim repetidas validações on-line.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	D2-CONFIGURAÇÃO...	Cnf_Variáveis
1.1-Configuração da ...	D5-PARAMETROS	Cnf_Parametros
1.1-Configuração da ...	D4-ECRANS	Ecrans

Flux extrants

Diagramme	Identifiant	Nom du flux
1.1-Configuração da ...	EE1-UTILIZADOR	Erros de Configuraçã...

Identifiant : 1.2
 Nom : Exploração On-Line
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
Operador		

Notice :

Funções desempenhadas pelo sistema durante a aquisição de dados do processo e em paralelo com esta.

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	EE1-UTILIZADOR	Comandos
1-SProcess-SCADA	EE2-PROCESSO	Dados
1-SProcess-SCADA	D4-ECRANS	Ecrans
1-SProcess-SCADA	D5-PARAMETROS	Cnf_Parametros
1-SProcess-SCADA	D2-CONFIGURAÇÃO...	Cnf_Gráficos
1-SProcess-SCADA	EE1-UTILIZADOR	Configurações
1-SProcess-SCADA	D2-CONFIGURAÇÃO...	Cnf_Variáveis
1-SProcess-SCADA	D2-CONFIGURAÇÃO...	Cnf_Tabelas

Flux extrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	EE2-PROCESSO	Actuação
1-SProcess-SCADA	D1-HISTÓRICO	Dados
1-SProcess-SCADA	EE1-UTILIZADOR	Alarmes
1-SProcess-SCADA	EE1-UTILIZADOR	Gráficos
1-SProcess-SCADA	EE1-UTILIZADOR	Sinópticos
1-SProcess-SCADA	EE1-UTILIZADOR	Tabelas

Identifiant : 1.2.1
 Nom : Visualização de Sinótipos
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Um interpretador dos ficheiros de configurações do EDITOR, actualiza, no écran, as imagens do processo e as respectivas representações numéricas, alfanuméricas, gráficas ou animadas, conforme os valores mais recentes das variáveis do sistema.

São permitidos até 8 quadros sinóticos (imagens) do processo seleccionáveis por menu ou por FKeys.

É executada em foreground e em paralelo com a aquisição de dados.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D4-ECRANS	Ecrans
1.2-Exploração On-Li...	D5-PARAMETROS	Cnf_Parametros
1.2-Exploração On-Li...	D3-Buffer: TABELA	Dados

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE1-UTILIZADOR	Sinóticos

Identifiant : 1.2.2
 Nom : Operação
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Função:

Oferece ao utilizador a possibilidade de intervir directamente no Processo:

- alterar valores de variáveis de saída (ex: fechar válvula, ligar ventilação).
- alterar estados do sistema .
- alterar definição das variáveis do proceso (ex: frequência de aquisição, armazenamento, etc)

Deverá permitir a comunicação com controladores locais para a alteração de parametros de controlo. Começará por assegurar o funcionamento com um controlador Armfield.

É executada em foreground e em paralelo com a aquisição de dados.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE1-UTILIZADOR	Comandos
1.2-Exploração On-Li...	D3-Buffer: TABELA	Dados
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE2-PROCESSO	Actuação

Identifiant : 1.2.3
 Nom : Aquisição
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Em períodos definidos são efectuadas, em background, leituras a canais de um sistema de aquisição de dados e controlo (SAC). Os valores lidos são colocados em memória na tabela de dados. Conforme as características do SAC os valores guardados podem estar já amplificados, filtrados e calibrados.

É uma rotina modular com ligação ao programa principal apenas pela tabela de dados e pelos ficheiros de configuração. Funciona, portanto, como um driver para o SAC em causa, no caso um Keithley 500 com placa para entradas analógicas.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE2-PROCESSO	Dados
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D3-Buffer: TABELA	Dados

Identifiant : 1.2.5
 Nom : Análises
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

A pedido do utilizador são efectuadas análises de dados pré-definidas na Configuração com resultados mostrados em forma gráfica ou tabular.

Características:

- Cálculos matemáticos
- Gráficos multilinhas
- Tabelas multivariável

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D2-CONFIGURAÇÃO...	Cnf_Gráficos
1.2-Exploração On-Li...	D3-Buffer: TABELA	Dados
1.2-Exploração On-Li...	D2-CONFIGURAÇÃO...	Cnf_Tabelas

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE1-UTILIZADOR	Gráficos
1.2-Exploração On-Li...	EE1-UTILIZADOR	Tabelas

Identifiant : 1.2.6
 Nom : Arquivo
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Com a periodicidade definida são guardados em disco os valores de variáveis do sistema que descrevem a evolução do processo.

O armazenamento de variáveis do sistema é implementado em background e em paralelo com a aquisição de dados.

São produzidos ficheiros de dados - Históricos.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	Cnf_Variáveis
1.2-Exploração On-Li...	D3-Buffer: TABELA	Dados

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D1-HISTÓRICO	Dados

Identifiant : 1.2.7
 Nom : Tratamento de Alarmes
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

É executada uma monitorização dos valores adquiridos do processo, avaliando se estes estão dentro dos limites previamente fixados. Quando tal não se verifica é emitido um alarme que é processado com visualização de uma mensagem ao operador. As mensagens devem ser validadas de forma individual conforme as situações de alarme forem sendo ultrapassadas.

É implementado em background e em paralelo com a aquisição de dados eo arquivo.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D3-Buffer: TABELA	Dados
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE1-UTILIZADOR	Alarmes

Identifiant : 1.2.8
 Nom : Configuração
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Alumas características do sistema podem ser alteradas sem interromper o processo on-line de aquisição de dados.

Deve ser possível alterar as configurações das Variáveis e das Análises.

É executada em foreground e em paralelo com a aquisição de dados.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	EE1-UTILIZADOR	Configurações
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D2-CONFIGURAÇÃO...	Cnf_Gráficos
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	Cnf_Variáveis
1.2-Exploração On-Li...	D2-CONFIGURAÇÃO...	Cnf_Tabelas

Identifiant : 1.2.9
 Nom : Inicialização
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

O início da exploração on-line cacterizada pela aquisição de dados é antecedido pela leitura das configurações das variáveis, de ficheiros em disco para buffers em Ram, por testes ao hardware existente e pela instalação dos recursos para implementar processamento em paralelo.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D2-CONFIGURAÇÃO...	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	D15-Buffer: CNF_V...	

Identifiant : 1.3
 Nom : Operação Off-Line
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
Operador		

Notice :

Os dados existentes no ficheiro de histórico do processo podem ser seleccionados e escritos para ficheiros em formatos compatíveis com packages de uso generalizado para análise, classificação e produção de tabelas ou gráficos: Lotus 123, Dbase III, Statgraphics.

Permite executar estes packages no ambiente da aplicação.

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	D1-HISTÓRICO	Dados

Flux extrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	EE1-UTILIZADOR	Relatórios

Identifiant : 1.3.2
 Nom : Converte em ASCII
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Converte os ficheiros de dados dos canais seleccionados de binário para um formato exportável para aplicações de uso geral.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	D1-HISTÓRICO	Dados
1.3-Operação Off-Lin...	EE1-UTILIZADOR	Nr_Canal

Flux extrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	D10-HISTÓRICO-A...	Dados-A

Identifiant : 1.3.3
Nom : SGBD (DbaseIII)
Alias :
Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Executa o DBase.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	D10-HISTÓRICO-A...	Dados-A

Flux extrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	EE1-UTILIZADOR	Relatórios

Identifiant : 1.3.4
 Nom : Spreadsheet (Lotus)
 Alias :
 Traitement : Information

Nom de la ressource	Type	Rôle
---------------------	------	------

Notice :

Executa o Lotus.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	D10-HISTÓRICO-A...	Dados-A

Flux extrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	EE1-UTILIZADOR	Relatórios

3. Dicionário de Depósitos

Identifiant : D15
 Nom : Buffer: CNF_VARIÁVEIS
 Alias :

Notice :

É um buffer em RAM com as configurações das variáveis do processo incluindo as condições de alarme e arquivo.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	1.2.9-Inicialização	
1.2-Exploração On-Li...	1.2.8-Configuração	Cnf_Variáveis

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	1.2.6-Arquivo	Cnf_Variáveis
1.2-Exploração On-Li...	1.2.7-Tratamento de ...	Cnf_Variáveis
1.2-Exploração On-Li...	1.2.8-Configuração	Cnf_Variáveis
1.2-Exploração On-Li...	1.2.2-Operação	Cnf_Variáveis
1.2-Exploração On-Li...	1.2.3-Aquisição	Cnf_Variáveis

Nom de la donnée	I	E	Type	Longueur
Alarme-inferior	•		R	8
Alarme-superior	•		R	8
Arquivo-flag	•		B	1
Var-endereço	•		I	4
Var-nome	•		A	30
Var-numero	•		I	2
Var-nr_total	•		I	2
Var-tipo	•		A	1
Var-tamanho_fila	•		I	2
Var-frequência	•		I	4
Var-calibração_K1	•		R	8
Var-calibração_K2	•		R	8
Var-parametro_ecran	•			0

Identifiant : D3
 Nom : Buffer: TABELA
 Alias :

Notice :

Guarda os valores dos dados mais recentes relativos a variáveis lidas do processo:

- Identificação (Nr_variável)
- Respetivos tempos da leitura (Tempo)
- Respetivos valores da leitura (Valor)

Flux intrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	1.2.3-Aquisição	Dados

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	1.2.2-Operação	Dados
1.2-Exploração On-Li...	1.2.1-Visualização de...	Dados
1.2-Exploração On-Li...	1.2.5-Análises	Dados
1.2-Exploração On-Li...	1.2.6-Arquivo	Dados
1.2-Exploração On-Li...	1.2.7-Tratamento de ...	Dados

Nom de la donnée	I	E	Type	Longueur
Dados-tempo	•		I	4
Dados-valor	•		R	8
Dados-nr_variavel	•		A	8

Identifiant : D2
 Nom : CONFIGURAÇÕES
 Alias :

Notice :

Guarda com carácter permanente a seguinte informação:

- caracterização das variáveis do sistema incluindo condições para arquivo e alarme
- descrição das análises on-line (gráficos e tabelas)

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	1.1-Configuração da ...	Cnf_Variáveis
1-SProcess-SCADA	1.1-Configuração da ...	Cnf_Análises
1.2-Exploração On-Li...	1.2.8-Configuração	Cnf_Gráficos
1.2-Exploração On-Li...	1.2.8-Configuração	Cnf_Tabelas
1.1-Configuração da ...	1.1.2-Definir Variáve...	Cnf_Variáveis
1.1-Configuração da ...	1.1.3-Definir Análise...	Cnf_Gráficos
1.1-Configuração da ...	1.1.3-Definir Análise...	Cnf_Tabelas

Flux extrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	1.2-Exploração On-Li...	Cnf_Gráficos
1-SProcess-SCADA	1.2-Exploração On-Li...	Cnf_Variáveis
1-SProcess-SCADA	1.2-Exploração On-Li...	Cnf_Tabelas
1.2-Exploração On-Li...	1.2.5-Análises	Cnf_Gráficos
1.2-Exploração On-Li...	1.2.9-Inicialização	Cnf_Variáveis
1.2-Exploração On-Li...	1.2.5-Análises	Cnf_Tabelas
1.1-Configuração da ...	1.1.2-Definir Variáve...	Cnf_Variáveis
1.1-Configuração da ...	1.1.3-Definir Análise...	Cnf_Gráficos
1.1-Configuração da ...	1.1.5-Validar configu...	Cnf_Variáveis
1.1-Configuração da ...	1.1.3-Definir Análise...	Cnf_Tabelas

Nom de la donnée	I	E	Type	Longueur
Gráfico-parametros	.		S	0
Gráfico-pontos [][]	.		I	0
Gráfico-legenda[][]	.		A	0
Gráfico-formula	.		A	30
Gráfico-linha/canal	.		I	2
Var-numero		.	I	2
Var-nome		.	A	30
Var-endereço		.	I	4

Alarme-superior	•	R	8
Alarme-inferior	•	R	8
Arquivo-flag	•	B	1
Tab-nr_pontos	•	I	4
Tab-coluna1	•	I	2
Tab-coluna2	•	I	2
Tab-coluna3	•	I	2
Tab-coluna4	•	I	2

Identifiant : D4
 Nom : ECRANS
 Alias :

Notice :

Conjunto de ficheiros imagem (ecrans EGA) representando partes do processo e produzidos no editor gráfico, Editor.

1 écran EGA - 112.018 KB

Os ficheiros são ECRAN#.IMG e encontram-se no directório EDITOR\PROCESS\IMG

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	1.1-Configuração da ...	Ecrans
1.1-Configuração da ...	1.1.1-Editor Gráfico	Ecrans

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	1.2.1-Visualização de...	Ecrans
1-SProcess-SCADA	1.2-Exploração On-Li...	Ecrans
1.1-Configuração da ...	1.1.1-Editor Gráfico	Ecrans
1.1-Configuração da ...	1.1.5-Validar configu...	Ecrans

Nom de la donnée	I	E	Type	Longueur
Editor_imagem (em KB)	.	.		113

Identifiant : D1
 Nom : HISTÓRICO
 Alias :

Notice :

São guardados com carácter permanente os seguintes dados relativos a variáveis lidas do processo:

- Identificação (Nr-variável)
- Respectivos tempos da leitura (Tempo)
- Respectivos valores da leitura (Valor)

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	1.2-Exploração On-Li...	Dados
1.2-Exploração On-Li...	1.2.6-Arquivo	Dados

Flux extrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	1.3-Operação Off-Lin...	Dados
1.3-Operação Off-Lin...	1.3.2-Convert em A...	Dados

Nom de la donnée	I	E	Type	Longueur
Dados-nr_variavel	•		A	8
Dados-tempo	•		I	4
Dados-valor	•		R	8

Identifiant : D10
 Nom : HISTÓRICO-ASCII
 Alias :

Notice :

É o arquivo de dados históricos do processo, mas no formato ASCII aceite por aplicações de uso geral.

Flux intrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	1.3.2-Convert em A...	Dados-A

Flux extrants

Diagramme	Identifiant	Nom du flux
1.3-Operação Off-Lin...	1.3.3-SGBD (DbaseI...	Dados-A
1.3-Operação Off-Lin...	1.3.4-Spreadsheet (L...	Dados-A

Nom de la donnée	I	E	Type	Longueur
Dados-valor	•		R	8
Dados-nr_variavel	•		A	8
Dados-tempo	•		I	4

Identifiant : D5
 Nom : PARAMETROS
 Alias :

Notice :

Guarda os parâmetros que caracterizam o tipo de actualização (numérica, gráfica e animada) de variáveis do sistema nos écrans do processo.
 São os ficheiros ECRAN#.PRO no directório EDITOR\PROCESS\PAR .

Flux intrants

Diagramme	Identifiant	Nom du flux
1-SProcess-SCADA	1.1-Configuração da ...	Cnf_Parametros
1.1-Configuração da ...	1.1.1-Editor Gráfico	Cnf_Parametros

Flux extrants

Diagramme	Identifiant	Nom du flux
1.2-Exploração On-Li...	1.2.1-Visualização de...	Cnf_Parametros
1-SProcess-SCADA	1.2-Exploração On-Li...	Cnf_Parametros
1.1-Configuração da ...	1.1.1-Editor Gráfico	Cnf_Parametros
1.1-Configuração da ...	1.1.5-Validar configu...	Cnf_Parametros

Nom de la donnée	I	E	Type	Longueur
Editor_parametros	.	.	S	0

APÊNDICE C

Concepção dos Módulos: Fichas de Especificação

ÍNDICE

Pág.

1. Fichas de Especificação:

Gestor da Aplicação	4
Gestor On-Line	5
Lê_password.....	6
Testa_password.....	6
Editor (alterações a produzir).....	7
Escreve_imagem.....	7
Conf_variáveis.....	8
Lê_var.....	8
Escreve_var	9
Conf_variáveis_s	9
Lê_var_s.....	10
Escreve_var_s.....	10
Conf_gráfico	11
Lê_gráfico	11
Escreve_gráfico.....	12
Conf_tabela.....	12
Lê_tabela.....	13
Escreve_tabela	14
Validar Configurações	14
Inicialização	14
Monit_calibra.....	15

Monit_instala.....	16
Reg_instala.....	16
Monitorização.....	17
Registo.....	19
Mostra_alarmes.....	19
Valida_alarmes.....	20
Mostra_gráfico.....	20
Tabela_array.....	21
Mostra_tabela.....	21
Armação.....	22
Mostra_valor.....	22
Mostra_sinóptico.....	23
Lê_imagem.....	23
Conf_var_saída.....	24
Envia_var_saída.....	24
Envia_ficheiro.....	25
Envia_comando.....	25
Terminação.....	26
Monit_desinstala.....	26
Reg_desinstala.....	27
Seleccção de canal.....	27
Conversão de dados.....	28
 2. Módulos implementados.....	 29

1. Fichas de Especificação

Módulo: Gestor da Aplicação

Tipo:

Procedimento principal (interface)

Funções:

Implementa a interface principal com o utilizador usando menus de barras.

Permite seleccionar e chamar os procedimentos que implementam processamento Off-Line: Configuração da Aplicação e Operação.

Permite chamar o procedimento que implementa a Exploração On-Line.

Oferece valores de defeito para as configurações da aplicação.

Subordinados:

Configurar Sinópticos, Configurar Variáveis, Configurar Gráficos, Configurar Tabelas, Validar Configurações, Exploração On-Line, Conversão.

Dependência:

Interfaces:

Os procedimentos chamados são essencialmente independentes, poderiam ser todos programas executáveis. Cada procedimento é então responsável por implementar as suas entradas de dados e por produzir as suas saídas, sendo a comunicação entre os procedimentos aqui usados assegurada pela partilha de dados em ficheiro.

Processamento:

Chama Segurança

Actualiza menus e conforme a selecção do utilizador, chama:

 Configurações

 Sinópticos

 Variáveis

 Gráficos

 Tabelas

 Validar

 Operação Off_line

 Conversão

 Lotus

 DBase

 Exploração On-line

Dados:

Configurações globais do sistema.

Restrições:

A interface aqui implementada deve poder estender-se a todo o processamento off-line, mantendo a hierarquia proposta nos Mapas de Estrutura.

Irão poder ser configurados 6 gráficos e 6 tabelas.

Os procedimentos associados ao processamento off-line são essencialmente independentes não tendo necessidade de grandes trocas de informação.

A utilização dos procedimentos deve permitir que estes passem a ser programas executáveis. Os procedimentos associados ao processamento Off-Line, que não sejam

necessários, não devem permanecer residentes em RAM na Exploração On-Line, especialmente os de grande dimensão.

Este módulo deve funcionar como um centro de transacções.

Módulo: Gestor On-Line

Tipo:

Procedimento (interface)

Funções:

Implementa a interface com o utilizador durante o processamento on-line.

Permite seleccionar e chamar os procedimentos que implementam Visualização de Sinópticos, Análises (Gráficos e Tabelas), Alarmes (Visualização e Validação), Configuração e Operação do processo em paralelo com as funções de background, a Monitorização e o Registo.

Com a selecção do fim da operação chama o procedimento Terminação.

É um procedimento Gestor de Transacções, em que os procedimentos de background são chamados pelo sistema de hardware (o PIC do PC).

Subordinados:

Sinópticos (Interpretador de parâmetros do Editor), Análises (Gerador de Gráficos e de Tabelas), Configurações (de Gráficos e de Tabelas), Operação, Alarmes e Terminação.

Dependência:

É chamado pelo Gestor da Aplicação, depois da execução do Inicialização, a quem devolve o controlo com o fim do processamento em tempo real.

Interfaces:

A comunicação do Gestor On-Line com os procedimentos off-line é feito exclusivamente através de ficheiros de dados e configurações.

A comunicação entre os procedimentos chamados por Gestor On-Line é implementada através dos buffers Tabela e Lista_alarmes e das variáveis de configuração.

Processamento:

Inicializa menus para exploração on-line

Conforme selecção de utilizador actualiza menus e chama:

- Terminação.
- Sinópticos.
- Análises (Gráficos ou Tabelas).
- Alarmes (Visualização ou Validação).
- Operação (Comandos, Ficheiros ou Variáveis).
- Configuração (Gráficos ou Tabelas).

Dados:

Variáveis de configuração da aplicação e buffers Tabela e Lista_alarmes.

Restrições:

O número máximo de écrans sinópticos utilizáveis é de oito. O número máximo de gráficos e tabelas utilizáveis e configuráveis é de seis.

O tratamento dos processos de background é assegurado pelo sistema de hardware que, através do controlador programável de interrupções (PIC) do PC responde às interrupções gerados para monitorização em que é chamado o módulo Monitorização ou para registo em que é chamado o módulo Registo. Estes módulos de serviço a

interrupções quando terminam devolvem o controlo ao módulo que o detinha antes da interrupção ser atendida.

A interrupção para monitorização deve ser de maior prioridade que a para registo podendo interromper o processamento desta.

Módulo: Lê_password

Tipo:

Procedimento (interface)

Funções:

Interface com o utilizador para entrada do seu password que funciona como autorização de utilização da aplicação.

Subordinados:

Input.

Dependência:

É chamado pelo Gestor da Aplicação no arranque da aplicação.

Interfaces:

Devolve o password introduzido pelo utilizador.

Processamento:

Écran de pedido de password.

Chama Input.

Dados:

Variáveis auxiliares.

Restrições:

Módulo: Testa_password

Tipo:

Procedimento (processamento).

Funções:

Avaliação da autorização que o utilizador possui para trabalhar na aplicação. Pode ser nível 0 - não autorizado, nível 1 - supervisor, ou nível 2 - operador. Deverá existir um password de defeito para acesso incondicional.

Subordinados:

Dependência:

É chamado pelo Gestor da Aplicação sempre que pretenda restringir o acesso a determinadas funções, de acordo com o nível de autorização do utilizador.

Interfaces:

São passados como argumentos o password introduzido pelo utilizador e os passwords de supervisão e operação, previamente configurados nas variáveis globais do sistema.

Devolve nível de autorização.

Processamento:

Se password = password por defeito, retorna nível 1.

Se password = password de supervisão, retorna nível 1.

Senão, se password = password de operação, retorna nível 2.

Senão retorna 0.

Dados:

Não usa variáveis locais.

Restrições:

Pretende-se uma solução de segurança que permita proteger selectivamente determinadas funções chamadas pelo Gestor da Aplicação.

Módulo: Editor (alterações a produzir)

Embora o programa Editor já exista é necessário efectuar algumas alterações, já que ele oferece outras funções para além dum editor gráfico que não interessa considerar e por outro lado não inclui algumas características indispensáveis.

Assim serão efectuadas as seguintes alterações:

- Tornar o Editor um programa de execução independente no contexto de uma aplicação tipo SCADA. São retiradas funções relacionadas com a animação dos sinópticos e de representação de gráficos.
- Produzir a rotina de interface entre o Editor e a Aplicação: `Escrive_imagem`.
- Melhorar a operação com o Editor: concentrar menu de opções num só écran, melhorar a utilização do cursor e dos menus, e acelerar interacção.

Módulo: `Escrive_imagem`**Tipo:**

Procedimento (saída)

Funções:

Escrive em ficheiro (*.IMG) em disco a imagem existente no écran.

Subordinados:**Dependência:**

É chamado pelo Editor.

Interfaces:

Precisa do nome do ficheiro a produzir.

Processamento:

Calcula a dimensão de cada ponto do écran a ser processada.

Abre o ficheiro imagem (*.IMG).

Aloca dinamicamente memória para um buffer temporário.

Lê a 1ª parte do ficheiro para o buffer.

Coloca buffer no écran.

Lê a 2ª parte do ficheiro para o buffer.

Coloca buffer no écran.

Lê a 3ª parte do ficheiro para o buffer.

Coloca buffer no écran.

Liberta a memória ocupada pelo buffer.

Fecha o ficheiro.

Dados:**Restrições:**

O editor gráfico a usar na aplicação GP, deve permitir uma utilização muito independente que facilite desenvolvimento ou substituição.

A comunicação do editor com a aplicação será então assegurada por dois tipos de informação, uma de descrição da imagem estática desenhada no écran e outra de descrição da animação que o editor deve permitir especificar.

A imagem estática é a tratada pelo editor como uma imagem vectorial com um tempo de colocação no écran muito grande (5-10s) não compatível com uma selecção em tempo real de diferentes écrans. Optou-se então por adicionar ao editor uma opção de gravação em disco da imagem bitmapped do écran. Esta opção é em geral encontrada nos editor comerciais, mesmo nos vectoriais em que aqui estaremos mais interessados.

Esta imagem de écran têm como desvantagem produzirem ficheiros muito grandes (cerca de 110 KB para EGA), pelo que a sua utilização deve ser optimizada para não produzir picos de consumo de memória RAM. Não é considerado como limitante o consumo de memória em disco.

Uma solução radical para o consumo de memória pode ser a utilização de compressores de imagens que reduzem um écran EGA para 10 KB com um tempo de execução razoável.

Módulo: Conf_variáveis

Tipo:

Procedimento (interface)

Funções:

Permitir ao utilizador configurar as variáveis que caracterizam o processo.

Subordinados:

Lê_var, Input, Escreve_var.

Dependência:

É chamado pelo Gestor da Aplicação por selecção do utilizador.

Interfaces:

É chamado com o número de canais do sistema como argumento.

Comunica com a aplicação através do ficheiro Var.cnf.

Processamento:

Chama Le_var.

Enquanto em modo de edição

 Selecciona variável a editar.

 Edita chamando Input.

Chama Escreve_var.

Dados:

Variáveis auxiliares e as variáveis do processo a configurar: Nome, Tipo (I ou O), Endereço no SAC, tamanho da fila respectiva, frequência de aquisição para variáveis de entrada (I), Arquivo (Sim ou Não) e os valores para alarme inferior, superior e set point se justificado.

Restrições:

O desenvolvimento da aplicação passa por aumentar as possibilidades de configuração ao processo e a flexibilidade com que são efectuadas. Deve ser fácil incluir mais variáveis de configuração neste módulo.

Módulo: Lê_var

Tipo:

Procedimento (entrada).

Funções:

Leitura a disco das configurações das variáveis do processo.

Subordinados:

Dependência:

É chamado na Configuração das Variáveis e na Inicialização da Exploração On-line.

Interfaces:

Devolve o buffer Configurações de Variáveis com os valores lidos do ficheiro.

Processamento:

Prepara nome do ficheiro a ler.

Abre o ficheiro para leitura.

Até ao fim do ficheiro

Copia conteúdo do ficheiro para o buffer.

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:**Módulo: Escreve_var****Tipo:**

Procedimento (saída).

Funções:

Escrita em disco das configurações especificadas para o processo.

Subordinados:**Dependência:**

É chamado por Conf_variáveis na Configuração das Variáveis/Processo.

Interfaces:

Na chamada são passadas as variáveis que são escritas no ficheiro Var.cnf.

Processamento:

Prepara nome do ficheiro.

Abre ou cria o ficheiro.

Escreve em ficheiro o conteúdo das Configurações das Variáveis.

Fecha o ficheiro.

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:**Módulo: Conf_variáveis_s****Tipo:**

Procedimentos (interface)

Funções:

Permitir a configuração de variáveis globais ao sistema.

Incluem-se as seguintes variáveis:

- nr_canais;
- simulação;
- password 1;
- password 2;
- frequência de aquisição.

Subordinados:

Lê_var_s, Input, Escreve_var_s.

Dependência:

É chamado pelo Gestor da Aplicação por selecção do utilizador.

Interfaces:

Comunica com a aplicação através do ficheiro Varsis.cnf.

Processamento:

Chama Le_var_s.

Enquanto em modo de edição

 Selecciona variável a editar.

 Edita chamando Input.

Chama Escreve_var_s.

Dados:

Variáveis auxiliares.

Restrições:

Deve ser fácil incluir mais variáveis de configuração global do sistema.

Módulo: Lê_var_s**Tipo:**

Procedimento (entrada).

Funções:

Executa leitura a disco das configurações do sistema guardadas no ficheiro Varsis.cnf.

Subordinados:**Dependência:**

É chamado por Conf_variáveis_s no início da Configuração das Variáveis/Sistema.

Interfaces:

Devolve as variáveis do sistema actualizadas.

Processamento:

Prepara nome do ficheiro.

Abre o ficheiro.

Até ao fim do ficheiro,

 Copia conteúdo do ficheiro para as variáveis respectivas.

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:**Módulo: Escreve_var_s****Tipo:**

Procedimento (saída).

Funções:

Escrita em disco das configurações especificadas para o sistema.

Subordinados:**Dependência:**

É chamado por Conf_variáveis_s na Configuração das Variáveis/Sistema.

Interfaces:

Na chamada são passadas as variáveis de sistema que são escritas no ficheiro Varsis.cnf.

Processamento:

Prepara nome do ficheiro.

Abre ou cria o ficheiro.

Escreve em ficheiro o conteúdo das Configurações.

Fecha o ficheiro.

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:

Módulo: Conf_gráfico

Tipo:

Procedimento (interface)

Funções:

Implementa a interacção com o utilizador que possibilita a especificação ou alteração das características dos gráficos a serem produzidos em tempo real. Permite uma pré-visualização do gráfico configurado usando valores de defeito.

Subordinados:

Lê_gráfico, Input_numérico, Input_string, Escreve_gráfico e Gráfico.

Dependência:

É chamado pelo Gestor da Aplicação e pelo Gestor On-line, conforme selecção do utilizador.

Interface:

Como argumento recebe o número do gráfico a configurar.

Processamento:

Lê_gráfico

Enquanto o modo 1 = configurar

 Enquanto modo 2 = editor

 Selecciona a variável a editar.

 Edita chamando Input_Numérico ou Input_String.

 Permite mudar de modo de funcionamento.

 Enquanto modo 2 = visualizar

 Inicia modo gráfico

 Chama gráfico para gerar o gráfico configurado.

 Permite mudar de modo

Chama Escreve_gráfico

Dados:

Restrições:

A comunicação com a aplicação faz-se pelos ficheiros de configuração dos gráficos.

Módulo: Lê_gráfico

Tipo:

Procedimento (entrada)

Funções:

Executa leitura a disco das configurações existentes para um dado gráfico.

Subordinados:**Dependência:**

É chamado por Conf_gráfico no início da configuração de um gráfico, quer off-line quer on-line.

Interfaces:

É passado, como argumento do módulo, o número do gráfico a ler do disco e como variáveis globais são usadas as configurações do gráfico a ler de disco (pontos, linhas, fórmulas, legendas e linha_canal).

Processamento:

Preparar nome do ficheiro a partir do número do gráfico.

Abrir o ficheiro para leitura.

Ler do disco as várias variáveis de configuração (estrutura pontos, arrays linhas, fórmulas, legendas e linha_canal).

Dados:

Variáveis temporárias para manusear o ficheiro.

Restrições:

O acesso a disco para leitura das configurações deve ser uma operação rápida.

Módulo: Escreve_gráfico**Tipo:**

Procedimento (saída)

Funções:

Grava em disco as configurações especificadas para um dado gráfico.

Subordinados:**Dependência:**

É chamado por Conf_gráficos no fim da configuração de um gráfico, tanto em off-line como on-line.

Interfaces:

São passados como argumentos do módulo, o número do gráfico e as configurações do gráfico (pontos, linhas, fórmulas, legendas e linha-canál).

Processamento:

Prepara o nome do ficheiro a partir do número do gráfico.

Abre ou cria o ficheiro para escrita.

Escreve em disco as várias variáveis da configuração (pontos, legenda, fórmulas, linhas e linha-canál).

Reporta eventuais erros

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:

A operação de escrita em disco deve ser rápida já que também é usada na Exploração On-line.

Módulo: Conf_tabela**Tipo:**

Procedimento (interface).

Funções:

Entrada ou alteração das configurações para uma dada tabela de dados a ser produzida em tempo real.

Subordinados:

Input_numérico, Lê_tabela e Escreve_tabela.

Dependência:

É chamado pelo Gestor da Aplicação e pelo Gestor On-line, conforme selecção do utilizador.

Interfaces:

Como argumento recebe o número da tabela. A comunicação com a aplicação faz-se pelos ficheiros de configuração das tabelas.

Processamento:

Chama Lê_tabela.

Enquanto edição,

 Selecciona a variável a editar;

 Edita chamando Input_numérico.

Chama Escreve_tabela.

Dados:

Variáveis de configuração das tabelas e variáveis auxiliares.

Restrições:**Módulo: Lê_tabela****Tipo:**

Procedimento (entrada).

Funções:

Executa leitura a disco das configurações existentes para uma dada tabela guardadas no ficheiro Tab#.cnf.

Subordinados:**Dependência:**

É chamado por Conf_tabela no início da configuração de uma tabela quer off-line quer on-line.

Interfaces:

Como argumento é passado o número da tabela e são devolvidas as configurações da tabela actualizadas com os valores do ficheiro.

Processamento:

Prepara o nome do ficheiro a partir do número da tabela.

Abre o ficheiro para leitura.

Copia os valores do ficheiro para as variáveis de configuração.

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:

O acesso a disco para leitura das configurações deve ser uma operação já que é usada na Exploração On-line.

Módulo: Escreve_tabela**Tipo:**

Procedimento (saída).

Funções:

Grava em disco as configurações especificadas para uma dada tabela.

Subordinados:**Dependência:**

É chamado por Conf_tabela ao terminar a configuração de cada tabela quer off-line quer on-line.

Interfaces:

Como argumento são passados o número da tabela e as variáveis de configuração.

Processamento:

Prepara nome do ficheiro a partir do nome da tabela.

Abre ou cria o ficheiro.

Escreve em disco as variáveis de configuração da tabela (nr. de valores, coluna 1, coluna 2, coluna 3, coluna 4).

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:

A operação de escrita em disco deve ser rápida já que também pode ser usada na Exploração On-line.

Módulo: Validar Configurações**Tipo:**

Sub-programa executável (processamento/interface)

Funções:

Verificar as configurações da aplicação gravadas em disco, detectando possíveis erros ou inconsistências. Ao longo da verificação são emitidas mensagens de erro ou de aviso ao utilizador. Evita erros durante a Exploração On-line.

Algumas validações a efectuar:

Existência dos ficheiros de configurações.

Se o número de valores das linhas dos gráficos é menor ou igual ao tamanho da respectiva fila circular.

Subordinados:**Dependência:**

É chamado pelo Gestor da Aplicação.

Interfaces:

É chamado sem parâmetros de entrada. Usa os ficheiros de configurações para recolher a informação que precisa e imprime as mensagens resultantes no écran.

Processamento:

Percorre os vários tipos de ficheiros de configurações que existem,

Se encontra ficheiro avisa "Existe o ficheiro...",

Senão existe esse tipo de ficheiro avisa "Atenção: Não existem ficheiros de..."

Percorre as configurações dos vários gráficos,

Se o número de valores de cada linha for maior que o tamanho do canal respectivo avisa "Atenção Inconsistência: o número de valores por linha é maior que o tamanho..."

Dados:

Variáveis auxiliares.

Restrições:

Pretende-se libertar a execução on-line de repetidas funções de teste à possibilidade de erros nas configurações. Deverá ser fácil adicionar novas validações a este programa.

Módulo: Inicialização

Tipo:

Procedimento (processamento).

Funções:

Instalar rotinas de serviço às interrupções para Monitorização e Registo.

Ler as configurações das variáveis de ficheiro para o buffer Configurações de Variáveis.

Calibrar o hardware de aquisição de dados.

Dimensionar e criar os buffers Tabela, Lista_alarmes e variáveis auxiliares.

Subordinados:

Monit_calibra.

Monit_instala.

Reg_instala.

Lê_var.

Dependência:

É chamado pelo Gestor da Aplicação.

Interfaces:

Partilha as variáveis globais a exploração on-line.

Devolve todos os buffers e variáveis que dimensiona devidamente inicializadas.

Processamento:

Lê_var.

Alocação dinâmica dos vários buffers e variáveis.

Inicializações.

Reg_instala.

Monit_calibra.

Monit_instala.

Dados:

Só usa dados externos.

Restrições:

Antecede a execução do Gestor On-Line.

Módulo: Monit_calibra

Tipo:

Procedimento (processamento).

Funções:

Promover a auto-calibração do sistema Keithley.

Subordinados:

Dependência:

É chamada na Inicialização da Exploração On-line.

Interfaces:

São usadas as definições constantes do header kh500.h.

Processamento:

Seleção do modo de calibração e início do processo.

Aguarda até fim do processo de calibração.

Dados:

Variáveis auxiliares.

Restrições:

A auto-calibração deve ser executada antes de se iniciar a utilização do SAC.

Procedimento específico do SAC Keithley 500.

Módulo: Monit_instala**Tipo:**

Procedimento (processamento).

Funções:

Inicia a execução em background da Monitorização.

Subordinados:**Dependência:**

É chamada na Inicialização da Exploração On-line.

Interfaces:

São usados os dados globais do SAC. Precisa do endereço da rotina Monitorização e da frequência pretendida para programar o relógio do SAC. Devolve o conteúdo da posição OxOA da tabela na variável endereço.OA.

Processamento:

Programa o relógio do SAC Keithley para periodicamente gerar interrupções tipo IRQ2.

Desactiva geração de interrupções

Instala a rotina Monit_int de serviço às interrupções para monitorização, na posição HOA da tabela de vectores do DOS.

Activa geração de interrupções

Activa a geração das interrupções pelo SAC.

Dados:

Não usa dados internos.

Restrições:

A interrupção que implementa a Monitorização deve ser de maior prioridade, já que os dados adquiridos são a matéria prima da aplicação.

A geração de interrupções é assegurada por um relógio do SAC.

Este módulo é específico do SAC keithley 500.

É um procedimento específico do hardware de aquisição de dados -Keithley 500.

Módulo: Reg_instala**Tipo:**

Procedimento (processamento).

Funções:

Inicia a execução em background do Registo.

Subordinados:**Dependência:**

É chamada na Inicialização da Exploração On-line.

Interfaces:

Precisa do endereço da rotina Registo de serviço às interrupções.

Devolve o conteúdo da posição 0x28 da tabela de vectores na variável endereço_28.

Processamento:

Desactiva geração de interrupções.

Guarda conteúdo da posição 0x28 na variável endereço_28.

Coloca nesta posição o endereço da rotina Registo que passa a ser chamada pela interrupção 0x28 do DOS.

Activa geração de interrupções.

Dados:

Não usa dados internos.

Restrições:

A interrupção que implementa o registo deve ser de baixa prioridade.

Módulo: Monitorização**Tipo:**

Rotina de serviço a interrupções (processamento/entrada)

Funções:

Implementa as funções de aquisição de dados do processo e de identificação de situações de alarme.

Executa a leitura do tempo e do valor aos canais do SAC configurados como de entrada analógica: selecções do canal, conversão A/D, leitura e inserção na fila circular respectiva. A selecção do canal inclui a especificação do ganho, da filtragem e da gama de leitura.

As leituras são efectuadas à frequência especificada para cada canal, múltipla da frequência das interrupções.

O valor da leitura do canal pode ser calibrado ($= K1 + K2 \times \text{Valor}$) antes de ser guardado na fila circular do buffer Tabela.

Os valores adquiridos são testados em relação aos limites inferiores e superiores especificados para o canal, gerando sinais de alarme caso sejam excedidos.

Subordinados:**Dependência:**

Uma vez instalada na Inicialização (Monit_instala), é activada por hardware (Int 0x0A-IRQ2), até ser desinstalada na Terminação (Monit_desinstala).

Interfaces:

Associado a um canal do SAC, correspondente a uma variável do processo, existe o seguinte conjunto de vectores que configura a Monitorização e fazem parte da estrutura do buffer cnf_variáveis:

— frequência — contêm a frequência de leitura do canal traduzida num número inteiro que indica o número de varrimentos que reparam cada leitura.

— par_M2A — contêm oito bits que parametrizam o funcionamento do SAC para a leitura do canal. No Keithley é usado para seleccionar o filtro, o ganho local e o modo de aquisição de dados do canal.

— **par_M2B** — contém oito bits que parametrizam o funcionamento do SAC para a leitura do canal. No Keithley é usado para seleccionar o módulo de hardware no SAC, ganho global e gama de trabalho.

— **end_módulo** — endereçamento do canal no SAC.

— **nr. canais** — guarda a sequência dos canais (menos internos), configurados para leitura.

— **contador** — vai registando para cada canal o número de varrimentos ao SAC já produzidos desde a última leitura. Quando esse número atingir a frequência, uma nova leitura deve efectuar-se.

— **K1 e K2** — são respectivamente os parâmetros de calibração do valor da leitura do canal ($K1 + K2 \times \text{Valor}$).

— **alarm_sup e alarm_inf** — estabelecem os limites para os valores aceitáveis como normais.

Os dados adquiridos são colocados na buffer Tabela e as situações de alarme eventualmente identificadas são sinalizadas no buffer Lista_alarmes.

Processamento:

Suspende a geração de interrupções pelo SAC. Lê o tempo actual em décimos de segundo e actualiza o canal Tempo (canal 0). Decrementa o contador do canal actual. Se o contador atingir o valor 0:

- Repõe o contador do canal actual na frequência configurada.
- Aguarda até à última conversão (canal 0) ser concluída.
- Prepara leitura do conversor e inserção na fila circular.
- Lê o valor do conversor
- Efectua as operações de calibração e insere na fila circular o valor da leitura do canal 0 e respectivo tempo de leitura.
- Actualiza o canal Tempo de Simulação com o tempo de leitura.
- Inicia a conversão para a leitura do canal 1.
- Efectua as selecções para o canal actual, o canal 2, a partir do conjunto de canais configurados para leitura.

- Actualiza os apontadores para os canais em processo de leitura.

Senão:

- Incrementa o canal actual.
- Se ultrapassar o número total de canais o canal actual volta a ser o primeiro.
- Activa geração de interrupções pelo SAC.
- Executa alguma função que existisse na posição 0x0A da tabela de vectores.

A rotina termina a interrupção passando o processamento para a instrução interrompida.

Dados:

São usadas variáveis auxiliares para identificar os canais em processo de leitura (canal 0, canal 1 e canal 2) e para o número do canal_actual.

Restrições:

O módulo é específico para o sistema de aquisição Keithley 500. Neste SAC a aquisição de dados de um canal é executada em dois tempos, selecção e acondicionamento com até 0.8 μ s de settling time e conversão em 20 μ s. Todas operações sobre o SAC têm settling time que têm que ser respeitados.

Este módulo deve ser muito eficiente, sendo executado no mínimo tempo possível.

Para garantir a melhor utilização do recurso tempo, deve procurar-se paralelizar as operações no SAC, implementando um piping de tarefas realmente simultaneas. Assim enquanto o canal servido no tempo 0, (canal 0) é lido do buffer de saída do conversor, o canal servido no tempo 1 (canal 1) está a ser convertido e o canal que está a ser servido no tempo actual (canal 2), está a ser seleccionado e acondicionado.

Módulo: Registo

Tipo:

Procedimento de serviço a interrupções (processamento/saída).

Funções:

Arquivar dados do buffer Tabela nos ficheiros correspondentes a cada canal configurado para escrita em disco.

Responder a sinais de alarme emitindo mensagens do utilizador, em qualquer écran.

Subordinados:

Dependência:

É instalada na Inicialização e fica activa em background durante a Exploração On-Line até à Terminação.

Interfaces:

Usa os buffers de Tabela e Configurações de Variáveis para receber informações dos outros procedimentos. Mensagens de alarme e erros são impressos no écran.

Processamento:

Percorre todos os canais. Para cada um:

- Se houver sinal de alarme: produz mensagem ao utilizador.
- Se o canal corresponder a uma variável configurada para ser arquivada e se a fila circular respectiva contiver o número mínimo de dados (metade da fila):
 - Produz o nome do ficheiro de dados.
 - Se o ficheiro não existir procede à sua criação.
 - Abre o ficheiro para escrita.
 - Escreve em disco metade da fila circular a partir da última escrita.
 - Fecha o ficheiro.

Dados:

Variáveis auxiliares para manusear ficheiros.

Restrições:

O fluxo de saída de dados da tabela, deve ser maior que o de entrada, assegurado pela Monitorização.

A interrupção para Monitorização deve ter uma prioridade maior que a do registo.

Para minimizar o número de escritas em disco, esta realiza-se para, pelo menos, metade dos dados de cada fila. Assim, para um canal, a frequência de registo pode ser menor que a de monitorização por um valor de: tamanho [canal] / 2 vezes, o que para um tamanho de fila de 200 quer dizer 100 vezes menor.

Módulo: Mostra_alarmes

Tipo:

Procedimento (saída).

Funções:

Permite sinalizar todos os alarmes ocorridos e ainda não validados

Subordinados:

Dependência:

É chamada pelo Gestor On-line conforme selecção do utilizador.

Interfaces:

Usa como variáveis globais o buffer Lista_alarmes e respectivas variáveis auxiliares.

Processamento:

Percorre o número de alarmes existentes

Imprime no écran a caracterização da situação de alarme, tipo, canal e hora.

Dados:

Variáveis auxiliares.

Restrições:

Módulo: Valida_alarmes

Tipo:

Procedimento (interface).

Funções:

Permite visualizar cada alarme pendente e validá-lo caso esteja ultrapassado.

Subordinados:

Dependência:

É chamada pelo Gestor On-line conforme selecção do utilizador.

Interfaces:

Usa como variáveis globais o buffer Lista_alarmes e respectivas variáveis auxiliares.

Processamento:

Percorre o número de alarmes existentes.

Visualiza as características do alarme.

Permite validação (sim ou não) pelo utilizador.

Dados:

Variáveis auxiliares.

Restrições:

Módulo: Mostra_gráfico

Tipo:

Procedimento (processamento)

Funções:

Interface com o gerador gráfico para produção de um gráfico sobre os dados mais recentes do processo.

Subordinado:

Gráfico, Lê_gráfico e Tabela_array.

Dependência:

É chamado por Gestor On-line.

Interfaces:

Partilha o buffer Tabela

É chamado com passagem do número do gráfico a produzir.

Processamento:

Chama Lê_gráfico.

Chama Tabela_array.

Prepara chamada do gerador.

Chama Gráfico.

Repõe modo gráfico anterior.

Dados:

Variáveis de configuração de um gráfico.

Restrições:

Assegura independência do gerador gráfico usado.

Módulo: Tabela_array

Tipo:

Procedimento (processamento).

Funções:

Assegura interface entre a geração de gráficos e o buffer Tabela de dados do processo. Os últimos valores de cada fila circular são copiados para os vectores respectivos sobre os quais o gerador do gráfico opera.

Subordinados:

Dependência:

É chamado por Mostra_gráfico depois de serem lidas as configurações para um dado gráfico

Interfaces:

É chamado com o número do gráfico actual.

São usadas variáveis globais de configuração do gráfico em causa (pontos, linha-canal), os vectores de valores (dados) e a tabela de dados do processo (Tabela).

Processamento:

Para as várias linhas que constituem o gráfico:

Percorre o número de pontos que constituem a linha

Determina a correspondente posição na Tabela

Copia o valor e o tempo da Tabela para Dados.

Dados:

Variáveis auxiliares.

Restrições:

Para manter a independência entre o gerador de gráficos e aplicação é proposta esta rotina, que copia os dados do formato usado na aplicação (Tabela) para o usado no gerador, neste caso um vector tridimensional. O gerador aqui usado é o Gráfico que teve um desenvolvimento prévio e independente da aplicação SP. Esta rotina é executada na sequência de um pedido de visualização de um gráfico pelo utilizador, pelo que deve ser optimizada quanto ao tempo de execução.

Módulo: Mostra_tabela

Tipo:

Procedimento (processamento).

Funções:

Gerador da tabela de dados do processo.

Subordinados:

Armação, Mostra_valor e Lê_tabela.

Dependência:

É chamado pelo Gestor On-line.

Interfaces:

Como argumento de chamada é passado o número da tabela. As configurações da tabela são lidas do disco. O buffer Tabela é passado como variável global.

Processamento:

Chama Lê_tabela.

Chama Armação.

Para cada coluna da tabela,

 Percorre os valores do canal respectivo;

 Chama Mostra_valor.

Dados:

Variáveis auxiliares e variáveis de configuração das tabelas (nr. de valores, coluna 1, coluna 2, coluna 3, e coluna 4).

Restrições:

Deve ser de execução rápida já que é paralela a outras funções da Exploração On-line.

Módulo: Armação**Tipo:**

Procedimento (interface).

Funções:

Imprime o aspecto gráfico pretendido para a tabela.

Subordinados:

É chamado por Mostra_tabela.

Dependência:**Interfaces:**

Não partilha dados com a aplicação.

Processamento:**Dados:**

Variáveis auxiliares.

Restrições:

Isola uma tarefa de acesso a écran e constante.

Módulo: Mostra_valor**Tipo:**

Procedimento (interface).

Funções:

Imprime um valor na devida posição na tabela e com o arranjo gráfico pretendido.

Subordinados:**Dependência:**

É chamado por Mostra_tabela.

Interfaces:

Como argumento são passados o valor a imprimir e a respectiva posição.

Processamento:**Dados:**

Não precisa de variáveis auxiliares.

Restrições:

Deve manter Mostra_tabela independente dos acessos ao écran.

Módulo: Mostra_sinóptico**Tipo:**

Procedimento (interface/processamento)

Funções:

Apresenta um quadro sinóptico animado de acordo com o seleccionado pelo utilizador.

Assegura a interface com o utilizador nesta opção de visualização de sinópticos, permitindo a selecção de diferentes sinópticos por F Keys.

Subordinados:

Inicia_modos_gráfico, Lê_parâmetros, Lê_imagem, Anima_sinóptico.

Dependência:

É chamado pelo Gestor On-line conforme selecção do utilizador.

Interfaces:

É passado o número do quadro sinóptico que começa por representar e como variável global o buffer Tabela.

Processamento:

São feitas inicializações para a interface gráfica, sendo chamado Inicia_modos_gráfico.

Enquanto a selecção do utilizador for diferente de Saída:

Faz leitura dos parâmetros, chama Lê_parâmetro,

Faz leitura da imagem, chama Lê_imagem,

Enquanto não for seleccionado outro sinóptico:

Chama Anima_sinóptico,

Oferece selecções de outro sinóptico (F1 a F8) ou Saída (Esc)

Dados:

Os dados comuns ao Lê_parâmetro e ao Anima_sinóptico são definidos globalmente neste módulo.

Restrições:

A animação do sinóptico deve ser feita de uma forma tão rápida quanto possível.

Deve funcionar como módulo de interface em relação às configurações produzidas pelo Editor embora de execução completamente independente. O módulo Anima_sinóptico é o interpretador dos ficheiros de parâmetros produzidos pelo Editor.

Módulo: Lê_imagem**Tipo:**

Procedimento (entrada).

Funções:

Efectua leitura a disco de um ficheiro imagem e coloca-a no écran.

Subordinados:**Dependência:**

É chamado por Mostra_sinóptico

Interfaces:

Os dados de entrada são o nome do ficheiro, a resolução do écran em x e y e o número de bytes por pixel, usado para caracterizar as imagens do écran.

Processamento:

Calcula a dimensão de cada parte do écran a ser processada.

Abre o ficheiro imagem (#.IMG).

Aloca dinamicamente memória para um buffer temporário.

Lê a 1ª parte do ficheiro para o buffer.

Coloca o buffer no écran.

Lê a 2ª parte do ficheiro para o buffer.

Coloca o buffer no écran.

Lê a 3ª parte do ficheiro para o buffer.

Coloca o buffer no écran.

Liberta a memória ocupada pelo buffer.

Fecha o ficheiro.

Dados:

Variáveis e buffer auxiliares.

Restrições:

A rotina deve ser independente do sistema gráfico.

Como um ficheiro imagem é grande (ex:110 KB) a sua leitura e correspondente colocação no écran será feita em três tempos. Reduz-se assim um pico de consumo de memória RAM.

Módulo: Conf_var_saída

Módulo de concepção idêntica ao Conf_variáveis, mas em que apenas as variáveis previamente especificadas como de saída (output) podem ser reconfiguradas.

Para uma destas variáveis ser enviada para um dado sistema de aquisição e controlo (SAC), é chamado o módulo Envia_var_saída.

Módulo: Envia_var_saída**Tipo:**

Procedimento (processamento/saída).

Funções:

De acordo com o configurado para uma dada variável de saída é enviado para um dado SAC o valor pretendido para esta variável, por escrita nos endereços de memória configurados.

Subordinados:**Dependência:**

É chamada por Conf_var_saída.

Interfaces:

Precisa das configurações da variável a enviar e dos endereços de memória do SAC.

Processamento:

De acordo com o SAC utilizado envia para canais de saída os valores configurados para a variável.

Dados:

Variáveis auxiliares para as várias configurações.

Restrições:

A actuação produzida não pode interferir com o processo de monitorização a decorrer em background. O procedimento deve ser executado em foreground.

Módulo: Envia_ficheiro

Tipo:

Procedimento (interface/saída).

Função:

Permite o enviar pela porta série um ficheiro de comandos de um dado controlador previamente escrito para executar uma dada actuação sobre o processo.

Subordinados:

Input_string.

Dependência:

É chamado pelo Gestor On-Line por selecção pelo utilizador da correspondente opção em Operar.

Interface:

Não partilha informação com a aplicação.

Processamento:

Chama Input_string para entrada do nome do ficheiro.

Testa se o ficheiro existe,

Se existir

Envia-o pela porta série

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o ficheiro.

Restrições:

O procedimento deve ser executado em foreground, em paralelo com a Monitorização e o Registo. Pode ser um módulo de execução independente.

Módulo: Envia_comando

Tipo:

Procedimento (interface/saída).

Funções:

Permite o envio pela porta de série de um comando para um dado controlador de processo.

Subordinados:

Input_string.

Dependência:

É chamado pelo Gestor On-line por selecção pelo utilizador da correspondente opção em Operar.

Interfaces:

Não partilha informação com a aplicação.

Processamento:

Chama Input_string para entrada do comando a enviar.

Envia o comando pela porta série.

Reporta eventuais erros.

Dados:

Variáveis auxiliares para manusear o comando.

Restrições:

O procedimento deve ser executado em foreground, em paralelo com a Monitorização e o Registo. Pode ser um módulo de execução independente.

Módulo: Terminação**Tipo:**

Procedimento (processamento).

Funções:

Termina a Exploração On-Line.

Chama Monit_desinstala e Reg_desinstala que desactivam os processos de execução em background.

Liberta a memória alocada dinamicamente.

Transfere o controlo para o Gestor da Aplicação.

Subordinados:

Monit_desinstalação.

Reg_desinstala.

Dependência:

É chamada pelo Gestor On-Line conforme selecção do utilizador.

Interfaces:**Processamento:**

Monit_desinstalação

Reg_desinstala

Libertar o espaço ocupado em RAM pelas variáveis alocadas dinamicamente.

Dados:**Restrições:**

No módulo Inicialização é alocada a memória que aqui deve ser libertada.

Módulo: Monit_desinstala**Tipo:**

Procedimento (processamento).

Funções:

Terminar o processo em background de Monitorização desinstalando a rotina de serviço às interrupções.

Subordinados:**Dependência:**

É chamado pelo módulo Terminação da Exploração On-line.

Interfaces:

Usa a variável endereço OA.

Processamento:

Desactiva interrupções

Repõe o conteúdo na posição 0xOA da tabela de vectores com o valor existente antes da instalação de Monitorização e guardado na variável endereço_OA.

Activa interrupções.

Dados:

Não usa dados internos

Restrições:

Realiza o processamento complementar ao de Monit_instala.

Módulo: Reg_desinstala**Tipo:**

Procedimento (processamento).

Funções:

Termina o processo em background do Registo, desinstalando a rotina de serviço às interrupções.

Subordinados:**Dependência:**

É chamado pelo módulo da Exploração On-line.

Interfaces:

Usa a variável endereço_28.

Processamento:

Desactiva interrupções.

Repõe o conteúdo da posição 0x28 da tabela de vectores do DOS com o valor existente antes da instalação do Registo e guardado na variável endereço_28.

Activa interrupções.

Dados:

Não usa dados internos.

Restrições:

Realiza o processamento complementar ao do Reg_instala.

A execução de Reg_desinstala não garante que o processamento em curso por Registo tenha sido completo.

Módulo: Selecção de canal**Tipo:**

Sub-programa executável (interface).

Funções:

Implemente Interface com o utilizador para selecção do ficheiro de dados a converter.

Chama o procedimento Conversão de dados para o ficheiro seleccionado.

Subordinados:

Chama Input (entrada de dados com edição) e Conversão de dados.

Dependência:

É chamado pelo Gestor da Aplicação por a escolha do utilizador.

Também pode ser executado directamente no DOS.

Interfaces:

Os nomes dos ficheiros origem e destino são passados ao módulo Conversão de dados.

A comunicação com a aplicação é feita pelos ficheiros de dados do processo.

Processamento:

Pede ao utilizador o número do canal a converter.

Prepara parâmetros de chamada da Conversão de dados, ficheiro origem canal#.dad e ficheiro destino canal#.txt.

Chama conversão de dados.

Dados:**Restrições:**

Deve ser um procedimento de execução independente do resto da aplicação. Terminando liberta a memória RAM que ocupava.

Módulo: Conversão de dados**Tipo:**

Procedimento (processamento/saída).

Funções:

Converte um ficheiro binário tipo CANAL#.DAD para um ficheiro ASCII tipo CANAL#.TXT.

Subordinados:**Dependência:**

É chamado pela Selecção de dados.

Interface:

Os parâmetros necessários acompanham a chamada do procedimento (Convert CANAL#.DAD CANAL#.TXT) no sistema operativo.

Processamento:

Testa se o ficheiro a converter existe.

Senão emite mensagem de erro e termina.

Abre o ficheiro origem (binário).

Abre ou cria o ficheiro destino (texto).

Desde o início do ficheiro até encontrar o fim:

- Lê os dados do ficheiro origem.
- Escreve os dados no ficheiro destino.

Fecha os dois ficheiros.

Emite mensagem de fim de conversão.

Dados:**Restrições:**

2. Módulos implementados

Os vários módulos aqui especificados foram implementados e constituem a aplicação protótipo. Apresenta-se uma listagem dos módulos concebidos e das correspondentes rotinas em C:

Gestor da Aplicação — main ()
Gestor On-Line — g_on-line ()
Lê_password — le_pass ()
Testa_password — testa_pass ()
Editor (alterações a produzir)
Escreve_imagem — escreve_img ()
Conf_variáveis — cfg_var ()
Lê_var — le_var ()
Escreve_var — escreve_var ()
Conf_variáveis_s — cnf_var_sis ()
Lê_var_s — le_var_sis ()
Escreve_var_s — escreve_var_sis ()
Conf_gráfico — cfg_graf ()
Lê_gráfico — lê_graf
Escreve_gráfico — escreve_graf ()
Conf_tabela — cfg_tabela ()
Lê_tabela — le_tabela ()
Escreve_tabela — escreve_tab ()
Validar Configurações — valida ()
Inicialização — inicia ()
Monit_calibra — aq_calibra ()
Monit_instala — aq_instala ()
Reg_instala — reg_instala ()
Monitorização — aq_int ()
Registo — reg_int ()
Mostra_alarmes — mostra_alarm ()

Valida_alarmes — valida_alarm ()
 Mostra_gráfico — mostra_grafico ()
 Tabela_array — tabela_array ()
 Mostra_tabela — mostra_tab ()
 Armação — armacao ()
 Mostra_valor — mostra_valor ()
 Mostra_sinóptico — mostra_sinóptico ()
 Lê_imagem — le_imagem ()
 Conf_var_saída — conf_var_s
 Envia_var_saída — envia_var_s ()
 Envia_ficheiro — a desenvolver
 Envia_comando — a desenvolver
 Terminação — termina ()
 Monit_desinstala — aq_desinstala ()
 Reg_desinstala — reg_desinstala ()
 Selecção de canal — main ()
 Conversão de dados — convert ()

Módulos pré-existent:

Anima_sinóptico
 Escreve_parâmetros
 Inicia_modos_gráfico
 Lê_parâmetro
 Input
 Input_numérico
 Input_string
 Gráfico
 Editor

Alguns módulos necessários já existiam implementados e apenas exigiram algum desenvolvimento adicional:

- Editor.exe, configuração de sinópticos (Beatriz Ribeiro)

- Graf.obj., gerador de gráficos (Pedro Guedes, Eduardo Luís Cardoso)
- Actpar.obj, actualiza sinópticos (Beatriz Ribeiro, Eduardo Luís Cardoso)

Foi também usado o conjunto de módulos pré-existentes que implementam tarefas de uso geral necessárias a qualquer aplicação:

- menus.obj, interface por sistema de menus (Carlos Sousa).
- exp.obj, interpreta e calcula fórmulas matemáticas (Carlos Sousa).
- rato.obj, utilização geral do rato (Pedro Guedes).
- input.obj, entrada formatada de dados (Carlos Sousa).
- in.obj, entrada formatada de dados (Carlos Sousa).

APÊNDICE D

Descrição Técnica do Sistema de Aquisição de Dados e Controlo Keithley - Series 500

ÍNDICE

	Pag.
1. Introdução.....	3
2. <i>Mother Board</i>	5
3. Interface IBIN-A PC.....	6
4. Módulo AMM2.....	8
4.1. Operação.....	8
4.2. Comandos.....	10
5. Módulo AIM7.....	18
5.1. Comandos.....	18

1. Introdução

O modelo 500-Series Measurement and Control System da Keithley [47] é um sistema de hardware projectado para constituir uma interface entre PCs e o meio exterior para aquisição de dados e controlo de processos, configurável a diferentes situações.

O sistema é constituído por uma placa da interface a colocar no bus do PC que assegura a comunicação entre o PC e o sistema externo constituído por uma placa de controlo (mother board) e por um bus com 10 slots aonde podem ser instaladas diferentes placas para funções como: entradas e saídas analógicas, entradas e saídas digitais, condicionamento e conversão de sinais, entradas específicas para termopares, sinais em corrente, e saídas específicas para controlo de motores, etc.

Na figura 1.1 encontra-se representado um diagrama geral do sistema Series 500.

A comunicação com o sistema é feita através de uma área de memória (memory-mapped input/output system) e para cada placa estão disponíveis, ao programador no PC, comandos de operação e controlo.

Associados a cada placa existem disponíveis ao programador no PC comandos para operar o sistema.

No capítulo 2 são descritas as funções da placa principal (mother board) do sistema externo. No capítulo 3 é apresentada a placa IBIN - A PC a colocar no bus do PC, a forma de ser configurada e os comandos disponíveis ao programador. No capítulo 4 e 5 são descritas duas placas específicas a instalar no sistema externo, o AMM2 e o AIM7 respectivamente. O AMM2 assegura condicionamento e conversão A/D de sinais para o sistema a par de aceitar entradas analógicas, e o AIM7 aceita entradas analógicas específicas de sensores de temperatura.

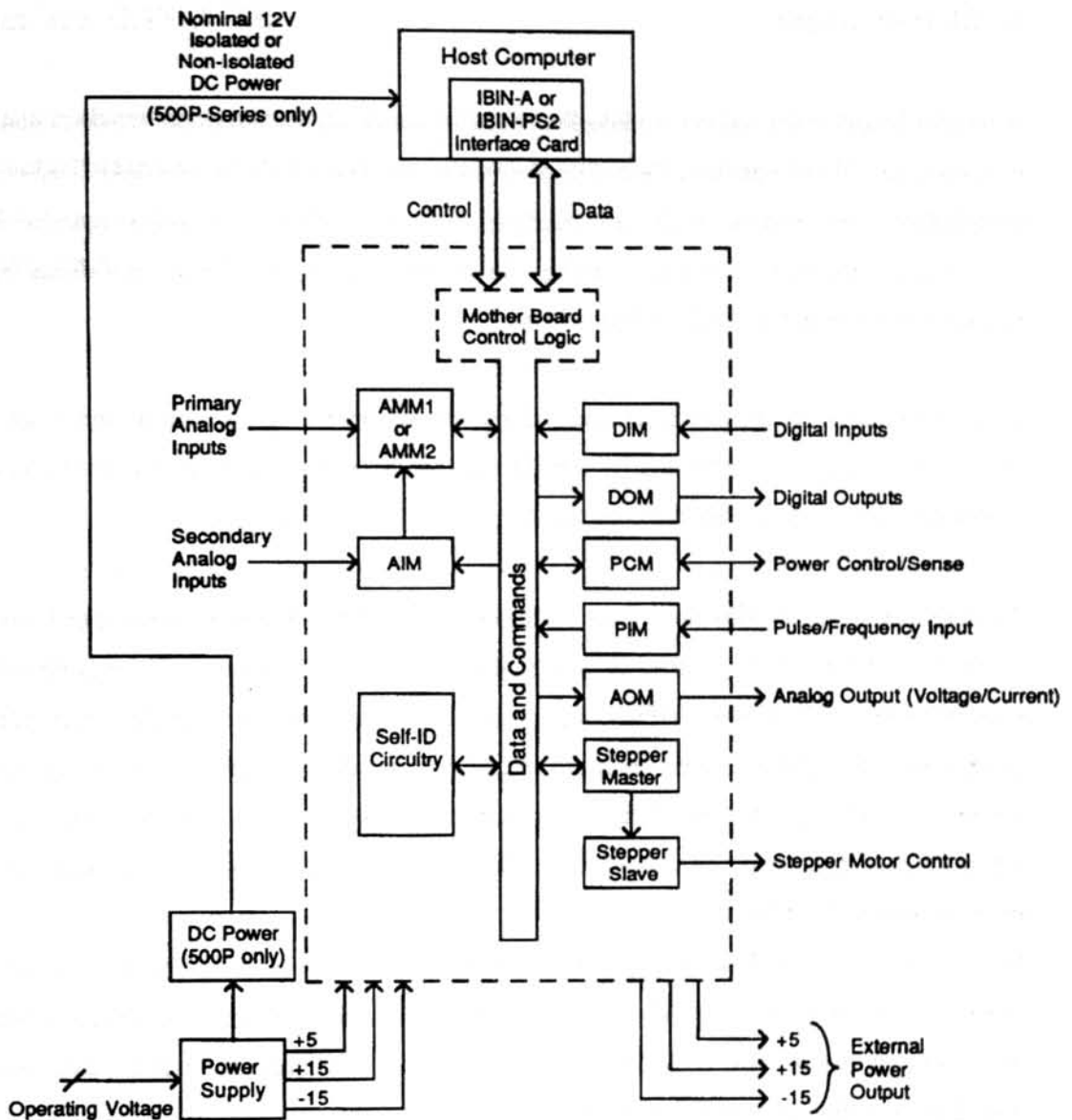


Figura 1.1. Diagrama geral do sistema da Series 500.

2. Mother Board

A mother board actua como uma interface entre todos os outros componentes do sistema: ligações, painel frontal, interface do computador e módulos da Series 500. Todos os comandos provenientes do computador são descodificados e enviados através dos circuitos da mother board para a slot do módulo especificado (os módulos são introduzidos numa das 10 slots disponíveis).

Os componentes da mother board podem dividir-se em três grupos funcionais: as 10 slots que aceitam os módulos, o circuito do sistema de controlo e o circuito de identificação automática (Self-ID circuitry).

As slots estão numeradas de 1 a 10 e têm como função estabelecer a ligação entre a mother board e os módulos. A cada slot estão associados dois comandos de endereçamento (CMDA e CMDB) que possibilitam ao computador dirigir a actividade dos módulos instalados. Estes comandos são específicos de cada slot e variam em função do módulo instalado. As três primeiras slots têm disponíveis comandos adicionais: a slot 1 tem dois comandos de endereçamento adicionais (CMDC e CMDD) e as slots 2 e 3 mais um comando (CMDC).

Se utilizarmos os módulos AMM1 ou AMM2, estes devem estar instalados na slot 1. Note-se que, no entanto, se não for necessário proceder a entradas analógicas, a slot 1 pode ser ocupada por qualquer outro módulo de conversão digital-analógica, entrada/saída digitais ou controlo ac/dc.

O circuito do sistema de controlo é o responsável pela geração de comandos. Este sistema descodifica a informação transmitida pelo computador e direciona os comandos para as slots apropriadas.

O circuito de identificação automática possibilita a utilização de software para identificar a presença e a localização (slot) dos módulos utilizados.

3. Interface IBIN-A PC

A IBIN-A funciona como uma interface entre o computador e a Series 500.

O endereço da interface na memória do sistema é seleccionado pelo Switch S1. Este Switch é composto por oito switches e está normalmente configurado para a localização hex CFF80. Neste caso, a posição dos oito switches será:

1	2	3	4	5	6	7	8
off	off	on	on	off	off	off	off

A tabela 3.1. indica as localizações da memória utilizadas pela interface. Estes endereços correspondem às funções comando CMDA e comando CMDB que estão associadas aos módulos disponíveis na Series 500. Note que alguns destes módulos têm disponíveis também os comandos CMDC e CMDD para funções especiais.

Tabela 3.1. Funções e localizações.

	Funções	Localização
Slot 1	CMDA	xxx00
	CMDB	xxx01
	CMDC	xxx1A
	CMDD	xxx1B
Slot 2	CMDA	xxx02
	CMDB	xxx03
	CMDC	xxx18
Slot 3	CMDA	xxx04
	CMDB	xxx05
	CMDC	xxx19
Slot 4	CMDA	xxx06
	CMDB	xxx07
Slot 5	CMDA	xxx08
	CMDB	xxx09
Slot 6	CMDA	xxx0A
	CMDB	xxx0B
Slot 7	CMDA	xxx0C
	CMDB	xxx0D
Slot 8	CMDA	xxx0E
	CMDB	xxx0F
Slot 9	CMDA	xxx10
	CMDB	xxx11
Slot 10	CMDA	xxx12
	CMDB	xxx13
	R/W COUNTER 0	xxx40
	R/W COUNTER 1	xxx41
	R/W COUNTER 2	xxx42
	COUNTER CONTROL	xxx43
	TIMER GLOBAL	xxx60
	TIMER STATUS	xxx61
	CLEAR INTERRUPT	xxx62
	SET INT LEVEL	xxx63

4. Módulo AMM2

O módulo AMM2, inclui três funções da Series 500:

- i. Funciona como um módulo de entradas analógicas e implementa condicionamento de sinal local.
- ii. Selecciona e condiciona sinais analógicos de outros módulos da Series 500.
- iii. Efectua a conversão analógico-digital (A/D) das suas entradas, bem como das de outros módulos.

Localmente o módulo AMM2 aceita entradas analógicas num total de 16 single-ended ou 8 diferenciais. A configuração de entrada é controlada por software e os canais de entrada analógica podem ser condicionados através de um amplificador local de ganho programável (x1 ou x10).

O condicionamento global inclui um amplificador, igualmente controlado por software, de ganho programável de x1, x2, x5 ou x10. Todos os sinais analógicos ligados à Series 500 passam através do circuito global, quer sejam originários do módulo AMM2, quer de outro módulo do sistema.

Para a conversão A/D, o módulo AMM2 utiliza um conversor de 16 bits de aproximação sucessiva. Sendo o tempo máximo de conversão de apenas 20 μ seg, pelo que este sistema possibilita uma frequência de aquisição que pode atingir os 50 kHz.

Como foi especificado anteriormente, o módulo AMM2 terá de ser colocado necessariamente no slot 1 da mother board.

4.1. Operação

4.1.1. Condicionamento de sinal

A figura 4.1 mostra um diagrama de blocos simplificado do módulo AMM2. O módulo está dividido em seis secções: multiplexer local, amplificador local de ganho programável, multiplexer global, amplificador global de ganho programável, filtro passa-baixo programável e conversor A/D de 16 bits.

Os sinais de entrada locais dos canais 0 até 15 (módulo AMM2) são conduzidos até ao multiplexer local para selecção. Apenas um canal pode ser seleccionado a cada instante,

utilizando para isso o comando SELECT CHANNEL (ver 4.2.1.). O sinal do canal seleccionado é então conduzido através do amplificador local de ganho programável para o multiplexer global para nova selecção e condicionamento de sinal.

O multiplexer global selecciona um sinal entre os 10 slots do sistema. Desta maneira, sinais provenientes de qualquer um dos 10 slots podem ser seleccionados por software. O multiplexer global é controlado pelo comando SELECT SLOT, tal como será explicado posteriormente (ver 4.2.6.).

Depois de seleccionado o sinal, o amplificador global aplica um ganho de x1, x2, x5 ou x10. O sinal é conduzido finalmente através de um filtro de frequência seleccionável por software (100 ou 2 kHz).

Após este processo de condicionamento de sinal, este é conduzido ao conversor A/D de 16 bits para digitalização. Uma vez concluído o processo de conversão, o sinal digital, que representa o sinal analógico de entrada, é enviado ao computador via interface.

4.2. Comandos

Na tabela 4.1 estão sumarizados os comandos usados no módulo AMM2.:

Tabela 4.1. Comandos usados no módulo AMM2.

Comando	Endereço	Byte	Bits usados
SELECT CHANNEL	xxx80	CMDA (escrita)	D0-D3
SELECT LOCAL CHANNEL MODE	xxx80	CMDA (escrita)	D4
SELECT LOCAL GAIN	xxx80	CMDA (escrita)	D5
SELECT ACQUISITION MODE	xxx80	CMDA (escrita)	D6
SELECT FILTER	xxx80	CMDA (escrita)	D7
SELECT SLOT	xxx81	CMDB (escrita)	D0-D3
SELECT CMDA READ MODE	xxx81	CMDB (escrita)	D4
SELECT RANGE	xxx81	CMDB (escrita)	D5
SELECT GLOBAL GAIN	xxx81	CMDB (escrita)	D6-D7
RESET AND RECAL	xxx9A	CMDC (escrita)	Todos
A/D LOW DATA *	xxx80	CMDA (leitura)	Todos
A/D STATUS *	xxx80	CMDA (leitura)	Todos
A/D HIGH DATA	xxx81	CMDB (leitura)	Todos
A/D START	xxx9B	CMDD (escrita)	Todos
EOC STATUS	xxx9B	CMDD (leitura)	Todos

* Conforme o valor do comando SELECT CMDA READ MODE é seleccionado o tipo de informação que a leitura do CMDA oferece.

A indicação "xxx", na coluna de endereços, significa os três dígitos hexadecimais que formam o endereço de hardware que pode ser seleccionado por switches ou programado na interface. O endereço sugerido é &HCFF80, pelo que, "xxx"="&HCFF".

Select Channel, Local Gain, Filter, Acquisition Mode, and Channel Mode.

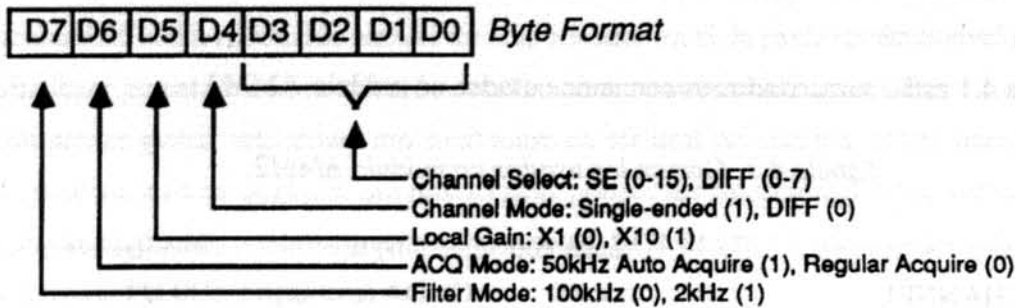


Figura 4.2. Formato de escrita do CMDA (endereço xxx80).

Select Slot, Range, Global Gain, and CMDA Read Mode.

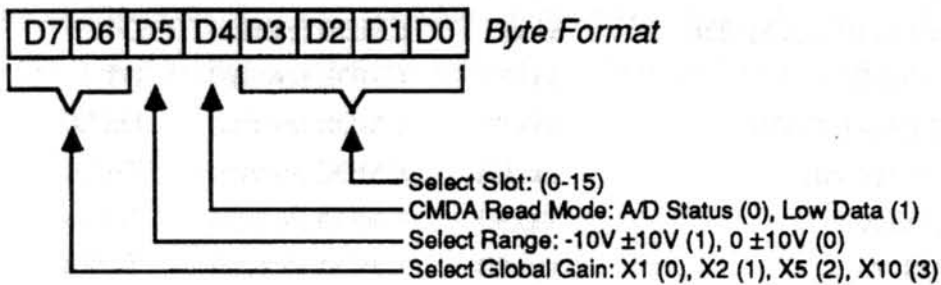


Figura 4.2. Formato de escrita do CMDB (endereço xxx81).

4.2.1. SELECT CHANNEL

Localização: xxx80

O comando SELECT CHANNEL é usado para controlar o multiplexer local do módulo AMM2, especificando qual dos canais locais será seleccionado para o processo de conversão A/D. Este comando actua apenas sobre as entradas locais do módulo AMM2 (não afecta os canais de entrada de outros módulos).

O comando SELECT CHANNEL deverá ser usado em conjunto com o comando SELECT SLOT (ver 4.2.6) para seleccionar os canais do módulo da slot 1 (posição do módulo AMM2).

Note que o número do canal ocupa os quatro bits menos significativos do CMDA (D0, D1, D2 e D3). Certifique-se que o número do canal escolhido combina da forma desejada com os quatro bits mais significativos antes de implementar o comando.

4.2.2. SELECT LOCAL CHANNEL MODE

Localização: xxx80

O comando SELECT LOCAL CHANNEL MODE controla a configuração do canal de entrada local do módulo AMM2. Os canais de entrada do AMM2 podem ser configurados como canais de 16 single-ended ou de 8 entradas analógicas diferenciais. Este comando é seleccionado atribuindo um valor ao bit da posição D4 do CMDA (figura 4.2.). O valor 1 configurará as entradas como single-ended; o valor 0 como diferenciais. Certifique-se que os outros bits do byte CMDA representam a configuração desejada antes de executar o comando.

4.2.3. SELECT LOCAL GAIN

Localização: xxx80

O ganho aplicado aos canais locais do módulo AMM2 pode ser programado através do valor do bit da posição D5 do CMDA. O valor 0 representará um ganho local de x1; o valor 1 aplicará um ganho local de x10. O ganho local pode ser alterado em qualquer altura antes da conversão ser iniciada.

Certifique-se que os outros bits do byte CMDA representam a configuração desejada antes de executar o comando.

4.2.4. SELECT ACQUISITION MODE

Localização: xxx80

O módulo AMM2 tem a capacidade de operar em dois modos de aquisição: o modo de aquisição regular e o modo de auto-aquisição a 50 kHz. O modo de aquisição é especificado através do bit da posição D6 do CMDA. Atribuindo o valor 0 será escolhido o modo de aquisição regular; o valor 1 especificará o modo de auto-aquisição a 50 kHz.

Para efectuar uma leitura analógica em modo de aquisição regular, devem ser especificados o slot, o canal e o ganho. Após decorrido um tempo de settling apropriado, o módulo AMM2 pode implementar o comando A/D START (ver 4.2.13.) inicializando o processo de digitalização. O comando EOC STATUS (ver 4.2.14) pode então ser utilizado para verificar o fim de conversão, após o qual, o valor já digitalizado pode ser lido. O processo de conversão demora aproximadamente 20 μ seg.

Logo após o comando A/D START ter sido implementado, a selecção do slot, canal e ganho pode ser especificada de imediato para uma nova leitura. Isto permitirá que o tempo de settling para a nova conversão seja assegurado enquanto decorre ainda a conversão da selecção anterior. Este tipo de operação não é obrigatória mas incrementará a capacidade do modo de aquisição regular.

O modo de auto-aquisição permite uma velocidade de aquisição de sinais analógicos de 50 kHz. Uma vez escolhido este modo de aquisição para o módulo AMM2, o conversor A/D executará uma aquisição contínua sem ser necessária a implementação do comando A/D START.

Depois de completa uma conversão, o módulo AMM2 inicia de imediato nova conversão. O comando EOC STATUS pode ser utilizado para sincronizar o programa de software com os processos de conversão. As conversões serão executadas sobre a slot e canal presentemente especificados a uma velocidade de 50 kHz.

4.2.5. SELECT FILTER

Localização: xxx80

No módulo AMM2 estão disponíveis dois filtros: um filtro a 100 kHz e outro a 2 kHz. Estes filtros restringem a largura da banda do sinal de entrada, rejeitando ruído e componentes de alta frequência indesejados.

Atribuindo o valor 0 ao bit da posição D7 do CMDA, seleccionará o filtro a 100 kHz; o valor 1 seleccionará o filtro a 2 kHz.

4.2.6. SELECT SLOT

Localização: xxx81

O comando SELECT SLOT controla o multiplexer global do módulo AMM2, seleccionando o slot no qual se vai implementar a leitura. O valor a ser escrito na localização do SELECT SLOT ocupa os quatro bits menos significativos do CMDB (D0, D1, D2 e D3). Certifique-se que o número do slot escolhido combina com os quatro bits mais significativos do CMDB antes de executar o comando.

4.2.7. SELECT CMDA READ MODE

Localização: xxx81

Este comando selecciona o modo de leitura do CMDA. Dois tipos de informação podem ser lidos: o byte menos significativo do conversor A/D (A/D LOW DATA) ou o estado do conversor A/D (A/D STATUS) (ver 4.2.11.).

Quando o modo de leitura do CMDA está seleccionado para A/D STATUS, uma sequência de RESET AND RECAL (ver 4.2.10) é inicializada sempre que é implementado um comando de A/D START (ver 4.2.13.). Notar que um comando de início de conversão pode ser implementado através de uma escrita no CMDD (A/D START) ou via hardware quando em modo de auto-aquisição (se este modo foi seleccionado atribuindo o valor 1 ao bit da posição D6 do CMDA).

O modo de leitura A/D STATUS só tem significado quando em modo de aquisição regular (ver 4.2.4). Para evitar um início indesejado de uma sequência RESET AND RECAL, certifique-se que o bit da posição D6 do CMDA está seleccionado para o valor 0 (modo de aquisição regular) antes de alterar o modo de leitura do CMDA para A/D STATUS. Não escreva para o CMDD (A/D START) nem mude o valor do bit da posição D6 do CMDA para 1, enquanto o modo de leitura do CMDA estiver configurado para A/D STATUS.

Atribua o valor 0 ao bit da posição D4 do CMDB para o modo A/D STATUS do CMDA ou o valor 1 para o modo A/D LOW DATA.

4.2.8. SELECT RANGE

Localização: xxx81

O módulo AMM2 possui duas gamas possíveis: $\pm 10\text{V}$ (bipolar 10V) e $0 \sim +10\text{V}$ (unipolar 10V). Atribuindo o valor 0 ao bit da posição D5 do CMDB, será seleccionada a gama unipolar 10V; seleccionando o valor 1, será seleccionada a gama bipolar 10V.

4.2.9. SELECT GLOBAL GAIN

Localização: xxx81

O comando GLOBAL GAIN controla o ganho do amplificador global do módulo AMM2. Desde que as entradas analógicas sejam processadas pelo amplificador global, este comando afecta todas as entradas analógicas ligadas à Series 500.

O valor especificador do GLOBAL GAIN ocupa os dois bits mais significativos do CMDB (D6 e D7 da figura 4.2.) e deverá ser combinado com os outros bits do byte

CMDB antes de ser implementado o comando. Quatro ganhos estão disponíveis no amplificador global programável, como mostra a tabela 4.2.

Tabela 4.2. Valores especificadores do ganho do amplificador global

Ganho	Byte CMDB
x1	00bbbbbb
x2	01bbbbbb
x5	10bbbbbb
x10	11bbbbbb

4.2.10. RESET AND RECAL

Localização: xxx9A

O comando RESET AND RECAL inicializa o processo de calibração A/D interna. Este processo demora aproximadamente 360 mseg e deverá ser implementado cada vez que o sistema é ligado.

Após ser implementado este comando, deve-se esperar pelo menos 360 mseg antes de se inicializar o processo de conversão. Para se certificar que a calibração foi efectuada, coloque o bit do READ MODE do CMDA em A/D STATUS, como foi descrito anteriormente (ver 4.2.7.). O bit de configuração do estado de calibração é descrito de seguida (ver 4.2.11.).

Este comando não tem associado um dado específico; qualquer valor enviado inicializará o processo de calibração.

4.2.11. A/D LOW DATA - A/D STATUS

Localização: xxx80

O conteúdo do CMDA (leitura) depende do estado do módulo AMM2 seleccionado pelo comando SELECT CMDA READ MODE (ver 4.2.7.). Se o bit da posição D4 do CMDB valer 0, o CMDA retorna o estado da conversão A/D (A/D STATUS); se valer 1, o byte menos significativo da conversão A/D é fornecido no CMDA (A/D LOW DATA).

Quando o módulo AMM2 está em modo de A/D STATUS, a configuração do byte CMDA é a seguinte:

- D0 sem significado
- D1 sem significado
- D2 sem significado

- D3 sem significado
- D4 sem significado
- D5 TRACKING (1=tracking em processo; 0=sem tracking em processo)
- D6 CONVERTING (1=conversão em processo; 0=sem conversão em processo)
- D7 CALIBRATING (1=calibração em processo; 0=sem calibração em processo)

Depois do conversor A/D completar a digitalização de um sinal analógico, inicia-se o processo de tracking. O conversor gasta 4 μ seg neste processo.

A relação temporal entre os bits de TRACKING, CONVERTING e EOC (ver 4.2.14.) está representada na figura 4.4.

O bit de calibração retorna o estado do comando RESET AND RECAL, tal como foi descrito anteriormente (ver 4.2.10).

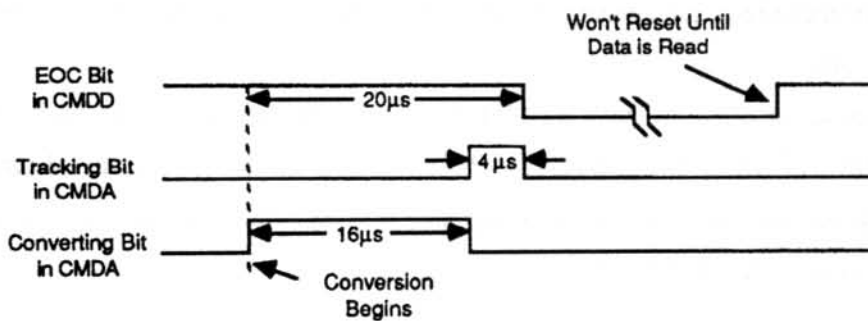


Figura 4.4. Relação temporal entre os bits de estado (em A/D STATUS).

Se o módulo AMM2 está em modo de A/D LOW DATA, o byte recebido no CMDA é o byte menos significativo dos 16 bits fornecidos pelo conversor A/D.

Notar que é possível ler um valor de conversão enquanto outra conversão está já em processo. Para verificar quando um valor digital está disponível, podemos usar os comandos A/D START e EOC STATUS descritos de seguida (ver 4.2.13. e 4.2.14.).

4.2.12. A/D HIGH DATA

Localização: xxx81

O comando A/D HIGH DATA executa uma função análoga à do comando A/D LOW DATA, excepto que, deste modo, é lido o byte mais significativo. Desde que o módulo

AMM2 tenha um conversor de 16 bits, todos os bits do HIGH DATA byte (conteúdo do CMDB em modo leitura) são significativos.

Uma vez obtido o LOW e o HIGH DATA bytes, o número total de counts que representa a conversão A/D pode ser determinado através da expressão:

$$CO = DL + 256 * DH$$

onde CO representa o número de counts e DL e DH representam o LOW e HIGH DATA bytes, respectivamente. Desde que o módulo AMM2 utilize um conversor de 16 bits, o número de counts terá um valor dentro da gama 0~65535 (2^{16} estados possíveis).

4.2.13. A/D START

Localização: xxx9B

O comando A/D START inicia o processo de conversão A/D. Apesar de qualquer valor (0~255) possa ser escrito para efectuar a conversão, deve ser utilizado o valor 255 para minimizar o ruído.

Não implemente o comando A/D START em modo de auto-aquisição ou enquanto decorre ainda um ciclo de conversão. Note que o ciclo de conversão demora aproximadamente 16 μ seg e que o estado do processo de conversão pode ser verificado utilizando o comando EOC STATUS.

4.2.14. EOC STATUS

Localização: xxx9B

O comando EOC STATUS fornece um byte de informação que indica o estado do processo de conversão. Se o valor do byte de informação for 1xxxxxxx, está a ser processado um ciclo de conversão; se for 0xxxxxxx, já foi atingido o fim de conversão.

5. MÓDULO AIM7

O módulo AIM7 suporta a entrada de sinais de termopares do tipo B, E, J, K, R, S e T dentro de uma gama de ± 100 mV. Este módulo pode ser instalado nos slots 3~10. No entanto, para minimizar o ruído, é recomendável colocá-lo no slot de número mais baixo disponível (slot 3, se possível).

5.1. COMANDOS

O comando utilizado no módulo AIM7 é o comando SELECT CHANNEL que selecciona o canal onde se vai processar a leitura (de entre 16 canais de entrada analógica disponíveis).

A localização do comando depende do slot onde está instalado o módulo, como mostra a tabela 5.1.

Tabela 5.1. Localização do comando SELECT CHANNEL no módulo AIM7

Slot	SELECT CHANNEL (CMDA)
Slot 3	CFF84
Slot 4	CFF86
Slot 5	CFF88
Slot 6	CFF8A
Slot 7	CFF8C
Slot 8	CFF8E
Slot 9	CFF90
Slot 10	CFF92

Em todos os casos, o número do canal a ser lido deve ser escrito na localização do comando SELECT CHANNEL, como mostra a tabela 5.2. Por exemplo, se queremos seleccionar o canal 0, o valor 0 deve ser escrito na localização SELECT CHANNEL.

Tabela 5.2. Valores a serem escritos na localização SELECT CHANNEL.

Canal	Binário	Hex	Decimal
Canal 0	00000	H0	0
Canal 1	00001	H1	1
Canal 2	00010	H2	2
Canal 3	00011	H3	3
Canal 4	00100	H4	4
Canal 5	00101	H5	5
Canal 6	00110	H6	6
Canal 7	00111	H7	7
Canal 8	01000	H8	8
Canal 9	01001	H9	9
Canal 10	01010	HA	10
Canal 11	01011	HB	11
Canal 12	01100	HC	12
Canal 13	01101	HD	13
Canal 14	01110	HE	14
Canal 15	01111	HF	15

Um comando SELECT CHANNEL deve ser seguido de um comando do AMM2 SELECT SLOT que indica o slot onde está instalado o módulo AIM7 (ver comandos do módulo AMM2).





FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

BIBLIOTECA



0000034589