



**Universidade de
Aveiro
2007**

Departamento de Electrónica,
Telecomunicações e Informática

**Ana Margarida Barreto Sistema de Monitorização Remota para Elevadores
de Miranda Sargento**



**Universidade de
Aveiro
2007**

Departamento de Electrónica,
Telecomunicações e Informática

Ana Margarida Barreto de Miranda Sargento **Sistema de Monitorização Remota para Elevadores**

Dissertação apresentada à Universidade de Aveiro para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Engenharia Electrónica e Telecomunicações, realizada sob a orientação científica do Prof. Dr. Manuel Bernardo Salvador Cunha, Professor auxiliar do Departamento de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro.

o júri

presidente

Prof. Doutor Valeri Skliarov
Professor Catedrático da Universidade de Aveiro

Prof. Doutor Jorge Augusto Fernandes Ferreira
Professor Auxiliar da Universidade de Aveiro

Prof. Doutor Manuel Bernardo Salvador Cunha
Professor Auxiliar da Universidade de Aveiro

agradecimentos

Ao professor Bernardo Cunha, pela oportunidade que me deu de realizar este trabalho, aceitando ser meu orientador, e dando-me todo o apoio durante a sua realização.

À LifTech e aos Engenheiros António Garrido e Miguel Tato, pela oportunidade e pelo apoio na realização do trabalho que serviu de base a esta dissertação.

Ao Alexandre Santos, colega de desenvolvimento do trabalho que serviu de base a esta dissertação, pelas muitas discussões que se revelaram produtivas, pelo trabalho incansável e pela transmissão de conhecimentos essenciais ao desenvolvimento do trabalho apresentado.

A todos os meus amigos e colegas de trabalho, pela paciência que manifestaram quando não lhes pude dar o apoio esperado.

Aos meus pais, irmã, cunhado e restantes familiares, pela enorme motivação, apoio e carinho em cada novo desafio.

Ao Luís, pela sua ajuda a todos os níveis, pela motivação, paciência, e amor sempre presentes.

palavras-chave

Monitorização remota; elevadores; sistema adaptável e modular; registo de assistências técnicas; geração de alarme; histórico de eventos; protocolos de comunicação; resistência a falhas

resumo

O trabalho que deu origem a esta dissertação foi desenvolvido no âmbito da disciplina de projecto do 5º ano do curso de Engenharia Electrónica e de Telecomunicações da Universidade de Aveiro, no ano lectivo 2003/2004. Este trabalho enquadrava-se num conjunto de três trabalhos baseados em propostas de desenvolvimento provenientes da *LifTech*, uma empresa exterior à Universidade. Esta empresa assumiu há alguns anos um sector de actividade da EFACEC, desenvolvendo, fabricando e comercializando sistemas de comando para elevadores. Os resultados destes três trabalhos tinham o objectivo de vir a integrar as soluções tecnológicas oferecidas pela empresa.

Foi desenvolvido um sistema capaz de fazer a supervisão remota de elevadores ou conjuntos de elevadores. O sistema é adaptável a qualquer tipo de elevador, tem capacidade para monitorizar um conjunto de sinais e para disponibilizar 1 ou 2 comandos controláveis externamente. Suporta ainda um *interface* para fazer o registo de assistências técnicas.

A monitorização propriamente dita é efectuada a partir de um Centro de Supervisão que poderá estar em contacto com vários conjuntos de elevadores. O Centro de Supervisão é constituído por um PC com o *software* adequado, o qual foi também parcialmente desenvolvido no âmbito deste trabalho.

A troca de informação entre o sistema de monitorização e o referido PC pode ser desencadeada a partir do Centro de Supervisão ou, de um modo automático, em resposta a eventos específicos nos elevadores. A comunicação pode ser efectuada recorrendo a uma linha telefónica fixa ou a SMS sobre GSM.

O sistema permite a ligação local a um PC para descarregar um histórico de eventos e registos de assistência. A configuração e parametrização dos módulos de supervisão locais pode ser efectuada também a partir do PC em modo local.

keywords

Remote monitoring; elevators, adaptable and modular system; technical assistance registration; alarm; events report; communication protocols; fail resistance

abstract

The work that was the basis for this thesis was developed in the context of the final year project on the Engineering Electronics and Telecommunications course, on 2003/2004.

This work was part of a group of three projects based on developing proposals from LifTech, a company external to the university. For some years now, LifTech assumed a former business unit from EFACEC, developing, constructing and commercializing Elevator Control Systems. Results of these three projects aimed to integrate the technological products portfolio offered by LifTech.

A system capable of remote monitoring of an elevator or group of elevators was developed. The system is adaptable to any kind of elevator, can monitor a group of signals, and can provide 1 or 2 commands externally controlled. It also supports an interface for technical assistance registration.

Monitoring is performed through a Supervisor Centre, which can be connected to different sets of independent elevator groups. The Supervisor Centre consists of a PC equipped with the appropriate software, which was also partially developed within this project.

Information exchange between monitoring system and Supervisor PC can be started by the Supervisor Centre itself, or automatically triggered by events generated in the elevators. Communication can be performed by circuit switching line or SMS over GSM.

The system allows connection to a local PC, for events and assistance registration logs download. System configuration and parameterization can also be performed by the local PC.

Índice Geral

Índice Geral	1
Índice de Figuras	5
1 Introdução	7
1.1 Enquadramento e motivações.....	7
1.2 Objectivos e estrutura da dissertação.....	8
2 Estado-da-arte	11
3 Especificação Funcional do Sistema	25
3.1 Introdução.....	25
3.2 Descrição global do sistema.....	25
3.3 Unidade Local de Monitorização.....	26
3.3.1 Atravancamento do sistema.....	27
3.3.2 Barramento de interligação entre módulos.....	28
3.3.3 Módulo Principal.....	28
3.3.4 Módulo de Comunicações.....	30
3.3.5 Módulo I/O.....	31
3.3.6 Módulo de Alimentação.....	31
3.4 Centro de Supervisão.....	31
3.4.1 Configuração remota da Unidade Local de Monitorização.....	32
4 Unidade Local de Monitorização – Hardware	35
4.1 Introdução.....	35
4.2 Breve Estudo Funcional e Económico de Componentes da ULM.....	35
4.2.1 Dispositivos de Identificação.....	35
4.2.2 Microcontrolador.....	37
4.2.3 Barramento de Comunicação.....	38
4.2.4 Memória não volátil.....	42
4.2.5 Interfaces das Entradas e Saídas.....	43

4.2.6	Módulo USB.....	44
4.3	Set de Módulos da ULM.....	44
4.3.1	Módulo Principal	44
4.3.2	Módulo de Comunicações	45
4.3.3	Módulo I/O	46
4.3.4	Módulo de Alimentação	46
4.3.5	Set de Módulos Mínimo	48
4.4	Protótipo da Unidade Local de Monitorização.....	48
5	Especificação do Protocolo de Comunicação entre os Módulos	49
5.1	Introdução.....	49
5.2	Cenários de Utilização.....	49
5.2.1	<i>Power up da Unidade de Monitoração Local.....</i>	49
5.2.2	<i>Configuração dos módulos.....</i>	50
5.2.3	<i>Comunicação de ocorrências.....</i>	50
5.2.4	<i>Comunicação exterior (via Módulo de Comunicações).....</i>	51
5.2.5	<i>Modelo de Comunicação.....</i>	51
5.3	Relações de Precedência e Limitações da Comunicação.....	52
6	Unidade Local de Monitorização – Software.....	53
6.1	Introdução.....	53
6.2	Módulo Principal	53
6.2.1	Algoritmo geral.....	53
6.2.2	Detalhes de implementação	65
6.3	Módulo de Comunicações.....	66
6.3.1	Algoritmo geral.....	66
6.3.2	Detalhes de implementação	68
6.4	Módulo I/O.....	68
6.4.1	Algoritmo geral.....	68
6.4.2	Algoritmo de amostragem dos portos de entrada	69
7	Protocolo de Comunicação entre PC e ULM.....	73
7.1	Introdução.....	73
7.2	Fluxos de Informação	73
7.2.1	Recepção de alarme.....	74
7.2.2	Estabelecimento de Comunicação com uma nova ULM.....	75
7.2.3	Estabelecimento de Comunicação com uma ULM Existente.....	77
7.2.4	Acerto da Hora da ULM	79
7.2.5	Actuar numa saída da ULM.....	80
7.2.6	Análise do Estado da Memória de Histórico da ULM.....	81
7.2.7	Download do Histórico	82
7.2.8	Envio da Configuração para a ULM	83
7.3	Relações de Precedência e Limitações da Comunicação.....	86

7.4	Comunicação por GSM ou Rede Fixa Comutada	87
8	Software do PC.....	89
8.1	Introdução.....	89
8.2	Estrutura da Base de Dados	89
8.3	PC do Centro de Supervisão	92
8.3.1	Cenários de Utilização.....	92
8.3.2	Estrutura do <i>Software</i>	92
8.4	PC de Assistência.....	95
8.4.1	Cenários de Utilização.....	95
8.4.2	Estrutura do <i>Software</i>	96
8.5	<i>Software</i> de Demonstração.....	96
8.5.1	Introdução.....	96
8.5.2	Cenários de Utilização.....	97
8.5.3	Estrutura do <i>Software</i>	97
9	Conclusões.....	105
9.1	Especificação do sistema	105
9.2	Hardware Escolhido	106
9.3	Software da ULM.....	106
9.4	Comunicação GSM vs. PSTN vs. GPRS	106
9.5	Software do PC.....	107
9.6	Actualidade do Trabalho Desenvolvido	107
	Anexos	109
A	Características e Preços de Memórias.....	111
A.1	EEPROMs.....	111
A.1.1	Catalyst.....	111
A.1.2	Farnell.....	111
A.1.3	Mouser Electronics	111
A.1.4	RS.....	112
A.2	Memórias Flash	112
A.2.1	ChipCatalog	112
A.2.2	Farnell.....	113
A.2.3	SST	113
B	Esquemáticos Completos	115
C	Listagem dos comandos de comunicação entre os módulos da ULM	119

D Listagem dos comandos de comunicação entre PC e ULM 121

Publicações do autor no âmbito do trabalho exposto na Dissertação..... 123

Lista de Acrónimos 125

Referências 127

Índice de Figuras

Figura 1 – Unidade Central do Sistema ADAM.....	12
Figura 2 – Sistema <i>REMAS</i>	14
Figura 3 – Sistema da Mitsubishi	15
Figura 4 – Configuração do sistema de monitorização de elevadores e escadas <i>MelEye</i>	17
Figura 5 – Configuração do Software do <i>MelEye</i>	18
Figura 6 – Sistema <i>QarVision</i> da <i>Qameleon</i>	20
Figura 7 – Diagrama de blocos do sistema.....	25
Figura 8 – Diagrama de blocos da Unidade Local de Monitorização	27
Figura 9 – <i>Memory stack</i> da <i>EEPROM</i>	29
Figura 10 – Módulo de Comunicações	30
Figura 11 – Esquemático dos circuitos de interface das entradas e saídas, respectivamente	43
Figura 12 – Esquema do Módulo de Comunicações	44
Figura 13 – Esquema do Módulo de Comunicações	45
Figura 14 – Esquema do Módulo I/O	46
Figura 15 – Esquema do Módulo de Alimentação	47
Figura 16 – Modelo de comunicação para um pedido de informação.....	52
Figura 17 – Modelo de comunicação para a transmissão de um comando de actuação.....	52
Figura 18 – Diagrama de estados do software do módulo principal.....	54
Figura 19 – Fluxograma do software de arranque do módulo principal.....	54
Figura 20 – Fluxograma do estado VALIDAR <i>EEPROM</i>	55
Figura 21 – Fluxograma do estado VALIDAR DISPOSIÇÃO FÍSICA.....	56
Figura 22 – Fluxograma do estado VERIFICAR OCORRÊNCIAS	57
Figura 23 – Representação da tabela de ocorrências	58
Figura 24 – Descrição da configuração de um evento	59
Figura 25 – Representação do <i>buffer</i> circular e de uma entrada da lista de espera.....	60
Figura 26 – Fluxograma do estado TRATAR EVENTOS	61
Figura 27 – Estrutura das entradas do histórico de um evento normal e das duas excepções	62
Figura 28 – Fluxograma do estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES EM TEMPO- REAL.....	63
Figura 29 – Fluxograma do estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES	64
Figura 30 – Fluxograma da rotina de interrupção associada à recepção de tramas <i>CAN</i>	66
Figura 31 – Fluxograma do algoritmo de controlo do módulo de comunicações	67
Figura 32 – Fluxograma do Módulo I/O	69
Figura 33 – Fluxograma do algoritmo de processamento dos sinais de entrada	70

Figura 34 – Exemplo de um sinal à entrada de um dos portos.....	70
Figura 35 – Sinal resultante da amostragem do sinal apresentado na Figura 34	71
Figura 36 – detectado com “estado inicial=0”	71
Figura 37 – Estado detectado com “estado inicial=1”	71
Figura 38 – Sequência de mensagens trocadas aquando da recepção de um alarme	75
Figura 39 – Sequência de mensagens trocadas aquando da ligação a uma nova ULM	76
Figura 40 – Sequência de mensagens trocadas quando é efectuado um pedido de informação... 76	
Figura 41 – Sequência de mensagens trocadas aquando da ligação a uma ULM existente.....	78
Figura 42 – Sequência de mensagens trocadas aquando do acerto da hora da ULM	80
Figura 43 – Sequência de mensagens trocadas aquando da actuação numa saída.....	81
Figura 44 – Sequência de mensagens trocadas aquando do pedido de estado do Histórico na ULM	81
Figura 45 – Sequência de mensagens trocadas aquando do download do Histórico da ULM	83
Figura 46 – Sequência de mensagens trocadas aquando da configuração dos módulos da ULM. 84	
Figura 47 – Sequência de mensagens trocadas aquando da configuração dos eventos da ULM.. 85	
Figura 48 – Diagrama de Entidades Relacionais.....	90
Figura 49 – Estrutura do <i>Software</i> do Centro de Supervisão	93
Figura 50 – Janela Principal da Aplicação de Demonstração.....	98
Figura 51 – Janela de configuração dos modems	98
Figura 52 – Janela de Escolha da Porta de Comunicação	99
Figura 53 – Janela da ULM	99
Figura 54 – Janela de Visualização do Histórico	100
Figura 55 – Janela de Configuração dos Parâmetros Gerais da ULM	101
Figura 56 – Janela de Configuração de um Módulo I/O	101
Figura 57 – Janela de Configuração de um Evento.....	102
Figura 58 – Esquemático do protótipo do Módulo Principal.....	115
Figura 59 – Esquemático do protótipo do Módulo de Comunicações	116
Figura 60 – Esquemático do protótipo do Módulo de I/O (parte 1)	117
Figura 61 – Esquemático do protótipo do Módulo de I/O (parte 2)	118

1 Introdução

1.1 Enquadramento e motivações

Actualmente os edifícios são projectados para ser cada vez mais altos para responder a diferentes funcionalidades, e atingem muitas vezes complexidades de funcionamento que os denominam de “edifícios inteligentes”. Os elevadores assumem um papel extremamente importante como meios de transporte no interior dos edifícios. Quando um elevador não funciona correctamente não se trata apenas de uma inconveniência, pode impedir pessoas de fazer o seu trabalho, interferir com a sua vida e eventualmente resultar em prejuízos para o dono do edifício. A existência de sistemas de monitorização multifuncionais e de elevada performance torna-se de elevada importância para o controlo destes equipamentos.

O trabalho descrito nesta dissertação foi desenvolvido no âmbito da disciplina de projecto do 5º ano do curso de Engenharia Electrónica e de Telecomunicações da Universidade de Aveiro, no ano lectivo 2003/2004. Este trabalho enquadrava-se num conjunto de três projectos baseados em propostas de desenvolvimento provenientes de uma empresa exterior à Universidade de nome LifTech. Esta empresa assumiu há alguns anos um sector de actividade da EFACEC, desenvolvendo, fabricando e comercializando Sistemas de Comando para Elevadores. Os resultados destes três projectos poderiam vir a integrar as soluções tecnológicas oferecidas pela empresa.

A proposta apresentada visava o desenvolvimento de um sistema capaz de fazer a supervisão remota de elevadores ou conjuntos de elevadores. Pretendia-se que o sistema fosse adaptável a qualquer tipo de elevador e tivesse capacidade quer para monitorizar um conjunto de sinais quer para disponibilizar 1 ou 2 comandos actuáveis externamente. O sistema deveria suportar uma interface para um leitor de cartões magnéticos com o objectivo de fazer o registo de assistências técnicas.

A monitorização propriamente dita seria efectuada a partir de um Centro de Supervisão que estaria em contacto com vários conjuntos de elevadores. O Centro de Supervisão seria constituído por um PC com o software adequado, o qual deveria ser também desenvolvido no âmbito deste projecto. A troca de informação com o referido PC deveria poder ser desencadeada a partir do Centro de Supervisão onde se encontrava ou de um modo automático em resposta a eventos específicos nos elevadores. A comunicação deveria ser efectuada recorrendo a uma linha telefónica fixa ou a SMS sobre GSM.

Como opção, deveria ainda ser considerada a possibilidade de estabelecer uma ligação local a um *Palmtop Computer* para descarregar um histórico de ocorrências e registos de assistência. A configuração e parametrização do(s) módulo(s) de supervisão local(ais) deveria poder ser efectuada a partir do próprio Centro de Supervisão ou através do referido *Palmtop Computer* em modo local.

O Plano de trabalho proposto dividia-se nas seguintes etapas:

- Desenvolvimento do módulo base de monitorização que deveria suportar 6 entradas (isoladas opticamente) e 2 saídas (por relé). Este módulo deveria ainda suportar a integração de módulos de expansão (número a definir) com idêntico número de entradas e saídas.
- Desenvolvimento do software de monitorização para o PC remoto.
- Desenvolvimento do software para descarga de histórico e parametrização do sistema para *Palmtop Computer*.
- Estudo e produção de um relatório com uma avaliação comparativa dos custos de comunicação sobre linha fixa ou recorrendo a GSM.

1.2 Objectivos e estrutura da dissertação

Esta dissertação tem como principal objectivo descrever o trabalho realizado no âmbito da disciplina de projecto do 5º ano no ano lectivo 2003/2004 analisado à luz do contexto actual. Nesta perspectiva, o *Estado-da-arte*, analisado na altura da realização do trabalho, foi revisto tendo em conta os anos decorridos deste então e a evolução que os sistemas equivalentes sofreram até à data. Esta dissertação pretende também avaliar a actualidade do trabalho previamente desenvolvido bem como a viabilidade da aplicação prática do mesmo, analisando a sua integração nas soluções tecnológicas da LifTech.

O trabalho foi desenvolvido por um grupo formado por duas pessoas. Esta dissertação apresenta especial ênfase na especificação do sistema, definição dos protocolos de comunicação e desenvolvimento de *software* de demonstração para o PC de assistência, uma vez que o desenvolvimento do *software* de monitorização integrado na ULM esteve maioritariamente sob a responsabilidade do colega do grupo de trabalho.

A estrutura da dissertação permite percorrer as várias fases de desenvolvimento do trabalho, analisando o sistema numa perspectiva *top-down* que progressivamente vai detalhando cada um dos elementos que o constitui.

O capítulo 2, *Estado-da-arte*, apresenta um estudo relativo aos sistemas equivalentes existentes no mercado à data da realização do trabalho bem como actualmente.

O capítulo 3 apresenta a especificação funcional do sistema desenvolvido, a arquitectura dos módulos que o compõem e uma breve descrição do *software* de suporte aos módulos.

No capítulo 4 são apresentados os resultados de um breve estudo efectuado tendo em vista a adequação funcional e económica das tecnologias a integrar neste sistema, bem como as soluções encontradas e a implementação das mesmas.

O capítulo 5 apresenta uma descrição do protocolo de comunicação entre os módulos da ULM (Unidade Local de Monitorização).

No capítulo 6 é feita uma descrição do software integrado na ULM (Unidade Local de Monitorização).

No capítulo 7 é descrito o protocolo de comunicação entre PC e ULM.

No capítulo 8 é apresentada uma descrição do software do PC de supervisão e as suas funcionalidades.

Finalmente, no capítulo 9, são apresentadas as conclusões do trabalho.

O anexo A contém uma listagem das características e preços das memórias analisadas (durante o desenvolvimento do trabalho, ano de 2003/2004)

O anexo B apresenta os esquemáticos dos circuitos desenvolvidos para os vários módulos da ULM.

O anexo C contém a listagem dos comandos de comunicação entre os módulos da ULM, enquanto que o anexo D contém a listagem dos comandos de comunicação entre o PC e a ULM.

2 Estado-da-arte

O desenvolvimento do Sistema de Monitorização de Elevadores foi precedido de uma pesquisa de sistemas semelhantes existentes no mercado. Esta pesquisa teve como objectivo avaliar e fundamentar as escolhas efectuadas no decorrer do trabalho e também permitir o desenvolvimento de uma solução tão completa e actual quanto possível.

À data da escrita desta dissertação foi efectuada uma nova pesquisa de sistemas de monitorização de elevadores com vista a avaliar a actualidade do trabalho desenvolvido e a evolução que os sistemas semelhantes sofreram.

Alguns dos sistemas encontrados são direccionados para os elevadores das próprias empresas que os desenvolvem, estando dotados de funcionalidades bastante específicas. Por outro lado, outros foram desenvolvidos com o intuito de serem integrados na generalidade dos elevadores já existentes e em funcionamento. O sistema desenvolvido e descrito nesta dissertação insere-se nesta última categoria.

O sistema desenvolvido pela OTIS denomina-se “*Remote Elevator Monitoring*” [1] e destina-se apenas à instalação em equipamentos da mesma empresa. Cada unidade local ao detectar um problema, faz um diagnóstico da causa e da localização do mesmo, guarda a informação e efectua uma chamada para a central de monitorização. O sistema permite ainda detectar a deterioração de componentes e prever algumas outras anomalias. O conjunto de ocorrências relacionadas com cada elevador permite a elaboração de um relatório que pode ser apresentado aos clientes. Uma vez que este sistema foi desenvolvido para utilização exclusiva desta empresa, não é disponibilizada de forma pública qualquer informação quanto à tecnologia utilizada.

Foram encontradas referências para este sistema tanto em 2003 como em 2006. As funcionalidades apresentadas estão presentes desde 2003. Actualmente, o sistema da OTIS atingiu já a versão 5.0. Possivelmente as evoluções que sofreu melhoraram as funcionalidades já disponíveis no sistema apresentado anteriormente mas não são apresentadas novidades significativas além da disponibilização de um serviço *web* que permite que o operador de manutenção tenha acesso a uma análise da performance e ao histórico de registos de cada elevador.

Este sistema é usado na monitorização de cerca de 10000 elevadores nos EUA.

A *Advantech* desenvolveu um sistema para a *China-Ryoden*, designado “*A Centralized Elevator Monitoring System*” [2] [3]. Dado que o sistema se destina a uma gama de elevadores muito vasta e que o fabricante pretende que seja aplicado a um grande número de instalações, o produto desenvolvido assenta numa arquitectura modular, facilitando a manutenção e a substituição de módulos danificados. Esta arquitectura permite ainda uma fácil expansão.



Figura 1 – Unidade Central do Sistema ADAM

Este sistema implementa uma monitorização equivalente ao sistema da OTIS, permitindo a sua programação remota e uma transmissão automática de situações de alarme para um centro de controlo. É constituído por três partes fundamentais: uma unidade central, um modem que suporta as comunicações e uma fonte de tensão comutada.

A unidade central possui um controlador programável baseado num PC, ADAM-5510, com dois módulos com 8 entradas digitais cada, ADAM-5052, e dois outros módulos com 8 saídas digitais por relé, ADAM-5068. Cada saída digital opera a 125V e 0.5A, AC, 30V e 1A, DC, e 110V e 0.5A com saída por relé. Estas saídas podem ser utilizadas para actuar na ligação de um altifalante, fazer o *reset* do modem ou alterar as sub-unidades em funcionamento, permitindo ligar ou desligar diferentes sub-módulos.

As funcionalidades apresentadas datam de 2003, no entanto foram encontradas referências para este sistema tanto em 2003 como em 2006. As referências actuais não apresentam novas funcionalidades, pelo que a possível evolução do sistema deve ter assentado numa melhoria das funcionalidades já disponibilizadas.

A *Elevator World* desenvolveu um sistema, “*Remote Elevator Monitoring and Alarming System*” (*REMAS*) [4], que tanto pode ser aplicado a conjuntos de elevadores de instalações futuras como aos conjuntos de elevadores já existentes e em funcionamento, incluindo os que estão baseadas em controladores lógicos com relés. Para além de fazer a monitorização do estado dos elevadores, é suportada uma comunicação por voz entre o centro de monitorização e os passageiros presos num elevador bloqueado, sem necessidade de modificar o sistema de comunicação.

De acordo com a publicação que descreve o sistema *REMAS*, um sistema de monitorização de elevadores deve cumprir as seguintes especificações:

- indicação de estado de elevador em funcionamento;
- sinalização de alarme de passageiro preso em elevador bloqueado;
- sinalização de avaria no sistema de alarmes do elevador;
- imediata transmissão de alarmes e estado do elevador ao centro de monitorização;
- recolha automática dos dados relativos ao funcionamento do elevador;
- comunicação por voz, bidireccional, com os passageiros presos num elevador bloqueado;
- configuração remota dos módulos do sistema (entradas, saídas, etc);
- possibilidade de supervisionar a actividade do elevador “*on line*”;
- avaliação opcional da performance dos elevadores e
- análise de dados.

Neste documento, é referida a importância de avaliar os eventos que devem reportar alarme e os que devem constar apenas do registo de ocorrências, assim como o conjunto de eventos que se podem associar para reportar um único alarme, ou ainda, os eventos que só fazem sentido num determinado tipo de instalações, novas ou antigas. Esta avaliação revela-se de extrema importância do ponto de vista do centro de monitorização, para que o congestionamento da linha de comunicação seja mínimo.

O sistema *REMAS* detecta os seguintes acontecimentos anormais:

- porta aberta durante mais de 50 segundos;
- falha de tensão de alimentação no controlador do elevador;
- falha no relé de segurança;
- pressão do botão de emergência durante mais de 3 segundos;
- falha da tensão de alimentação do sistema, 220 V (AC).

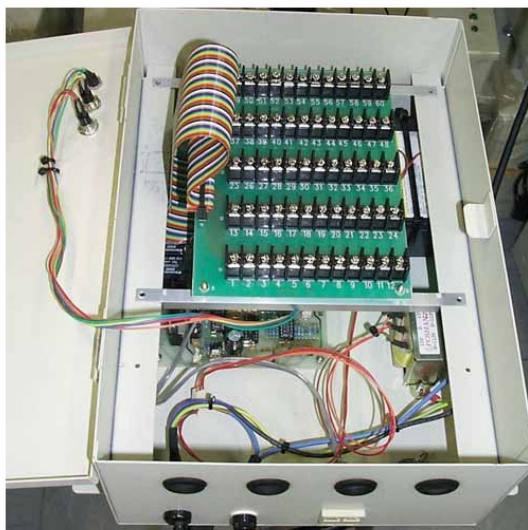


Figura 2 – Sistema *REMAS*

O sistema *REMAS* possui uma bateria de 12 V que assegura o seu funcionamento durante mais de duas horas no caso de falha de energia. Durante a operação normal dos elevadores esta bateria é carregada a partir dos 220 V da alimentação da rede.

O *REMAS* possui quatro fios de *interface* com os controladores de quatro elevadores, uma ligação à alimentação de 220V e uma ligação telefónica.

O sistema *REMAS* possui ainda um *LED* que, conforme está permanentemente aceso, apagado ou ligado intermitentemente com frequências predefinidas, indica o estado do mesmo à equipa de manutenção, em tempo real.

Durante a manutenção o *REMAS* não transmite nenhum alarme vindo dos controladores dos elevadores, assegurando que o centro de emergência não recebe nenhum falso alarme relacionado com a manutenção de rotina.

Neste sistema é utilizado um *watchdog* para fazer o *reset* ao microcontrolador no caso de ocorrer alguma falha. O *watchdog* é ainda utilizado como *timer* para agendar chamadas telefónicas sucessivas, no caso da linha estar ocupada.

Cada alarme é acompanhado de um conjunto de informações relevantes:

- número do elevador;
- número do contrato;
- morada do elevador de onde provém o alarme;
- descrição da falha;
- número de telefone da empresa de manutenção da área;

- número de telefone do porteiro do prédio;
- hora da chamada;
- hora do reconhecimento do alarme por parte do centro de monitorização.

Toda a informação é guardada em disco para posterior análise, em formato *EXCEL*.

As funcionalidades descritas faziam parte do sistema de monitorização existente em 2003. Actualmente não foi encontrada qualquer referência *web* a este sistema, de qualquer modo, a análise da informação disponível à data do desenvolvimento do trabalho, contribuiu significativamente para a procura de uma solução mais completa, chamando a atenção para especificidades para as quais se não estaria naturalmente desperto.

O sistema da *Mitsubishi*, “*A Remote Inspection System for Elevators*” encontrado em 2003 [5], tem como princípio de funcionamento o esquema da Figura 3.

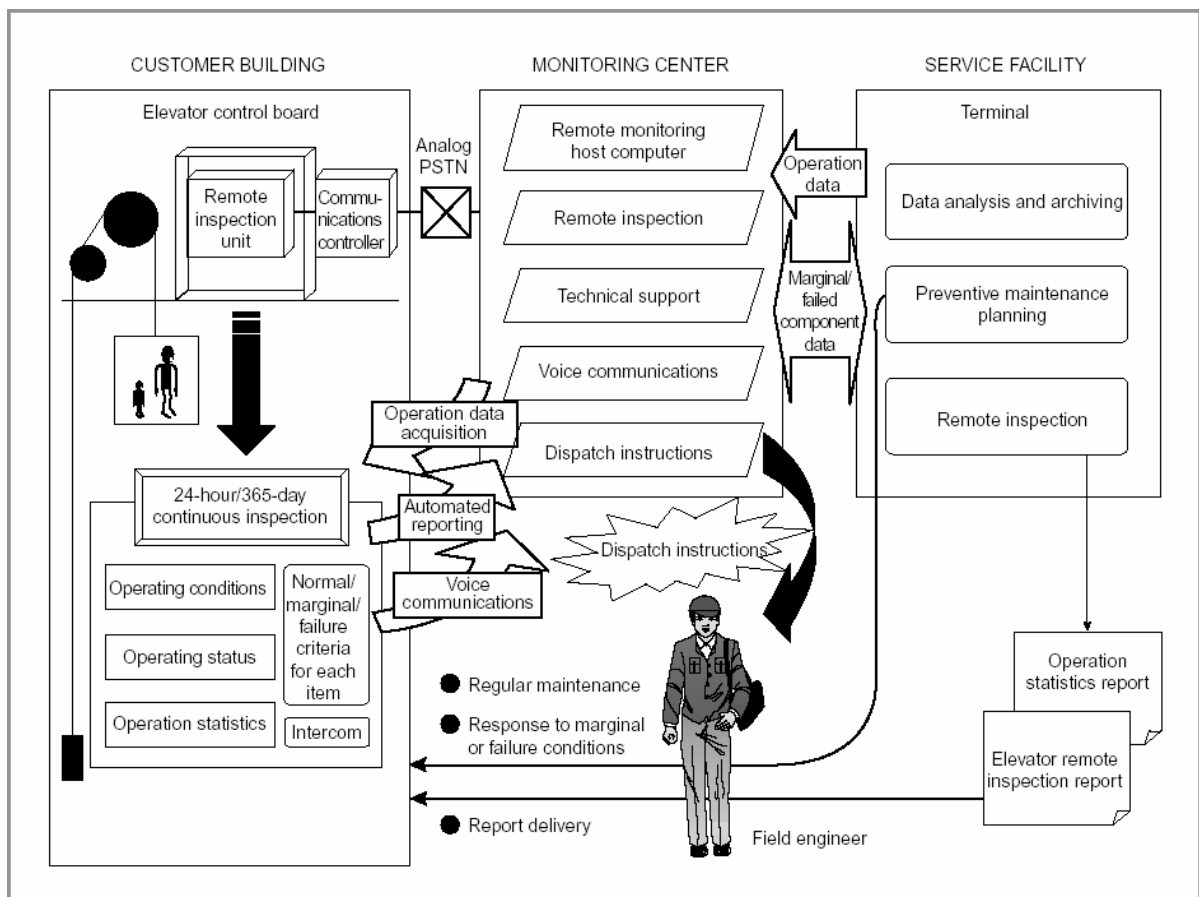


Figura 3 – Sistema da Mitsubishi

Esta solução não deixa de ser semelhante a outras já apresentadas, de onde se destacam apenas alguns pontos inovadores, nomeadamente a capacidade de monitorizar:

- a temperatura da casa das máquinas;
- o estado da válvula do travão;
- a falha de iluminação interior da cabina e
- o estado da luz de emergência e respectiva bateria.

Do mesmo modo que o sistema da OTIS, este oferece também a possibilidade de avaliação do desgaste de algum equipamento, dos quais se destaca:

- o número de travagens efectuadas;
- distância cumulativa percorrida pelo motor de tracção e
- precisão de paragem.

Este sistema permite também gerar automaticamente um relatório mensal da operação do elevador bem como das assistências efectuadas com o objectivo de informar os utentes e justificar gastos.

O sistema de monitorização de elevadores *MeEye* [6], também da Mitsubishi, foi o único desta empresa para o qual foi encontrada alguma referência à data de escrita da dissertação, 2006. Este sistema é baseado em tecnologia *web* e assenta na existência de um servidor *web* que concentra os dados e permite apresentar o estado de operação do elevador em tempo real, bem como imagens do interior do mesmo, obtidas através de uma câmara instalada no interior da cabina do elevador.

A configuração do sistema é apresentada na Figura 4. No edifício está instalado um PC que actua como servidor e monitoriza e controla todos os elevadores e escadas. Este computador recolhe, guarda e processa todos os sinais provenientes dos elevadores e comunica também com um PC do cliente, impressoras e outros periféricos que possam ser necessários para uma melhor análise e resolução dos problemas. O sistema é expansível, o número de PCs do cliente pode aumentar para um edifício com dois ou mais compartimentos de monitorização.

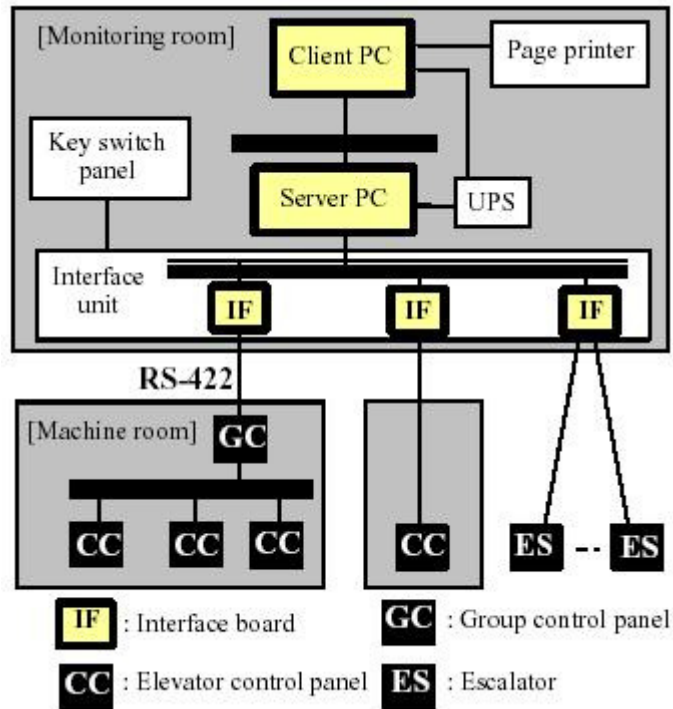


Figura 4 – Configuração do sistema de monitorização de elevadores e escadas *MeEye*

A configuração do *software* do sistema é apresentada na Figura 5. O *software* consiste em três unidades fundamentais: a unidade de *display*, que permite apresentar os dados monitorizados e páginas de configuração; o servidor, que processa e controla os dados monitorizados e determina as actuações a levar a cabo e a unidade de comunicação com os elevadores e escadas.

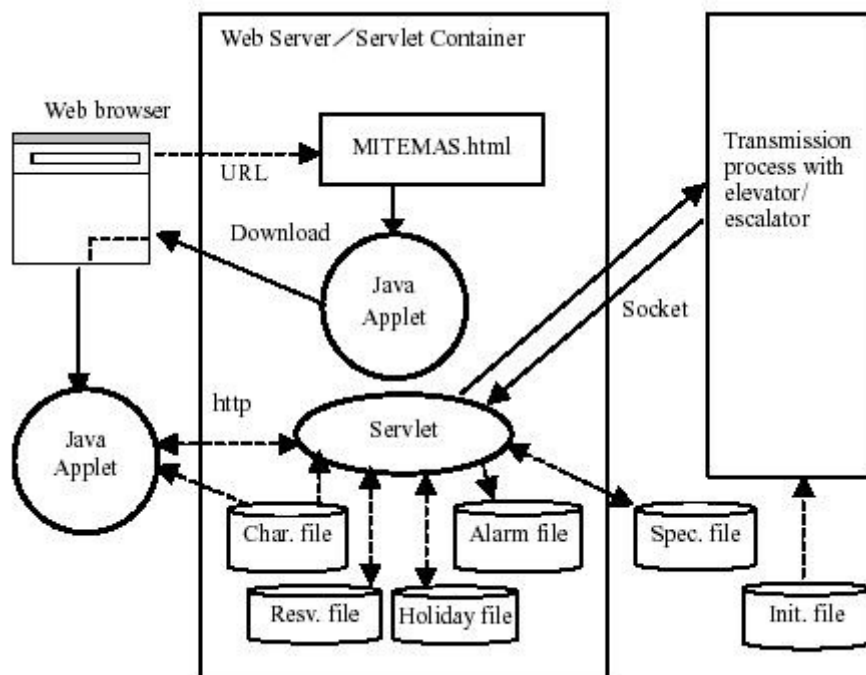


Figura 5 – Configuração do Software do MeEye

A apresentação gráfica deste sistema de monitorização permite uma visão global do conjunto de elevadores, apresentando o seu estado de um modo simples de entender para o utilizador e onde é possível efectuar a maior parte das operações através da utilização do rato.

O sistema de monitorização de Elevadores da Fujitec [7] (EMSTM), para o qual foi encontrada referência à data de escrita da tese, em 2006, permite apresentar representações gráficas dos controlos dos elevadores, tem funcionalidades de comunicação interactiva, capacidade de guardar dados de histórico e possui um mecanismo de registo de eventos para permitir responder a uma vasta gama de requisitos dos clientes da monitorização. O sistema funciona numa workstation Windows® NT e combina monitorização visual com controlo de comandos interactivos e análise de tráfego que permite a monitorização em tempo real dos eventos que ocorrem no elevador. O sistema permite assim modificar as operações do elevador, analisar o padrão de tráfego do mesmo e rever os eventos recentes de modo a poder alterar parâmetros do funcionamento e otimizar a performance do conjunto de elevadores.

O sistema pode ser configurado local ou remotamente através de uma *Local Area Network* (LAN) ou *Modem Accessed Network* (RAS). Os operadores autorizados podem assim aceder aos dados e partilhar os mesmos entre todas as estações ligadas entre si.

Neste sistema a monitorização visual é considerada de grande importância, permitindo reconhecer facilmente os parâmetros de funcionamento através de um código de cores. Através desta

funcionalidade é possível visualizar que chamadas a elevadores foram efectuadas e em que andares, que elevadores foram escolhidos para responder a essas chamadas e o modo de operação do conjunto de elevadores e de cada elevador em particular: a sua actual posição, o sentido em que se está a movimentar em cada instante e o estado de abertura ou fecho da porta.

Paralelamente à monitorização visual, o sistema permite modificar os parâmetros do conjunto de elevadores, *on-line*, de modo a alterar algum comando que não está a conduzir ao funcionamento pretendido e/ou de modo a configurar o seu funcionamento de acordo com as exigências de cada edifício em particular. Cada alteração pode ser configurada para ocorrer de imediato ou ser agendada para uma determinada altura com uma determinada periodicidade.

O sistema descrito possui ainda uma vertente de análise de tráfego. Esta funcionalidade permite, através da análise de dados recolhidos ao longo do tempo e a partir de vários elevadores, determinar o conjunto de parâmetros de funcionamento que conduz a uma utilização mais eficiente. Esta eficiência pode-se medir em termos de tempo de resposta às chamadas, tempos de abertura e fecho as portas ou tempos de transporte de andar para andar.

A análise dos dados recolhidos relativos aos parâmetros de funcionamento dos conjuntos de elevadores pode ser ainda utilizada para diagnóstico de possíveis situações de mau funcionamento.

O QarVision™ [8] constitui um sistema de diagnóstico remoto para elevadores, para o qual apenas foram encontradas referências em 2006. Este sistema permite que os técnicos que o utilizam efectuem o seu trabalho remotamente. O QarVision™ está dividido em duas partes: os sensores, os processadores que permitem o controlo e a tomada de decisões e os mecanismos de armazenamento de dados encontram-se numa pequena caixa à prova de água que pode ser colocada na base ou no topo do elevador. O painel de controlo e a *interface* com o utilizador é efectuada através de *PDA*s, *laptops* ou *desktop PCs*. Estes dispositivos comunicam utilizando rede *wireless WiFi*, *wired LAN*s ou internet. Na Figura 6 é apresentado um diagrama deste sistema.



Figura 6 – Sistema QarVision da Gameleon

O QarVision™ pretende ser um complemento aos sistemas de monitorização de elevadores existentes disponibilizando funcionalidades que estes últimos não possuem.

Este sistema pode ser instalado em qualquer tipo de elevador, incluindo elevadores já antigos. Utiliza sensores que fazem parte da unidade instalada no elevador e que podem facilmente ser instalados temporariamente na porta ou no mecanismo de locomoção. Permite avaliar os movimentos do elevador e comparar os mesmos com o que o controlador manda fazer, obtendo-se assim um *test drive* que permite ajustar parâmetros de funcionamento a partir da casa das máquinas ou mesmo do escritório.

O QarVision™ permite registar os seguintes eventos:

- data e hora de cada viagem;
- aceleração e desaceleração do elevador;
- corrente eléctrica no motor;
- temperatura do motor;
- pressão hidráulica;
- temperatura nos travões e na casa das máquinas;
- tempos de abertura e fecho de porta;
- duração das viagens;
- número de viagens efectuadas;
- número de actuações na porta;
- tempo total de funcionamento;

- velocidade.

Este sistema tem ainda a possibilidade de actuar nos alarmes do elevador.

A análise de todos os dados pode ser efectuada utilizando ferramentas gráficas que permitem uma visualização mais clara dos mesmos.

O *S3 Analyzer* é o sistema de monitorização remota da *wurtec* [9]. As referências para este sistema surgiram apenas em 2006, à data de escrita da dissertação. É baseado num microprocessador e foi projectado para poder efectuar diagnóstico remoto em qualquer conjunto de elevadores existente. O *S3 Analyzer* possui mecanismos de armazenamento de dados precisos e um osciloscópio virtual que permite facilitar a análise dos dados obtidos. O acesso ao sistema é limitado e a configuração do mesmo pode ser efectuada remotamente. Este sistema possui 16 entradas que podem ser ligadas a pontos específicos do controlador do elevador que correspondem a sinais cujo comportamento é conhecido. Quando determinadas situações, configuradas no sistema, se verificam, é enviado um e-mail para o técnico de monitorização, assegurando uma rápida resposta aos problemas.

Como resultado da pesquisa efectuada em 2003, foi possível especificar um sistema, supostamente ideal, capaz de reunir o conjunto de funcionalidades dos sistemas encontrados na altura.

Sendo assim, de acordo com o *Estado-da-arte* dos sistemas de monitorização de elevadores em 2003, o sistema mais completo devia ser baseado numa arquitectura modular de modo a facilitar, por um lado, a manutenção e a substituição de módulos danificados, e por outro, a expansão do mesmo.

O sistema deveria possuir as seguintes funcionalidades:

- possibilidade de configuração remota dos módulos do sistema (entradas, saídas, etc);
- recolha automática dos dados relativos ao estado do elevador;
- imediata transmissão de alarmes e estado do elevador para o centro de supervisão;
- sinalização de alarme de passageiro preso em elevador bloqueado;
- comunicação por voz, bidireccional, com os passageiros presos num elevador bloqueado;
- sinalização de avaria no sistema de alarmes do elevador;
- detecção de falha de iluminação interior da cabina;
- monitorização do estado da luz de emergência e respectiva bateria;

- detecção de ocorrências com duração parametrizável:
 - porta aberta durante mais de 50 segundos;
 - pressão do botão de emergência durante mais de 3 segundos;
- detecção de falha de tensão de alimentação no controlador do elevador;
- detecção de falha da tensão de alimentação do sistema, 220 V (AC);
- detecção de falha no relé de segurança;
- monitorização do estado da válvula do travão;
- monitorização da temperatura da casa das máquinas;
- possibilidade de contabilizar:
 - o número de travagens efectuadas;
 - a distância cumulativa percorrida pelo motor de tracção;
- possibilidade de supervisionar a actividade do elevador “*on line*”;

Cada alarme sinalizado na central deveria ter uma base de dados associada que permitisse a disponibilização e consulta da seguinte informação:

- número do elevador;
- número do contrato;
- morada do elevador de onde provém o alarme;
- descrição da falha;
- número de telefone da empresa de manutenção da área;
- número de telefone do porteiro do prédio;
- data e hora da ocorrência;
- hora do reconhecimento do alarme por parte do centro de monitorização.

Toda a informação relevante deveria ser guardada, periodicamente, em disco de modo a permitir gerar o relatório das assistências técnicas, bem como do histórico do estado de operação do elevador.

O sistema deveria ainda possuir uma bateria capaz de assegurar o seu funcionamento no caso de falha de energia, (permitindo o envio da informação recente para a central de monitorização).

Relativamente aos sistemas encontrados em 2006, as principais diferenças relativamente aos sistemas de 2003 estão relacionadas com as funcionalidades da aplicação de monitorização. Os

sistemas de recolha de dados não evoluíram significativamente, possivelmente porque apresentavam já a flexibilidade e robustez desejadas. As aplicações de monitorização, por seu lado, permitem agora uma grande variedade de opções gráficas de visualização dos dados, de forma correlacionada, permitindo detectar anomalias mais facilmente.

Adicionalmente, a proliferação das tecnologias *web* permitiu que hoje os sistemas de monitorização remotos possam estar acessíveis em qualquer lugar, sem que a monitorização esteja restrita às centrais, e através de ligações seguras que restrinjam o acesso aos profissionais autorizados.

3 Especificação Funcional do Sistema

3.1 Introdução

Tendo como base, por um lado os resultados da pesquisa efectuada em 2003 e, por outro lado, as indicações e expectativas fornecidas pelos representantes da Liftech, empresa que actua neste projecto como o cliente do produto a desenvolver, foi elaborada a especificação funcional do sistema.

Neste capítulo são descritos separadamente cada um dos módulos que compõem o sistema, quer ao nível da sua arquitectura, quer ao nível do software de suporte.

3.2 Descrição global do sistema

Este sistema tem como objectivo a supervisão remota de elevadores, permitindo não só a monitorização e registo de eventos, como também a actuação remota de comandos e o registo de assistências técnicas.

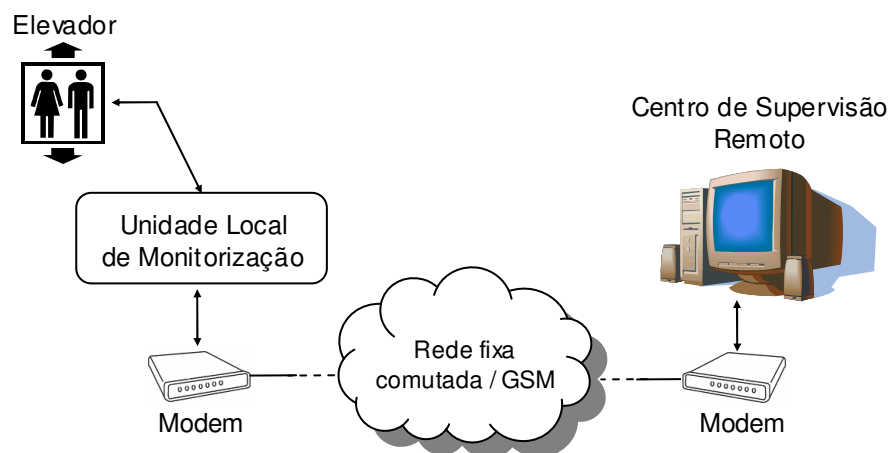


Figura 7 – Diagrama de blocos do sistema

Tal como é apresentado pela Figura 7, o sistema é composto por uma Unidade Local de Monitorização (ULM), a qual é colocada perto do elevador (ou conjunto de elevadores), e um Centro de Supervisão Remoto (CSR) constituído por um PC localizado remotamente. A comunicação entre ambos é efectuada através de rede fixa comutada ou rede GSM.

3.3 Unidade Local de Monitorização

Esta unidade é responsável pela monitorização e registo local de eventos e tem capacidade de reportar alarmes para o centro de supervisão, actuar em saídas a pedido da central e fazer o registo de assistências técnicas. O registo de eventos e de assistências técnicas constitui um histórico que, a pedido do centro de supervisão, é enviado para o mesmo. Cada evento é acompanhado da data e hora da ocorrência, com resolução de 250ms.

A sua implementação é baseada numa arquitectura distribuída que adopta um modelo funcional do tipo *master/slave*, podendo a unidade local integrar:

- 1 módulo principal;
- 1 módulo de comunicações;
- Até 30 módulos I/O (entradas/saídas digitais);

tal como é apresentado pela Figura 8.

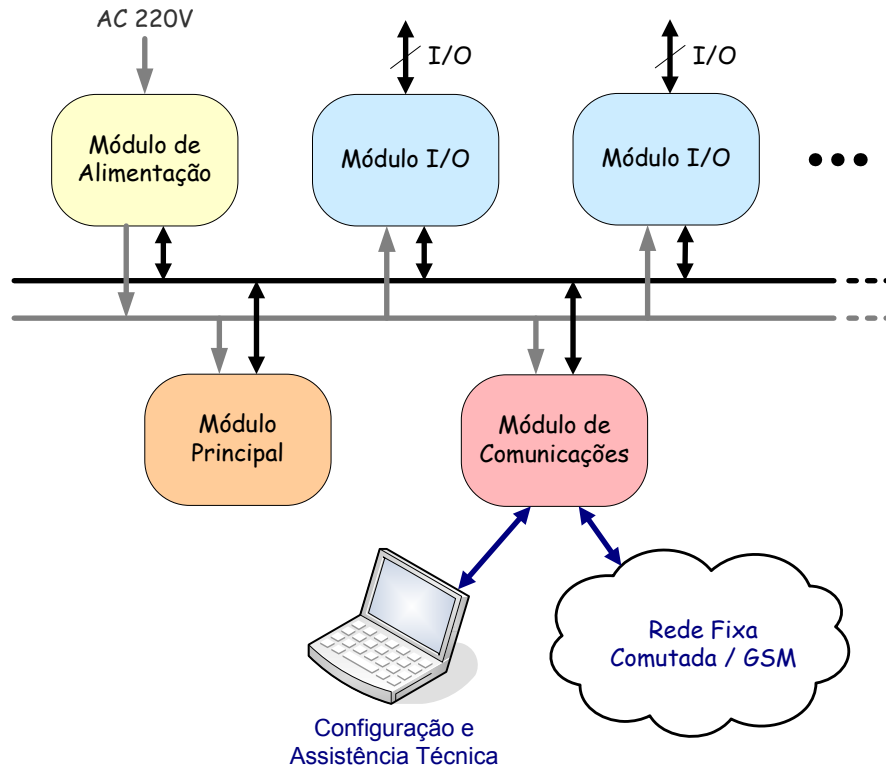


Figura 8 – Diagrama de blocos da Unidade Local de Monitorização

3.3.1 Atravancamento do sistema

Os vários módulos da ULM são colocados numa caixa que inclui calhas de suporte para cada um (*rack*), e um *backplane* que suporta a ligação dos barramentos de dados e alimentação a cada módulo.

Cada *rack* tem 8 *slots* (calhas de suporte). Cada *slot* tem associada uma identificação, definida no barramento, constituída por 5 sinais digitais, capazes de codificar até 32 módulos distintos. Esta solução, por um lado permite o *plug-and-play* físico do sistema, por outro, define o número máximo de módulos que este pode incluir.

Ao todo o sistema poderá ser constituído por um máximo de 4 *racks*, sendo que o número de *racks* efectivamente usadas dependerá do número de módulos I/O necessários em cada aplicação prática.

A ligação dos módulos de I/O e Comunicação ao mundo exterior será efectuada através de um interface físico no painel frontal.

3.3.2 Barramento de interligação entre módulos

Este barramento transporta as tensões de alimentação dos vários módulos e funciona como barramento de comunicação entre eles. A tecnologia do barramento de comunicação entre módulos será apresentada numa fase posterior desta dissertação.

3.3.3 Módulo Principal

Este módulo é responsável pela coordenação de todo o sistema. É ele que faz toda a gestão de eventos, desde a detecção ao seu tratamento, que possui em memória quer a configuração referente à ULM quer o histórico de eventos, e que controla a comunicação entre os módulos actuando como *master*.

A sua configuração e manutenção pode ser efectuada localmente através de um Laptop, ou remotamente através do centro de supervisão. A ligação deste módulo aos referidos dispositivos de configuração local é efectuada através do Módulo de Comunicações, o qual disponibiliza ao exterior uma interface USB.

Este módulo disponibiliza no painel frontal 1 LED tricolor para sinalização do estado de funcionamento de todo o sistema local de monitorização.

Os eventos que podem ser detectados pelo Módulo Principal:

- Início/fim de um período de manutenção no local por parte de um técnico especializado;
- Ocorrência de erros durante o funcionamento da ULM;
- Ocupação da memória de histórico acima de um *threshold* predefinido;
- Detecção de ocorrências associadas a cada porto de entrada.

Relativamente aos eventos provenientes dos Módulos I/O, estes podem ser constituídos por uma combinação de uma ou mais ocorrências, até um número máximo de 8 por evento. Os eventos apenas são gerados quando a combinação de ocorrências que estiver prédefinida se verificar verdadeira.

A detecção dos eventos correspondentes ao início/fim de um período de manutenção no local, por parte de um técnico especializado, é assegurada pelo Módulo Principal através de uma interface com um dispositivo de identificação do tipo *iButton* ID.

Tal como já foi referido, o Módulo Principal terá a seu cargo o armazenamento das configurações dos módulos da ULM e o armazenamento do histórico de eventos. Para garantir a integridade dos

dados a memória utilizada é do tipo EEPROM. Na Figura 9 é apresentada a estrutura da informação guardada na memória, a qual será descrita em mais pormenor no ponto 6.2.1.

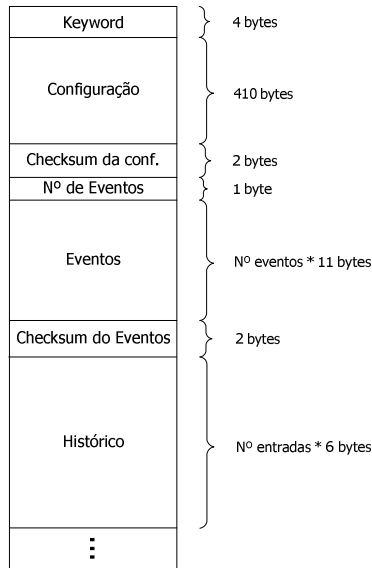


Figura 9 – Memory stack da EEPROM

Como o número de eventos configurado é variável, o espaço requerido para armazenar a configuração do sistema também o é. O espaço de armazenamento para o histórico (remanescente da memória) será assim tanto menor quanto maior for o número de eventos configurados. A configuração do sistema comporta no máximo 240 eventos distintos.

O espaço ocupado em memória pela configuração completa é dado pela expressão

$$C = 419 + n \cdot 11 \text{ (bytes)}$$

sendo 'n' o número de eventos numa dada configuração.

A capacidade de armazenamento da memória, em termos de número de entradas de histórico é dada pela expressão

$$(C_{Mem} - C) / 6 \text{ (entradas)}$$

sendo 'C_{mem}' a dimensão total da memória expressa em bytes.

3.3.4 Módulo de Comunicações

Este módulo é responsável por todas as comunicações com o exterior, ou seja, pelas comunicações entre a Unidade Local de Monitorização e o Centro de Supervisão e entre a Unidade Local de Monitorização e o PC de Assistência. O módulo é constituído por um sub módulo de interface, que possui uma interface RS232 para ligação ao modem e uma interface USB para ligação a um PC de Assistência, e um modem externo que permite a comunicação com o Centro de Supervisão, tal como é descrito pela Figura 10.

O sistema suporta dois modems distintos, para rede fixa comutada ou GSM, em regime de mútua exclusão. Ambos devem disponibilizar uma interface RS232. A comunicação com o Centro de Supervisão é feita através de uma chamada de dados, no caso da utilização de um modem para rede fixa comutada, ou através de SMS's, se se utilizar um modem GSM.

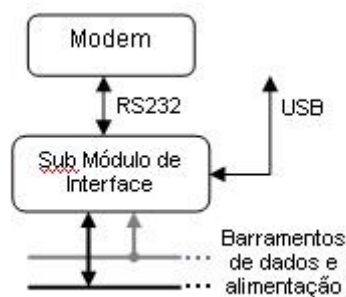


Figura 10 – Módulo de Comunicações

O Sub Módulo de Interface é responsável pela interface entre o modem e o Módulo Principal, tornando o sistema independente do tipo de modem (e respectiva rede de comunicação) que ali se encontra ligado. A interface é constituída por uma ligação série RS232. Este sub módulo possui ainda uma interface USB para permitir a ligação de um dispositivo de configuração local (Laptop). O atendimento a estas interfaces exteriores é efectuado em regime de mútua exclusão com precedência para o USB, de tal modo que a ligação de um dispositivo na interface USB determina a interrupção do atendimento ao modem.

O sub módulo dispõe ainda da capacidade de actuar um LED tricolor, com a finalidade de sinalizar o seu estado de funcionamento, e outros dois LEDs unicolores que sinalizam o funcionamento das linhas TX (transmissão) e RX (recepção) da interface RS232.

3.3.5 Módulo I/O

Cada módulo I/O é responsável pela interface física dos sinais a monitorizar ou controlar efectuando a detecção de ocorrências associadas. O módulo, caracteriza-se por:

- disponibilizar 8 entradas que permitem fazer o interface com contactos livres de tensão, em que a detecção é feita em loop de corrente; as entradas são isoladas opticamente.
- disponibilizar 2 saídas por relés, capazes de comutar uma corrente máxima de 5A a 250VAC.

É ainda disponibilizado um LED tricolor com a finalidade de sinalizar o estado de funcionamento do módulo, 10 LEDs para sinalizar o estado das entradas / saídas, e outro LED para sinalizar a presença da tensão de potência (+24V, desacoplada).

3.3.6 Módulo de Alimentação

A alimentação do sistema é suportada por uma fonte de alimentação de 24V DC. Esta fonte será usada para alimentar os circuitos de interface (entradas/saídas dos módulos I/O), desacoplados da restante electrónica do sistema, um modem e ainda, opcionalmente, um carregador de baterias de 12V DC. A partir a tensão de 24V é gerada uma outra tensão de 5V DC, desacoplada da primeira através de um conversor DC-DC. Esta segunda tensão é usada para alimentação dos vários módulos.

O sistema de alimentação pode incluir ainda uma bateria de 12V DC 1,2Ah que, no caso de falha da alimentação da rede, garanta o funcionamento parcial (suportar a alimentação do modem e da tensão de 5V DC) por um determinado período de tempo (dependente do número de módulos integrados).

A sinalização do estado de funcionamento deste módulo é feita através de dois LEDs, um para sinalizar a alimentação de rede e outro para sinalizar o estado de carga da bateria.

3.4 Centro de Supervisão

A monitorização remota é feita a partir de um Centro de Supervisão Remoto (CSR), dotado de um ou mais PCs, cada um capaz de se interligar a mais do que um sistema local de monitorização. A comunicação com as ULMs é feita através de um modem de rede fixa comutada, um modem

GSM, ou ambos, dado que cada centro de monitorização pode receber comunicação via rede fixa ou SMS.

Cada PC possui em disco uma base de dados referente às ULMs com as quais pode comunicar, podendo-se a partir deste centro configurar e parametrizar as mesmas, bem como gerir a base de dados correspondente a cada uma delas.

3.4.1 Configuração remota da Unidade Local de Monitorização

A configuração de cada ULM pode ser efectuada remotamente a partir do CSR. Do ponto de vista funcional, esta configuração envolve não só a criação de uma identificação de cada ULM registada em Base de Dados, como a especificação de todos os aspectos funcionais e de interligação eléctrica dessa mesma ULM.

Cada Unidade Local de Monitorização tem associado uma entrada numa base de dados existente no Centro de Supervisão Remoto. Cada entrada na base de dados tem por sua vez associados:

- O número de telefone do elevador (modem) ⇒ Campo obrigatório;
- O número de telefone do Centro de Supervisão Remoto;
- A morada do elevador de onde provém o alarme;
- O número do contrato;
- O número de telefone da empresa de manutenção da área;
- O número de telefone do porteiro do prédio;
- O tipo de modem presente na ULM;
- O número de tentativas para estabelecer chamada entre o CSR e a ULM;
- A percentagem de ocupação do histórico à qual deve ser enviado um alerta para a central;
- Uma cópia da configuração actual do sistema;
- Uma cópia de cada uma das configurações que o sistema já teve
- Uma cópia do estado actual das saídas.

A configuração dos Portos e Ocorrências de cada ULM permite:

- Associar uma descrição a cada porto;

- Configurar a ocorrência associada a cada porto:
 - Estado inicial com nível lógico '1' ou '0';
 - Estado a detectar com nível lógico '1' ou '0';
 - Parametrizar a duração temporal mínima de 250ms a 1 minuto, com resolução de 250ms.

A configuração de Eventos permite:

- A atribuição de um nome a um evento;
- Associar o evento a uma combinação lógica de ocorrências associadas a cada porto ou conjunto de portos de entrada;
- Definir se o evento deve ser registado no histórico;
- Definir se o evento deve reportar alarme;
- Para cada alarme, definir a mensagem de alerta;
- Definir se o evento deve conduzir à actuação numa saída, identificar a saída e o estado que se pretende que assuma.

4 Unidade Local de Monitorização – *Hardware*

4.1 Introdução

Neste capítulo é apresentada a fase de projecto da ULM ao nível do hardware. A escolha de determinados componentes do sistema envolveu um breve estudo na vertente funcional e económica, o qual é apresentado de seguida. O referido estudo teve o objectivo de adequar o trinómio funcionalidade, performance e custo do sistema global ao produto final pretendido. O projecto da ULM culmina com a apresentação da composição de cada módulo da ULM.

4.2 Breve Estudo Funcional e Económico de Componentes da ULM

4.2.1 Dispositivos de Identificação

A selecção do Dispositivo de Identificação esteve dependente de um conjunto de factores relacionados com o fim a que se destina o mesmo, bem como do local onde vai ser instalado:

- Dispositivo de Identificação:
 - Número relativo de dispositivos de identificação relativamente ao número de transdutores;
 - Informação que se pretende guardar nos mesmos;
 - Frequência da alteração da informação;
- Transdutor:
 - Número relativo de sistemas de leitura, fixos, relativamente ao número de dispositivos de identificação;
 - Características do ambiente onde vão ser instalados;
- Duração pretendida para cada um dos sistemas.

O sistema, quando instalado, deverá ser mantido por uma empresa de assistência técnica que deverá assegurar a assistência a um conjunto de elevadores de diferentes edifícios. Como a assistência deve efectuada em caso de avaria, solicitada pelo sistema de monitorização, ou periodicamente, para manutenção, cada técnico deve sustentar a manutenção de vários edifícios.

Assim, o número de transdutores a instalar deve ser bastante superior ao numero de dispositivos de identificação necessários, pelo que o custo do primeiro teve um maior peso na escolha do dispositivo a utilizar.

Cada dispositivo de identificação deverá conter a identificação do técnico assistente, pelo que não necessita de elevada capacidade de armazenamento de informação, nem de permitir alteração da informação nele contida.

O Transdutor deverá ser instalado na casa das máquinas. Este local mantém-se fechado sempre que não se está a efectuar a manutenção, estando portanto sujeito a acumulação de poeiras. O transdutor deve pois ser tanto quanto possível imune a estas condições. Esta imunidade pode ser conseguida à custa de ter um mecanismo que efectue a leitura sem contacto físico entre o dispositivo de identificação e o transdutor.

De entre as tecnologias de identificação existentes no mercado em 2003 foram destacados os cartões de identificação de banda magnética, os cartões de micro-chip, os cartões de rfid e os iButtons.

Os cartões de banda magnética [10], [11], constituem um dispositivo de identificação com elevada aplicação tanto no ano de 2003 como actualmente, desde os cartões bancários aos bilhetes de autocarro. A leitura da informação é efectuada fazendo passar a banda magnética do cartão por um dispositivo de leitura.

Os cartões de *micro-chip* [12], [13], [14], são também bastante utilizados como dispositivos de identificação. Estes cartões possuem uma capacidade de armazenamento mais elevada do que os cartões de banda magnética e permitem encriptação de dados com elevada segurança.

Os cartões de rfid [15], constituíam uma tecnologia emergente no mercado de 2003. Actualmente são muito utilizados como meio de identificação, controlo presencial e chave de acesso a zonas reservadas em edifícios públicos ou empresariais. A leitura da informação neles contida não necessita de contacto físico entre o cartão e o leitor, sendo efectuada através da emissão de radiação à frequência de rádio. O dispositivo torna-se então bastante resistente à agressividade do ambiente de instalação e prático do ponto de vista do utilizador.

Os *iButtons* [16], são dispositivos de armazenamento de dados, robustos, em forma de uma moeda, que actuam como um registo de identificação electrónica, possuindo um número único associado a cada dispositivo. A leitura da informação efectua-se através do contacto momentâneo com um dispositivo de leitura, o qual permite a transferência dos dados através de um protocolo série a dois fios.

Na Tabela 1 é apresentado um resumo das características destes dispositivos de identificação, bem como uma estimativa dos custos de cada tecnologia, reportando-se os valores ao ano de 2003.

	Custos		Vantagens	Desvantagens
	Transdutor	Dispositivo de Identificação		
Cartões de Banda Magnética	66€ – 120€	0.9€ – 8.8€	Baixo custo dos cartões	Sensível a ambientes agressivos
Cartões de Micro-Chip	16€ – 37€	1.9€ – 4.5€	Maior fiabilidade e durabilidade	Sensível a ambientes agressivos
Cartões de rfid	≈130€	≈2.8€	Bastante Imune a ambientes agressivos	Custo da tecnologia
iButtons	≈4.5€	≈1€	Baixo custo dos iButtons aliado à durabilidade e imunidade a ambientes agressivos	Pequena capacidade de armazenamento

Tabela 1 – Tabela Resumo das Características dos Dispositivos de Identificação

O dispositivo de identificação escolhido para efectuar a detecção de assistências técnicas em cada ULM foi o *iButton*. Esta solução foi inicialmente apontada pelos representantes da *Liftech* como uma solução de baixo custo e elevada resistência. Para além destes factores, o facto de o *iButton* conter um número único constitui a informação necessária e suficiente para a aplicação em questão. O baixo custo e a adaptação às condições do ambiente de instalação completaram as características que lavaram à adopção desta tecnologia.

4.2.2 Microcontrolador

O microcontrolador escolhido para integrar os Módulos Principal e I/O da ULM foi o PIC18F258 da *Microchip*, para o Módulo de Comunicações foi escolhido o PIC18F458, também da *Microchip*. A necessidade de escolher um microcontrolador diferente para o Módulo de Comunicações é explicada no capítulo 4.3.2. A escolha destes microcontroladores prende-se não só com razões de custo e adequação á aplicação em questão mas também com razões históricas relativas ao *know-how* acumulado. No Departamento de Electrónica, Telecomunicações e Infomática da Universidade de Aveiro (DETIUA) foram desenvolvidas, nos anos anteriores a 2003, várias aplicações tendo como base os microcontroladores da *Microchip*, incluindo os PIC18F258 e PIC18F458, o que permitiu testar este *hardware* e atribuir-lhe elevada fiabilidade. Para além disso, encontra-se já desenvolvido no DETIUA um significativo conjunto de *software* para esta plataforma computacional e de controlo, o qual já foi testado sendo considerado fiável. Estes microcontroladores possuem interface série RS232, interface SPI, controlador CAN, entre outras

características adaptáveis ao sistema que pretendemos desenvolver. Adicionalmente, o baixo custo da tecnologia (abaixo dos 5€) constituiu um factor de elevado peso na decisão.

Estes processadores possuem uma memória de código reduzida (32KB), o que não permite que o código seja muito extenso. Adicionalmente, a memória RAM disponível é também de capacidade muito reduzida (1536 bytes) dificultando a organização do código. Os processadores possuem ainda uma memória interna do tipo EEPROM. A sua reduzida capacidade (apenas 256 bytes), é, no entanto, insuficiente para guardar a informação relativa às configurações e ao histórico, obrigando à integração de uma memória externa, não volátil, para suprir essa necessidade. A escolha representa portanto uma solução de compromisso dado que acarreta a desvantagem de necessitar da integração de uma memória externa.

4.2.3 Barramento de Comunicação

Nesta secção são apresentados os tipos de barramentos de campo existentes no mercado, à data de realização do trabalho, permitindo no final concluir sobre a melhor opção/solução para a especificação do barramento físico da ULM.

O barramento *Foundation Fieldbus* [17] pode assumir vários modos de comunicação: cliente/servidor, produtor/consumidor (*publisher/subscriber*) e notificação de eventos.

Foundation Fieldbus tem dois protocolos de comunicação: o *Foundation Fieldbus H1* e o *Foundation Fieldbus HSE* (H2).

O protocolo H1 opera às velocidades de 31.25Kbps, 1Mbps e 2.5Mbps. O meio físico utilizado é o par entrançado ou a fibra óptica. A topologia de rede adoptada pode ser em estrela ou Bus e o tamanho das tramas é de 128 bytes. Possui detecção de erros de 16 bits CRC e suporta um número máximo de 240 nós por segmento de barramento.

O protocolo HSE é o protocolo *Ethernet* de alta velocidade. É utilizado no controlo de processos avançados, I/O remotos e automação de fabrico de alta velocidade. Para fornecer potência no modo de corrente usa um sinal AC de 16kHz. A tensão do sinal usada é de 5.5 a 9.0Vp-p. O meio físico utilizado é equivalente ao do protocolo H1, par entrançado ou fibra óptica. A topologia de rede que utiliza é a topologia em estrela. Usa o protocolo TCP/IP para a transmissão de tramas com tamanho variável. O acesso ao meio é feito através de um método de arbitragem CSMA/CD e utiliza um mecanismo de detecção de erros CRC [18].

O ControlNet [19] constitui um protocolo de comunicação a 5Mbps que permite prever a transmissão de dados e garantir a sua entrega ao destinatário. Apresenta uma performance

determinística e repetível, desejável em processos com restrições temporais rígidas. A rede suporta a integração de 48 nós, numa distância de 250 metros, os quais podem ser constituídos por simples dispositivos I/O ou controladores complexos. A topologia de rede pode ser em estrela, árvore, bus ou combinações destas topologias. O número de repetidores utilizado está sempre limitado pelo atraso que introduzem na comunicação. A rede utiliza grande largura de banda.

O protocolo Profibus [21][22] utiliza como meio físico a linha RS-485 ou a fibra óptica. O *Profibus* cobre não só as aplicações de dimensão industrial como também aplicações cobrindo áreas mais vastas. Opera a velocidades desde os 9.6kbps até aos 12Mbps, com um máximo de 126 nós no mesmo barramento. A família *Profibus* é composta por três protocolos:

- Profibus-DP: protocolo *master/slave* em que o *master* periodicamente sonda os *slaves*. Tem a possibilidade de assumir uma arquitectura *multi master* em que neste caso cada nó está associado a um *master* (todos os *masters* têm permissão de leitura nos vários nós, mas apenas um tem a permissão de escrita). Este protocolo está otimizado para transmissão de dados eficiente entre controladores e periféricos descentralizados. Pode ser usado para substituir um sinal paralelo com 24V ou de 0 a 20mA. A velocidade de transmissão é 9.6kbps a 1200m e 12Mbps a 100m.
- Profibus-PA: é um protocolo equivalente ao Profibus-DP mas possuindo funções de diagnóstico. É frequentemente utilizado em aplicações que exigem elevada segurança como controlo de processos químicos e petroquímicos. A velocidade de transmissão é de 31.25kbps.
- Profibus-FMS: é um protocolo *peer to peer* que permite comunicação entre *masters*.

O protocolo P-net [19] tem como base da sua especificação eléctrica a norma RS-485, permitindo o uso de um cabo de 1200m sem repetidores. Os dados são transmitidos sob a forma NRZ e é possível ligar 125 nós em cada segmento.

O barramento Swiftnet [19] foi desenvolvido para satisfazer as necessidades dos aviões comerciais da *Boeing* de possuir um barramento de comunicação que oferecesse garantias de sincronização, fiabilidade e transmissão a alta velocidade. Utilizando a norma RS-485, o comprimento do barramento vai desde os 360m com uma velocidade de transmissão de 5Mbps aos 4800m com uma velocidade de transmissão de 2.4 a 76kbps. Este barramento permite integrar numa rede até 30 milhões de nós. As performances atingidas com este barramento só são possíveis pela utilização de um protocolo estruturado de acesso ao barramento, TDMA (*Time Division Multiple Access*) síncrono.

O barramento WorldFIP [23][24] permite a transmissão de dados com restrições temporais críticas e de mensagens esporádicas. A camada física deste barramento admite uma taxa de transmissão de 31.25kbps, 1Mbps ou 2.5Mbps via par entrançado de cobre ou 5Mbps via fibra óptica. Na utilização deste barramento é permitida a inclusão de 64 nós por segmento de cabo, a inclusão de até quatro segmentos intercalados por repetidores e segmentos de 1Km, dependendo da taxa de transmissão, do tipo de cabo, e do número de nós. O protocolo de comunicação utilizado segue o modelo produtor/consumidor, possuindo um escalonador central.

O barramento FIPway [19] é baseado no WorldFIP e pode operar no modo cliente ou no modo servidor.

- No modo Cliente permite aceder a nós na rede e a nós pertencentes a sub-redes, utilizando endereçamento X-WAY. A comunicação permite multi-fluxo (máximo de 5 pedidos de leitura e 3 pedidos de escrita simultâneos). Para permitir a gestão de mais pedidos integra o conceito de *multi-frame*.
- No modo Servidor os clientes *FIPway* possuem uma Base de dados partilhada. Esta funcionalidade é utilizada para otimizar o *feedback* de dados. Assim, em vez de fazer o constante *polling* de dados para monitorizar o estado das variáveis, cada alteração constitui uma actualização da base de dados.

O *Interbus* [19] constitui um método de transmissão *I/O-oriented*, com um protocolo em malha fechada. O método de transmissão do *Interbus* possui as seguintes variantes físicas:

- Barramento remoto, para cobrir grandes distâncias;
- Barramento local, para comunicação na cabina de controlo (por exemplo);
- *Interbus loop*, para ligação de nós com pequeno número de entradas e saídas.

O barramento *Interbus* opera segundo o método *master/slave*, numa topologia em anel, ou seja, em que todos os nós estão activos numa malha fechada, admitindo um máximo de 512 nós.

O máximo comprimento num barramento remoto é de 400m entre dois nós e 12.8km no total. Num *Interbus loop* o máximo comprimento entre dois nós é de 20m e no total é de 200m. Este barramento tem uma velocidade de transmissão de 500kbps e 2Mbaud.

O protocolo *Interbus-S* [19], para rede distribuída, é baseado numa topologia em anel e tem uma taxa de transmissão de 500kbps. A topologia em anel permite a ligação dos nós em malha fechada. O cabo utilizado tem 5 fios e é permitida uma distância máxima de 400m entre dois nós,

utilizando a norma RS-485 e transmissão ponto a ponto. A integração de repetidores em cada nó permite uma extensão total do barramento de 13km e um número máximo de nós de 512.

O barramento I2C [20] constitui um barramento série a dois fios, bidireccional, para transmissão de dados (SDA) e transmissão de relógio (SCL), para controlo entre circuitos integrados.

Cada nó é identificado por um único endereço e pode operar como receptor, apenas, ou como transmissor com capacidade de receber e enviar informação. Tanto os receptores como os emissores podem operar com *master* ou como *slave*, dependendo se podem iniciar uma transferência de dados ou se isso só acontece quando são endereçados para o fazer. I2C é um barramento *multi-master*.

O barramento I2C básico opera a uma taxa de transferência de 100kbps e possui 7bits de endereço. Esta especificação foi melhorada passando a operar a velocidades até 3.4Mbps (*High-speed mode*) e com 10 bits de endereço.

O barramento CAN [25] é um barramento série que utiliza uma codificação de bit NRZ projectada para comunicações rápidas e robustas em ambientes electricamente agressivos. O barramento CAN implementa uma rede de comunicação entre nós independentes num sistema distribuído e opera a uma velocidade de transmissão até 1Mbps.

O protocolo implementa dois estados eléctricos, correspondendo a um bit dominante ('0') e um bit recessivo ('1') e um mecanismo de arbitragem em que as colisões das mensagens não afectam o *throughput* do barramento, CSMA/CA. Cada mensagem transmite entre 1 e 8 bytes de dados e possui um identificador de 11 bits (CAN 2.0A) ou 29 bits (CAN 2.0B). Cada nó decide pelo processamento ou descarte de uma determinada mensagem baseado no seu identificador.

Para implementar a comunicação entre os módulos constituintes da Unidade Local de Monitorização optou-se por utilizar o barramento CAN. Esta escolha prende-se com vários factores, a saber:

- As mensagens que se pretende fazer percorrer o barramento têm, na sua maioria, pouca informação a transmitir, em número de bytes de informação. Deste modo, não é necessário utilizar um barramento que permita a troca de mensagens de elevada dimensão.
- Não há necessidade de transmitir alimentação pelo barramento de comunicação.
- O sistema não está sujeito a restrições temporais rígidas, pelo que, não há necessidade de adoptar um barramento com comportamento determinístico em qualquer situação.

- Todo o sistema se vai concentrar num reduzido espaço físico.
- A topologia de rede que se pretende implementar é topologia bus.
- O protocolo de comunicação que se pretende estabelecer actua segundo o modelo *master/slave* relativamente à troca de mensagens de configuração, solicitação de informação e comunicação de ocorrências.
- O ambiente onde se pretende introduzir o sistema está sujeito a um elevado ruído electromagnético e a transmissão diferencial confere ao barramento CAN uma elevada imunidade ao ruído.
- Finalmente, existe por detrás desta escolha uma razão subjectiva bastante importante, relacionada com o *know-how* acumulado e com o baixo custo, associado ao *hardware* que se pretende utilizar (PIC18Fx58 da *Microchip*). Para além disso, existe *software* já desenvolvido para implementar toda a comunicação CAN, o qual já foi utilizado e testado tendo sido considerado fiável. Assim, a escolha do barramento CAN para a comunicação entre os módulos possibilitou ultrapassar o tempo de desenvolvimento que implicaria a escolha de qualquer outro dos barramentos referidos.

A escolha do barramento CAN não introduziu qualquer limitação adicional ao sistema a desenvolver, de modo que a adopção desta solução não implicou qualquer compromisso.

4.2.4 Memória não volátil

Tal como já foi referido no capítulo 4.2.2, o microcontrolador escolhido (PIC18Fx58) implica a introdução na ULM de uma memória externa não volátil para armazenamento de configurações e histórico de eventos. Para fundamentar a escolha foi efectuado um breve estudo económico das tecnologias existentes no mercado. No anexo A é apresentada uma listagem dos preços praticados por alguns fabricantes, à data de realização do trabalho, 2003.

Através do estudo efectuado, pode-se concluir que existe não só uma enorme diferença de preço entre as memórias do tipo *Flash* e EEPROM, como também as memórias do tipo *Flash* têm dimensões de no mínimo 512Kbit, o que excede à partida as necessidades requeridas, revelando-se portanto pouco adequado o seu uso na aplicação.

Relativamente às memórias do tipo EEPROM, podemos verificar que não existem diferenças técnicas significativas entre os vários modelos apresentados no anexo A. No entanto, a diferença de preço registada chega a ser, por vezes, o dobro entre memórias de diferentes marcas, e insignificante ou nula entre memórias com diferentes capacidades pertencentes ao mesmo fabricante.

As memórias da *Microchip* são as mais baratas de entre as encontradas, apresentando ainda a vantagem de pertencer a um fabricante com renome afirmado no mercado, conferindo uma certa confiança na fiabilidade dos componentes, pelo que foram as escolhidas para fazer parte do sistema desenvolvido.

4.2.5 Interfaces das Entradas e Saídas

A proposta de desenvolvimento do projecto especificava que as interfaces com o sistema de elevadores deveriam ser constituídas por entradas isoladas opticamente e saídas por relé (contactos secos).

Nas entradas, o desacoplamento óptico permite manter o restante circuito do módulo protegido. No caso de ser ligado na entrada um sinal fora do perfil esperado, apenas é danificado o fotodíodo que faz a transmissão do sinal para o circuito do microcontrolador.

Nas saídas, para manter a protecção, foram adicionados acopladores ópticos para interface com os relés, ver Figura 11.

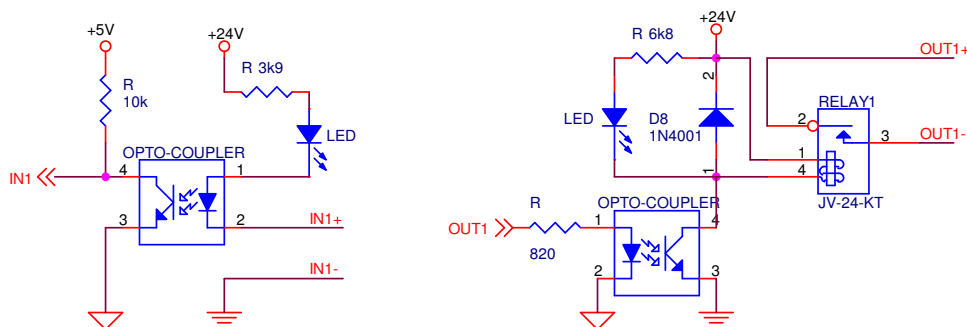


Figura 11 – Esquemático dos circuitos de interface das entradas e saídas, respectivamente

Durante o projecto da ULM surgiu a possibilidade de disponibilizar, em cada módulo, 6 entradas e dois portos configuráveis como entradas ou saídas. No entanto, após algum estudo da configuração do circuito, verificou-se que:

- para garantir o *switch* do porto (entrada ou saída),
- para efectuar uma correcta detecção da entrada garantindo não haver interferência por parte do circuito associado à saída e
- para garantir a protecção do restante circuito equivalente em todos os portos (*interfaces* opticamente isoladas),

era necessário ter o equivalente a 10 circuitos de interface para entradas, fazendo uso de apenas 8 (no máximo). Assim, considerou-se que seria preferível não ter portas configuráveis, permitindo as mesmas funcionalidades já propostas, dado que esta segunda opção resulta numa poupança nos circuitos de interface. Na solução final são disponibilizadas então 8 entradas e 2 saídas por cada módulo I/O.

4.2.6 Módulo USB

Uma vez que já à data e desenvolvimento do sistema, e também no contexto actual, a porta série RS232 começava a entrar em desuso como interface disponível num Laptop, foi equacionada a hipótese de fazer a ligação ao PC de configuração local através de uma interface USB. Esta proposta foi, inclusivamente, apresentada pelos responsáveis da Liftech, como uma mais valia do sistema, conferindo-lhe uma maior actualidade e compatibilidade com os dispositivos futuros.

Para implementar a comunicação USB optámos por utilizar um módulo da FTDI (FT245BM USB FIFO (USB - *Parallel*) I.C.) [32]. Este módulo possui *device drivers* que mascaram a porta USB como uma porta COM virtual. Assim, do ponto de vista da aplicação, o periférico ligado à *interface* USB comunica como se se tratasse de uma *interface* RS232.

4.3 Set de Módulos da ULM

4.3.1 Módulo Principal

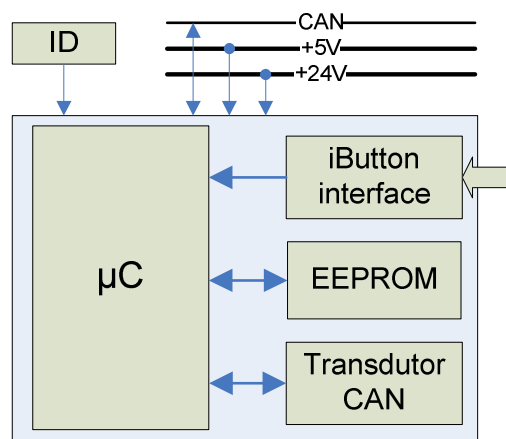


Figura 12 – Esquema do Módulo de Comunicações

O Módulo Principal inclui, para além do microcontrolador que terá a responsabilidade de coordenar toda a ULM, uma EEPROM para armazenamento do *setup* da ULM e do histórico de ocorrências, uma ligação a um leitor de *iButtons* para detecção de assistências técnicas e um transdutor CAN para adaptar os sinais para comunicação com os restantes módulos. A *Rack* fornece ao módulo uma identificação *hard-wired*, dependente da posição onde este se encontra, as alimentações desacopladas de 5V e 24V e as linhas de comunicação CAN.

Numa fase inicial do projecto o Módulo Principal possuía uma interface RS232 com o objectivo de fazer a ligação ao PC local para efectuar a (re)configuração da ULM e *download* do histórico. Posteriormente esta função foi transferida para o Módulo de Comunicações que ficou então responsável por todas as ligações com o exterior.

O esquemático completo do Módulo Principal é apresentado no Anexo B.

4.3.2 Módulo de Comunicações

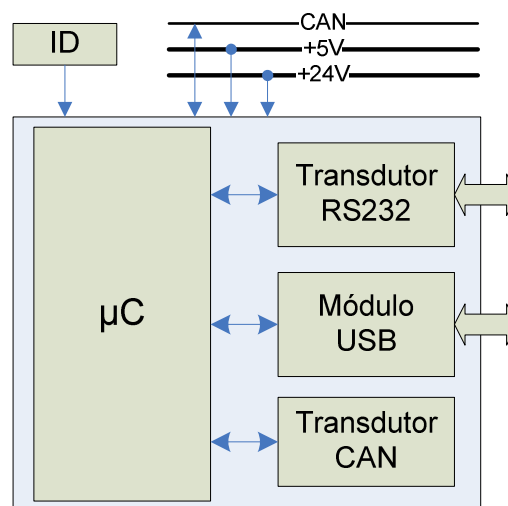


Figura 13 – Esquema do Módulo de Comunicações

O Módulo de Comunicações tem a seu cargo toda comunicação com o exterior. Possui então, para além do microcontrolador e do transdutor CAN para comunicação entre os módulos, um transdutor RS232 e a respectiva *interface* para ligação ao *modem*, e um módulo USB para ligação ao PC de configuração.

O microcontrolador utilizado neste módulo é o PIC18F458 da *Microchip*. Este microcontrolador apresenta 16 portos adicionais relativamente ao PIC18F258, permitindo a ligação do módulo USB o qual possui uma interface paralela constituída por um barramento de 13 bits.

A *Rack* fornece, tal como ao Módulo Principal, a identificação *hard-wired*, dependente da posição onde o módulo se encontra, as alimentações desacopladas de 5V e 24V e as linhas de comunicação CAN.

O esquemático completo do Módulo de Comunicações é apresentado no Anexo B.

4.3.3 Módulo I/O

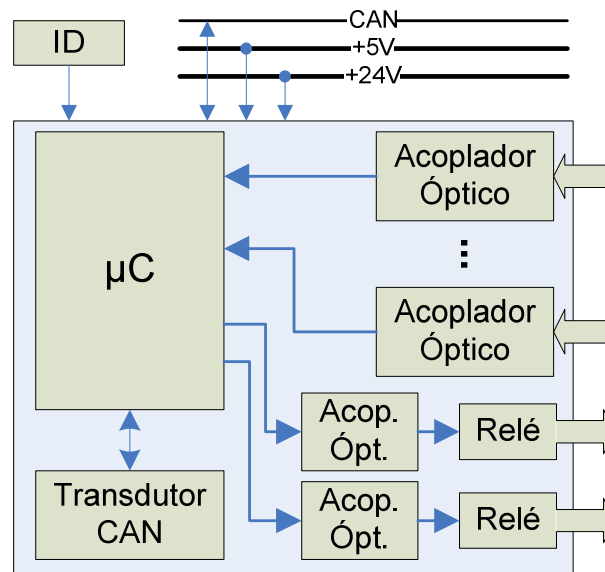


Figura 14 – Esquema do Módulo I/O

Cada Módulo I/O faz a *interface* com os sinais do sistema de elevadores, pelo que possui as *interfaces* já descritas em 4.2.5 para desacoplar os dois sistemas e proteger os circuitos internos da ULM. Cada Módulo I/O possui também o respectivo microcontrolador e o transdutor CAN.

Cada módulo recebe ainda da *Rack*, tal como os outros módulos, uma identificação *hard-wired*, dependente da posição onde o módulo se encontra, as alimentações desacopladas de 5V e 24V, para alimentar os circuitos internos do módulo e os circuitos de *interface* com o sistema de elevadores, e as linhas de comunicação CAN.

O esquemático completo de um Módulo I/O é apresentado no Anexo B.

4.3.4 Módulo de Alimentação

Para a alimentação do sistema, vão ser necessárias duas tensões desacopladas de 24V DC e 5V DC. Adicionalmente, se se considerar a hipótese de integrar uma bateria de *backup*, é necessário uma tensão variável de 12V-15V para efectuar a carga da mesma e uma unidade inteligente de

controlo de carga. Nesse caso a solução a adoptar passa por ter apenas uma fonte 24V DC, a qual suporta a carga da bateria (12V-15V DC), e ainda a alimentação de 5V, esta desacoplada por um conversor DC-DC. Desta forma a comutação entre a bateria e a fonte de tensão é feita apenas por 2 díodos (ver Figura 15).

O conversor DC-DC tem de suportar uma tensão de entrada de 10V-25V (limites). Os conversores DC-DC apresentam a vantagem de terem uma eficiência elevada (em regra superior a 80%), para além de assegurarem o necessário desacoplamento das duas alimentações.

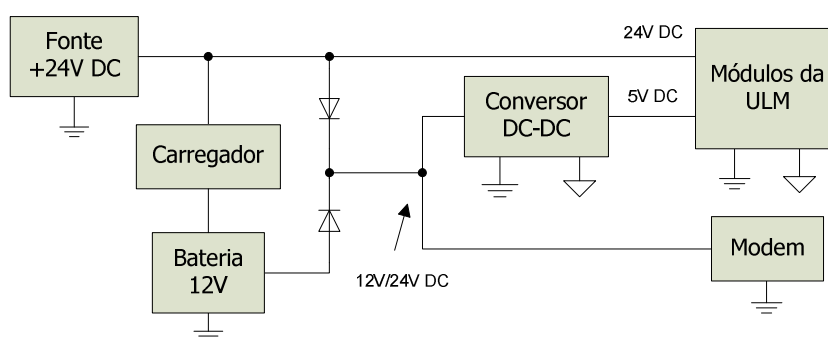


Figura 15 – Esquema do Módulo de Alimentação

A alimentação dos *modems* é garantida ou pela tensão de 24V DC ou pela bateria de 12V DC, conforme a alimentação da rede estiver disponível ou não, respectivamente. O modem GSM utilizado para testes (Siemens TC35) aceita uma tensão de entrada de 8V a 30V DC, permitindo ser alimentado do modo apresentado na Figura 15. O modem de rede fixa deverá ter especificações equivalentes para ser compatível com o módulo de alimentação apresentado.

A fonte de 24V DC deve ser capaz de fornecer cerca de 60W, os quais serão distribuídos pelos vários circuitos da seguinte forma:

24V → ≈2,1A;

5V → ≈1,1A;

12V → 500mA (*modem*).

A solução apresentada constitui apenas uma proposta de implementação, uma vez que não foi concretizada. A opção de não construir este módulo prende-se, por um lado, pelo facto de se considerar que o objectivo do projecto é, em primeiro lugar, o desenvolvimento do sistema de monitorização e, por outro lado, pelo facto de a construção deste módulo ser apenas uma integração de dispositivos já existentes no mercado, não exigindo o desenvolvimento de nova tecnologia ou configuração. Considerou-se então suficiente especificar as tensões de alimentação

necessárias e a potência requerida pela ULM, deixando ao utilizador do produto a responsabilidade de alimentar o sistema segundo as especificações.

4.3.5 Set de Módulos Mínimo

O Set de módulos mínimo necessário ao funcionamento de uma ULM é de 1 Módulo Principal, 1 Módulo de Comunicações, 1 Módulo I/O, 1 *Rack* e um módulo de alimentação (que no set mínimo não inclui a bateria de *backup*). A *Rack* terá sempre o tamanho standard que possibilita a ligação de 8 módulos, constituindo uma peça única. Esta ULM mínima permite monitorizar um conjunto de 8 sinais e actuar em 2 comandos. Caso a ULM possua um modem é possível efectuar a monitorização a partir da Central de Supervisão.

4.4 Protótipo da Unidade Local de Monitorização

Após a determinação da composição do sistema foi construído um protótipo da ULM para testes e desenvolvimento do *software* de controlo. Este protótipo é constituído por 1 Módulo Principal, 1 Módulo de Comunicações e 3 Módulos I/O, dos quais 2 possuem 2 entradas e 2 saídas e o outro possui 3 entradas e 1 saída, para permitir parametrizar eventos com maiores combinações de ocorrências. Foi ainda construída uma *Rack* para suporte, alimentação, comunicação e identificação dos módulos.

Os protótipos do Módulo Principal e dos Módulo I/O possuem um transdutor RS232 e respectiva interface para permitir a reprogramação do microcontrolador e o *debug* do *software*.

A alimentação da ULM protótipo, para efeitos de desenvolvimento do *software* e *debug*, foi suportada por fontes de alimentação disponíveis no laboratório.

Os esquemáticos completos dos módulos protótipo produzidos são apresentados no anexo B.

5 Especificação do Protocolo de Comunicação entre os Módulos

5.1 Introdução

A especificação do protocolo de comunicação é influenciada pelas características dos módulos constituintes do sistema distribuído e também pelas características particulares dos vários cenários de utilização possíveis. Nesta secção é apresentada uma análise da comunicação nos vários cenários de utilização e o modelo de comunicação adoptado.

5.2 Cenários de Utilização

5.2.1 *Power up da Unidade de Monitoração Local*

Aquando do primeiro *power up* da ULM o sistema não está configurado. Inicialmente é efectuado um reconhecimento, pelo módulo principal, da disposição física da ULM.

A detecção dos restantes módulos por parte do módulo principal é efectuada seguindo os seguintes passos:

O *master* deve enviar para o barramento uma mensagem com o comando IDREQ (ID *Request*), percorrendo sequencialmente todos os endereços possíveis, excluindo o seu próprio endereço.

Cada um dos módulos deve enviar uma mensagem de resposta (comando → IDRESP (ID *Response*)) para o ID do *master* (que passou a conhecer através da mensagem recebida), enviando o seu ID (1 byte) e o tipo de módulo que constitui (1 byte, 0x01 → Módulo I/O, 0x02 → Módulo de Comunicações e 0x03 → Módulo Principal).

A configuração dos módulos será efectuada posteriormente a partir de um PC.

Num *power up* após falha de alimentação a situação é equivalente à anterior, exceptuando o facto de cada ID ter já associada uma parcela da base de dados de configuração da ULM. O *master* deve então enviar IDREQ percorrendo todos os IDs, de modo a validar a antiga disposição dos módulos.

A configuração presente em cada módulo, exceptuando a do módulo principal, apenas reside em memória volátil, pelo que se torna necessário efectuar nova configuração todas as vezes que é efectuado novo *power up*. Neste caso a reconfiguração está a cargo do módulo principal, dado que este possui a anterior configuração de toda a ULM guardada em memória.

Uma falha de alimentação pode estar associada a uma operação de manutenção, a qual pode ter incluído remoção ou adição de módulos. No caso da disposição física da ULM ter sido alterada, é necessário reconfigurar todos os módulos para que a ULM possa funcionar. A nova disposição será automaticamente detectada (operação já descrita que está a cargo dos módulos) e disponibilizada ao *software* do PC, para permitir a reconfiguração.

5.2.2 Configuração dos módulos

A configuração dos módulos compreende a configuração das ocorrências associadas aos portos de entrada dos Módulos I/O, definição do estado dos portos de saída dos mesmos e a configuração do Módulo de Comunicações.

A configuração dos módulos I/O implica o envio de informação relativa aos itens *estado a detectar*, *estado inicial* e *duração temporal mínima* para parametrização das ocorrências e *estado* dos portos de saída. As mensagens incluem ainda, o comando CFG (Configuração) e a identificação do módulo destinatário.

As mensagens de configuração do Módulo de Comunicações incluem, para além do identificador e do comando de configuração, o tipo de *modem* ligado (GSM ou PSTN), o número de tentativas de ligação aquando da falha de comunicação com o CSR, o número de telefone da ULM, o número de telefone do CSR e o PIN do cartão (caso o *modem* seja GSM).

O comando de configuração e o identificador da mensagem (correspondente ao identificador do módulo em questão) fazem parte do campo de identificação da mensagem CAN. Numa mensagem CAN 2.0A [25] o campo de identificação tem 11 bits, como o ID de cada uma das mensagens ocupa 5 bits (identificação do módulo) restam 6 bits para codificação do comando.

5.2.3 Comunicação de ocorrências

O modelo de comunicação adoptado para a comunicação de ocorrências é um modelo de comunicação por *polling*. Segundo este modelo, será o Módulo Principal (*master*) a solicitar informação a cada um dos módulos I/O, com o objectivo de manter actualizada a base de dados. O período de cada pedido é de 250ms (não se trata de um sistema que exija uma rápida

reactividade, pelo que, o facto de a comunicação não ser *event-driven*, não deverá afectar significativamente a qualidade do serviço prestado).

As mensagens de pedido de informação por parte do *master* devem incluir o identificador do módulo e o comando respectivo (MIO_IN_REQ → Módulo I/O In Request).

As mensagens de resposta enviadas por cada módulo I/O devem incluir, do mesmo modo o identificador e o comando MIO_IN_RESP (Módulo I/O In Response) e um byte de dados em que cada bit representa um porto e sinaliza a detecção de ocorrência: bit a '0', não se detectou ocorrência respectiva e bit a '1', detectou-se ocorrência.

5.2.4 Comunicação exterior (via Módulo de Comunicações)

O Módulo Principal questiona, periodicamente, o Módulo de Comunicações sobre a existência ou não de informação exterior a transmitir (comando → COMM_REQ (Communication Request)). Mantém-se aqui também o modelo de comunicação por *polling*, actuando o Módulo Principal como *master*.

As mensagens de resposta enviadas pelo Módulo de Comunicações devem sinalizar a recepção de mensagem do exterior com o comando COMM_YES (*Communication Yes*). A mensagem vinda do exterior constitui um comando, para pedir informação ou para efectuar uma determinada acção, o qual é transmitido no campo de dados da mensagem COMM_YES.

Após cumprimento do comando recebido é enviada nova mensagem de volta ao Módulo de Comunicações, contendo a informação pedida ou a confirmação de cumprimento do comando.

5.2.5 Modelo de Comunicação

No modelo de comunicação apresentado um módulo actua como *master* da comunicação, havendo apenas troca de mensagens quando este toma a iniciativa de iniciar a mesma. Todas as mensagens trocadas implicam um *handshaking*, ou seja, o Módulo Principal (*master*), após ter enviado uma mensagem, espera por uma resposta ou uma confirmação (*acknowledge*) da recepção da mesma por parte do outro módulo interveniente na comunicação. Consoante o comando enviado constitui um pedido de informação ou um comando de actuação o modelo de comunicação é equivalente ao modelo apresentado na Figura 16 ou na Figura 17, respectivamente.

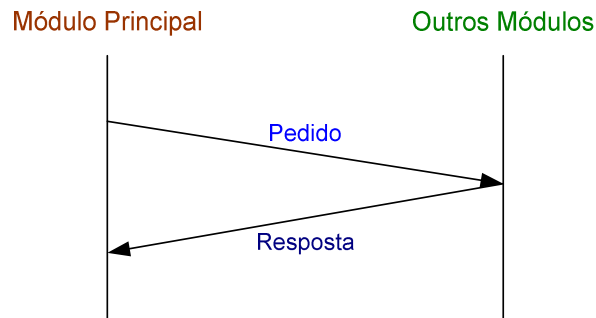


Figura 16 – Modelo de comunicação para um pedido de informação

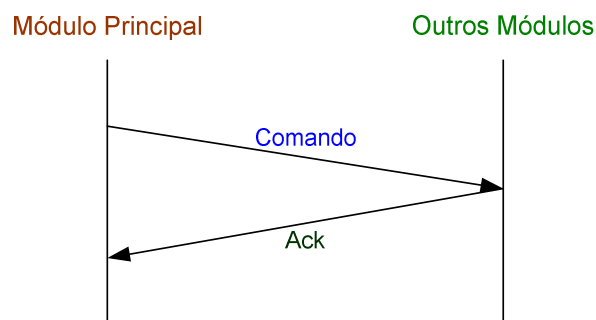


Figura 17 – Modelo de comunicação para a transmissão de um comando de actuação

Os esquemas apresentados evidenciam o papel de *master* que o Módulo Principal assume em todas as comunicações e o *handshaking* sempre presente.

5.3 Relações de Precedência e Limitações da Comunicação

Cada um dos módulos I/O pode receber, em qualquer altura do seu funcionamento, qualquer mensagem com um dos comandos que lhe pode ser dirigido. Relativamente ao Módulo de Comunicações acontece o mesmo, com a excepção das mensagens de configuração, as quais devem ser precedidas de uma mensagem de pedido para iniciar a configuração (comando → CFG_REQ).

A lista dos comandos que podem ser recebidos por cada um dos módulos é apresentada nos anexos C e D, os quais se referem à comunicação entre os módulos e à comunicação entre a ULM e um PC, respectivamente.

6 Unidade Local de Monitorização – Software

6.1 Introdução

Todo o *hardware* já descrito assenta numa arquitectura distribuída, sendo cada módulo controlado por um microcontrolador da família PIC18Fx58 da *Microchip*, tal como já foi referido na secção do *Hardware*, capítulo 4. O código desenvolvido para o microcontrolador foi escrito em C.

Nesta secção da dissertação é apresentada uma descrição sucinta dos algoritmos implementados. Esta descrição não apresenta detalhes uma vez a responsabilidade desta parte do trabalho desenvolvido esteve entregue à segunda pessoa do grupo de trabalho.

6.2 Módulo Principal

6.2.1 Algoritmo geral

A execução das várias tarefas a cargo deste módulo obedece à máquina de estados apresentada na Figura 18.

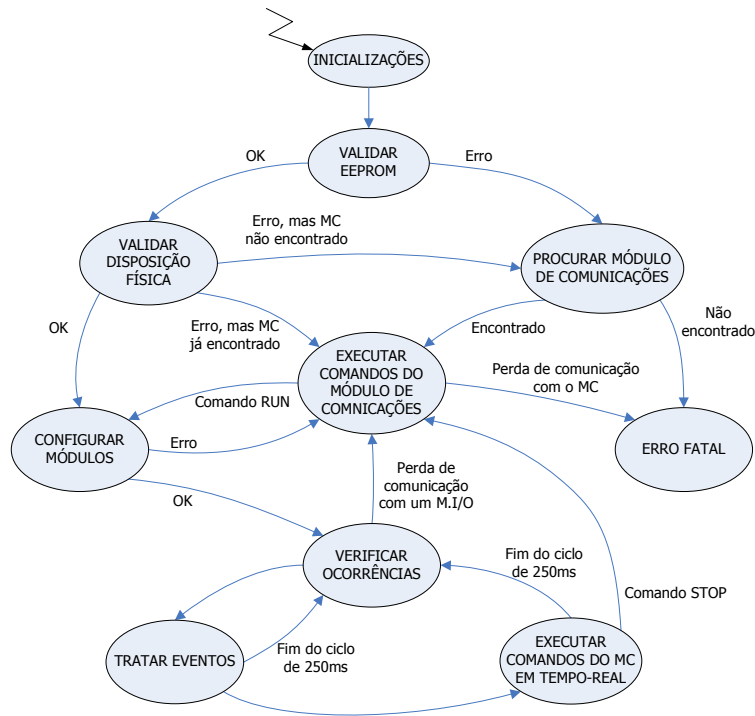


Figura 18 – Diagrama de estados do software do módulo principal

A fase de arranque do módulo principal corresponde à inicialização dos vários periféricos tanto internos como externos ao microcontrolador: inicialização dos portos, módulo da USART, módulo CAN, leitor de *iButtons*, EEPROM, *tick* do microcontrolador e das variáveis utilizadas (Figura 19).

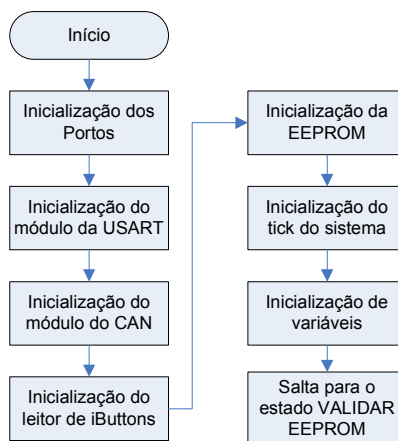


Figura 19 – Fluxograma do software de arranque do módulo principal

A seguir às inicializações é validado o conteúdo da EEPROM. O esquema de mapeamento dos dados na EEPROM foi já apresentado no ponto 3.3.3 desta dissertação.

Os primeiros 4 bytes da memória constituem a *keyword*, usada para detectar se a memória foi ou não previamente inicializada pelo Módulo Principal.

O segmento de configuração é de dimensão fixa e contém a configuração de todos os módulos da ULM, sendo seguido do respectivo *checksum* para validação da consistência dos dados. Se os dados armazenados estiverem corrompidos poderão dar origem a um mal funcionamento de todo o sistema, uma vez que a configuração fica comprometida e não é possível garantir a integridade dos dados guardados no histórico.

O segmento de eventos é de dimensão variável, de forma a maximizar o espaço disponível para o histórico (o número total de eventos pode ascender a 240). Para delimitar este segmento, existe, no início, um byte que indica o número de eventos configurados. O referido segmento de memória é seguido do respectivo *checksum*.

O último segmento compreende o histórico, sendo cada entrada do mesmo acompanhada de um *checksum* de 1byte.

A validação do conteúdo da EEPROM é feita seguindo o fluxograma apresentado na Figura 20.

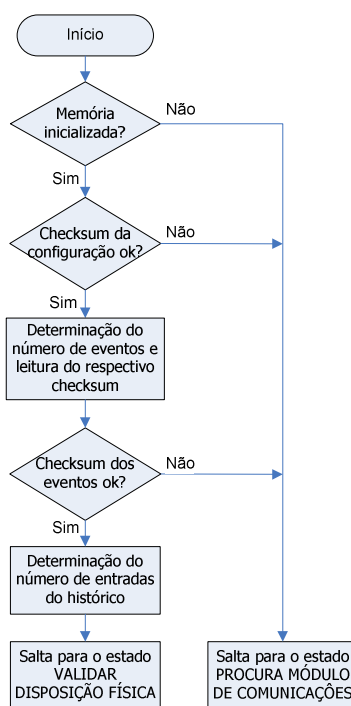


Figura 20 – Fluxograma do estado VALIDAR EEPROM

No caso de se verificar alguma incoerência da informação contida na EEPROM (*keyword* ou *checksums* inválidos), a ULM terá de aguardar a intervenção de um técnico. Apesar da monitorização estar comprometida a ULM poderá receber comandos do Módulo de Comunicações

se conseguir comunicar com ele. Deste modo o Módulo Principal procura determinar a posição do Módulo de Comunicações para estabelecer comunicação com este.

Caso se tenha verificado que a informação contida na EEPROM é válida, o estado de funcionamento passa para a VALIDAÇÃO DA DISPOSIÇÃO FÍSICA dos restantes módulos.

A disposição física dos módulos é, inicialmente, comparada com a disposição física existente em memória, no segmento da configuração. Na Figura 21 é apresentado o fluxograma deste estado.

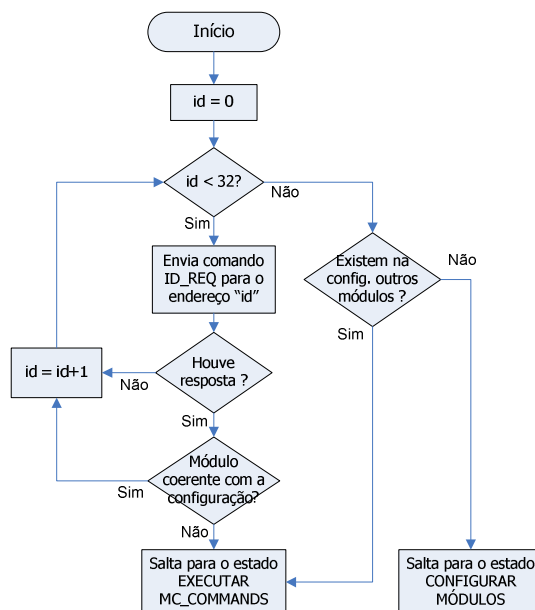


Figura 21 – Fluxograma do estado VALIDAR DISPOSIÇÃO FÍSICA

É enviado o comando ID_REQ (ID *Request*) para cada um dos IDs da ULM (0 a 31) para determinar quais os módulos inseridos no sistema. A resposta recebida é correlacionada com a informação presente em memória, no sector da configuração. Deste modo são detectadas alterações físicas no sistema ou avarias e, no caso de existir alguma incoerência, o módulo principal fica a aguardar ordens do módulo de comunicações no estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES (comunicação com o exterior).

No estado de CONFIGURAÇÃO DOS MÓDULOS a configuração presente em memória (EEPROM) é enviada cada um dos módulos para reconfiguração. Nesta altura a informação contida na EEPROM já foi validada, bem como a disposição física dos módulos, pelo que neste estado apenas podem ocorrer erros decorrentes de falha de comunicação com algum dos módulos. Caso esta falha de comunicação não seja com o Módulo de Comunicações, é reportado um alarme de erro para o Centro de Supervisão e o Módulo Principal fica a aguardar intervenção de um técnico (estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES). Caso a falha de comunicação seja com o Módulo de Comunicações, o estado seguinte é o estado ERRO FATAL (descrito

posteriormente nesta dissertação). Caso toda a configuração decorra conforme esperado, com sucesso, passa para o estado VERIFICAR OCORRÊNCIAS.

O estado VERIFICAR OCORRÊNCIAS, descrito pelo fluxograma da Figura 22, pode ser dividido em 3 fases.

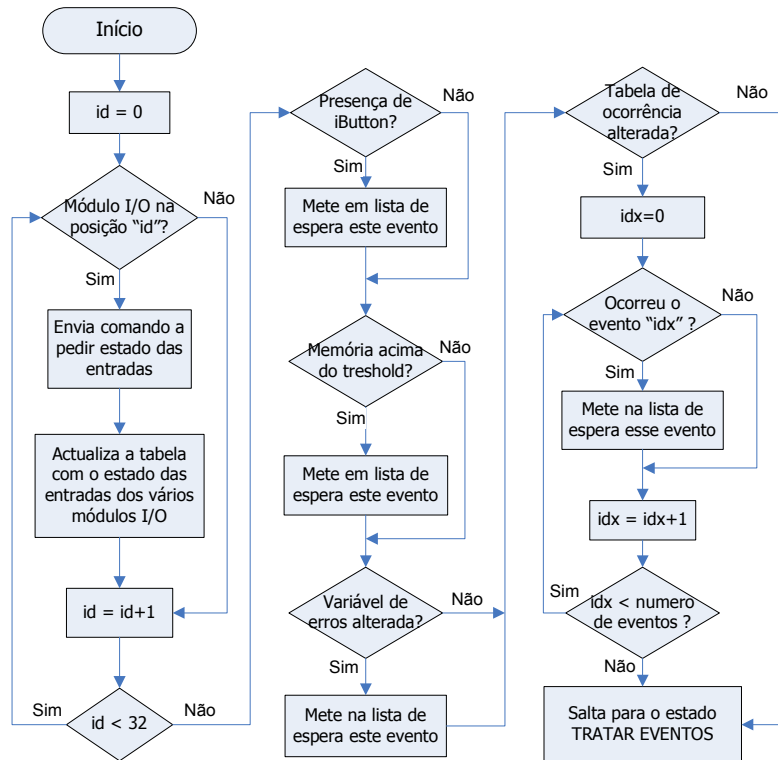


Figura 22 – Fluxograma do estado VERIFICAR OCORRÊNCIAS

A primeira fase consiste na actualização de uma tabela com o estado das ocorrências de todos os módulos I/O. Esta tabela é actualizada apenas neste estado a cada 250ms (período do ciclo de amostragem do sistema). A tabela contém 32 entradas, cada uma com 2 bytes: o primeiro byte é usado para indicar a existência ou não de um módulo nessa posição; o segundo para representar as ocorrências desse mesmo módulo (e.g., bit 0 – ocorrência associada ao porto 0; 8 portos de entrada ao todo; o estado do bit a '1' ou a '0' corresponde a ocorrência activa ou inactiva, respectivamente) (ver Figura 23).

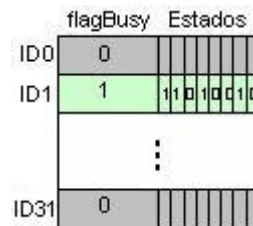


Figura 23 – Representação da tabela de ocorrências

Se a ULM for composta apenas por um módulo I/O, a tabela fica só com uma entrada preenchida, conduzindo a uma perda significativa de eficiência relativamente à ocupação de memória. Por outro lado, o acesso a uma entrada referente a um determinado módulo I/O é feito directamente através do ID como índice da tabela. Deste modo é garantida uma máxima eficiência dos tempos de acesso, uma vez que não é necessário efectuar nenhum tipo de pesquisa.

A segunda fase consiste em verificar outros 3 tipos de ocorrências: uma relacionada com a chegada ou saída de um técnico de assistência, ou seja, detecção do iButton; outra com a verificação da percentagem de ocupação do histórico, se já atingiu o valor pré-configurado pelo técnico; e outra relacionada com a ocorrência de erros que não provocam a paragem do sistema, mas que geram um evento (e.g., memória completamente cheia). Nesta fase, caso alguma das referidas ocorrências esteja activa, é colocado o respectivo evento numa lista de espera para ser tratado no estado seguinte (estas 3 ocorrências têm associados eventos com ID fixo: 241, 242 e 243, respectivamente).

Finalmente, a terceira fase, que consiste na validação dos eventos que são combinação de uma ou mais ocorrências, só é executada se o estado das ocorrências de algum dos módulos I/O foi alterado. Cada evento é lido da EEPROM e é validada a combinação lógica das ocorrências que o compõem.

O conteúdo de cada evento é descrito na Figura 24.

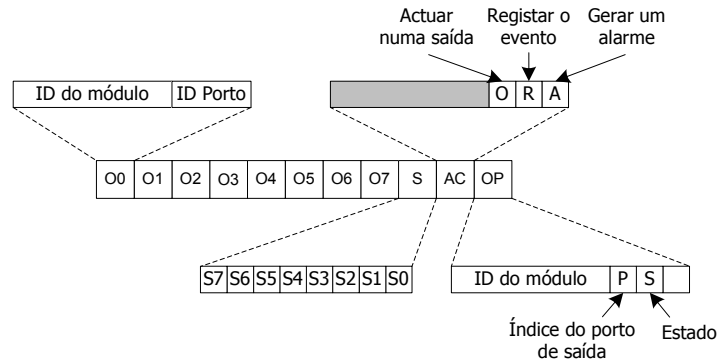


Figura 24 – Descrição da configuração de um evento

A validação de um evento ocorre quando a seguinte equação booleana se verificar:

$$\overline{(S_{O0} \oplus S0)} \cdot \overline{(S_{O1} \oplus S1)} \cdot \dots \cdot \overline{(S_{O7} \oplus S7)} = 1$$

S_{On} corresponde ao estado da ocorrência do porto endereçado por O_n . Os bits S_n correspondem ao estado da respectiva ocorrência para que o evento seja validado (e.g., $S7=0$ significa que a ocorrência associada ao porto $O7$ contribui para a validação do evento apenas se não estiver activa).

O campo representado por AC na Figura 24 indica os tipos de acções a tomar, actuar numa saída, registar o evento ou gerar um alarme; o campo representado por OP (*output port*) está relacionado com a actuação num porto de saída (este campo contém informação válida apenas se o bit 2 do campo AC estiver activo).

Se algum evento se verificar válido é colocado na lista de espera, tal como os três eventos especiais já referidos.

A lista de espera surge, neste estado, para que seja possível cumprir o tempo do ciclo de amostragem de 250ms. A especificação do sistema define que este permite detectar ocorrências com duração mínima de 250ms, o que obriga a que o tempo do ciclo e amostragem do sistema seja igual ou inferior a 250ms. Adicionalmente, cada evento pode gerar alarme, ser registado em memória e/ou actuar numa saída. Se for considerado o pior caso, em que podem ocorrer 240 eventos (máximo especificado), cada um teria de ser processado em cerca de 1ms para que se cumprissem os 250ms de tempo de ciclo. Este tempo de processamento não é possível de cumprir, quer na possibilidade de registo do evento em memória (a escrita de uma só entrada no histórico pode demorar até 5ms, tempo de escrita de uma página na EEPROM) quer na possibilidade de enviar um alarme para o centro de supervisão (estabelecimento de chamada e comunicação). Qualquer tipo de evento que ocorra é colocado na lista de espera e só é realmente processado quando houver tempo de ciclo disponível. Desta forma não há atraso no processamento de nenhum ciclo e não é perdido nenhum evento.

A lista de espera de eventos tem uma estrutura composta por um *buffer* circular de 256 bytes, dois ponteiros (escrita e leitura), duas variáveis usadas para contar o número de eventos e o número de bytes usados e uma variável para armazenar a hora completa do primeiro evento da lista (Figura 25).

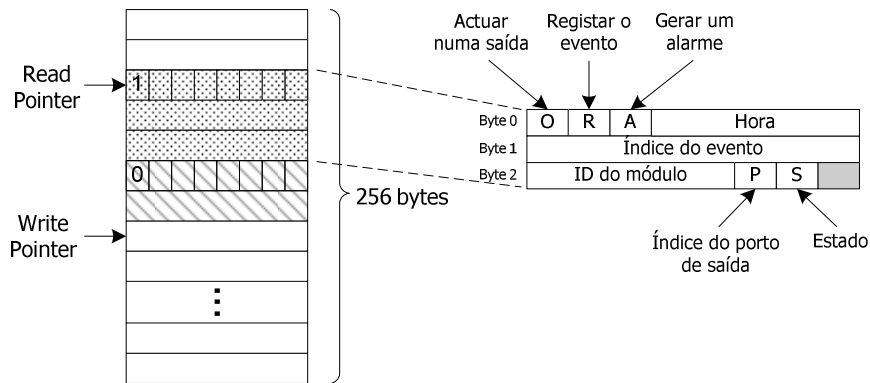


Figura 25 – Representação do *buffer* circular e de uma entrada da lista de espera

O formato usado para representar a data/hora de ocorrência de um evento ocupa 4 bytes (tempo em segundos relativo a uma data fixa predeterminada). Essa informação é guardada apenas para o primeiro evento da lista e os restantes possuem apenas um *offset* relativo à hora de ocorrência do primeiro.

Foram reservados 5 bits para *offset*, o que equivale a 31 segundos de diferença máxima entre o primeiro evento e o último. Este valor corresponde a mais de 100 ciclos de amostragem do sistema, pelo que foi considerado suficiente para o tratamento dos eventos presentes no ciclo de espera (gerar alarme, registar evento na memória e/ou actuar numa saída).

No estado TRATAR EVENTOS efectua-se o processamento dos eventos que se encontram na lista de espera, com a restrição de não ser ultrapassado o tempo de ciclo de amostragem do sistema (ver Figura 26).

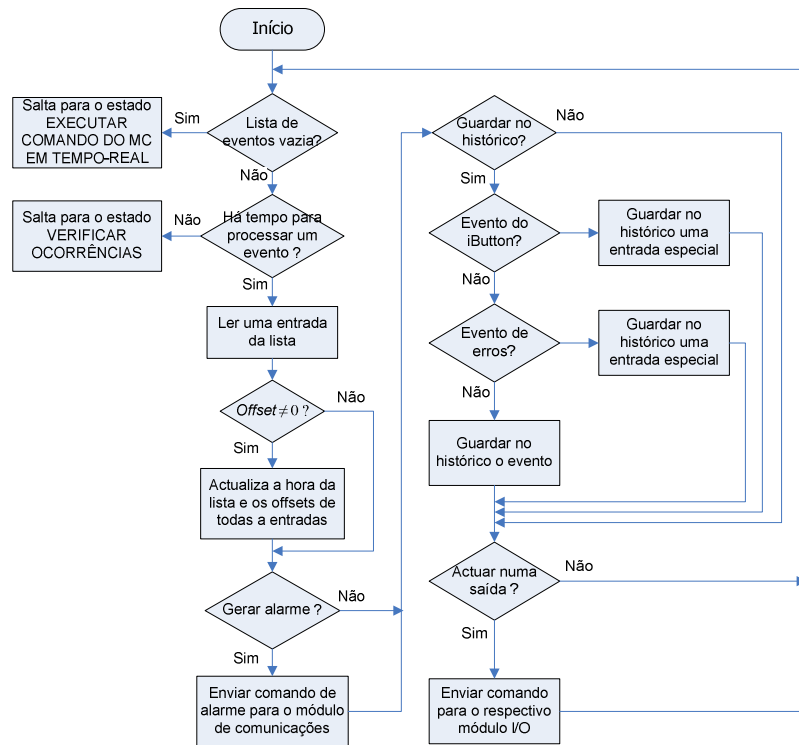


Figura 26 – Fluxograma do estado TRATAR EVENTOS

Na fase de processamento, após verificar se existem eventos a processar, verifica-se se há tempo para processar um evento sem esgotar o tempo do ciclo de amostragem. Se houver tempo suficiente, verifica-se se o *offset* relativo da hora desse evento é diferente de zero. Se tal se verificar a hora absoluta guardada na estrutura da lista não corresponde à do primeiro evento da lista (evento lido). Essa hora é então actualizada, tal como os *offsets* dos restantes eventos. O processo de actualização dos *offsets* é fundamental para evitar o *overflow* do campo correspondente.

Quando é testado se o evento deve ser guardado no histórico, os eventos associados à entrada/saída de um técnico e o evento de erros são distinguidos dos restantes, associados às ocorrências. Esta distinção está relacionada com o facto da informação a guardar no histórico variar consoante o tipo de evento. Na Figura 27 são apresentados os formatos das entradas para os três tipos de eventos.

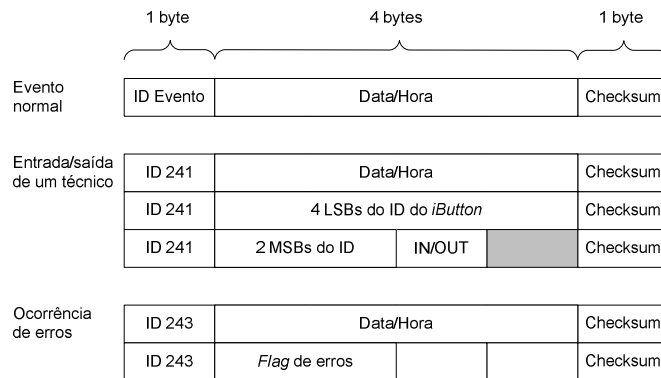


Figura 27 – Estrutura das entradas do histórico de um evento normal e das duas exceções

Quando se inicia um processo de assistência técnica é usado um dispositivo de identificação (*iButton*) que possui um ID de 6 *bytes*, havendo necessidade de guardar essa informação. Aquando da ocorrência de erros há, do mesmo modo, necessidade em armazenar a *flag* dos erros, 2 *bytes*. Estes eventos ocupam mais do que uma entrada do histórico para armazenar a informação adicional, mas sem alterar o formato de cada entrada individual, ou seja, cada entrada adicional contém igualmente o ID do evento bem como o *checksum* no último *byte*, mas no lugar da data/hora é guardado outro tipo de informação. Desta forma, de cada vez que é lida uma entrada do histórico, quer seja na validação ou no *download* de informação, não há leituras de exceção, o formato mantém-se qualquer que seja o evento guardado.

Durante a execução deste estado, caso o tempo de ciclo chegue ao fim, retorna-se ao estado anterior de modo a verificar novas ocorrências. Caso após tratar todos os eventos o tempo de ciclo não se tenha esgotado, passa-se para o estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES EM TEMPO-REAL.

Neste último estado são atendidos comandos externos à ULM (vindos do módulo de comunicações). É permitido responder, no máximo, a um comando do módulo de comunicações por ciclo, a menos da exceção para o comando de *upload* do histórico, sendo este atendido num ou mais ciclos de amostragem (o histórico encontra-se na EEPROM, o que implica um *overhead* significativo associado à leitura das várias entradas, o qual é tanto maior quanto maior o número de entradas do histórico).

O fluxograma apresentado na Figura 28 descreve de uma forma mais detalhada o funcionamento deste estado.

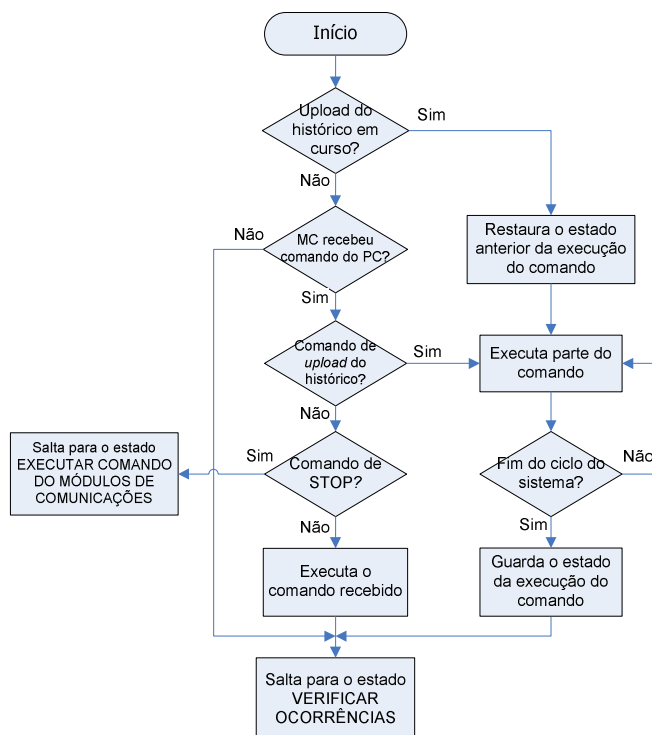


Figura 28 – Fluxograma do estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES EM TEMPO-REAL

A lista de comandos que pode ser executada neste estado (comandos que não implicam a paragem da amostragem do sistema por parte do módulo principal) é apresentada de seguida:

- Leitura da *flag* de erros;
- Limpeza da *flag* de erros;
- Obtenção da configuração física da ULM (disposição dos vários módulos);
- Leitura da hora actual da ULM;
- Mudança da hora da ULM;
- Actuação num porto de saída;
- Leitura do estado do histórico: número total de entradas e número de entradas ocupadas;
- *Download* do histórico;
- Leitura da data da configuração presente na ULM;
- Paragem da ULM (para permitir a execução de comandos de configuração).

O estado Executar COMANDOS DO MÓDULO DE COMUNICAÇÕES é em tudo semelhante ao estado descrito imediatamente antes, mas com a diferença de ser um estado bloqueante, ou seja, a ULM encontra-se dedicada à execução de comandos externos (vindos do PC).

Existem duas situações distintas que determinam a execução deste estado: ou foi executado o comando de paragem (através de uma intervenção humana), ou ocorreu algum erro que impossibilita o normal funcionamento do sistema, ficando a ULM à espera da intervenção de um técnico. Neste último caso, é gerado um alarme para a central e a *flag* de erros é guardada no histórico.

Na Figura 29 é apresentado o fluxograma que descreve o funcionamento deste estado.

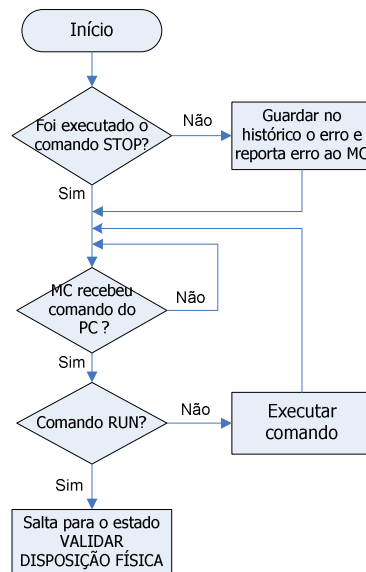


Figura 29 – Fluxograma do estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES

Neste estado, além dos comandos já descritos como possíveis para o estado anterior, é possível executar mais os seguintes comandos:

- Actualização da configuração dos módulos da ULM;
- Actualização da configuração dos eventos;
- Colocar de novo a ULM em funcionamento (reconfigura os módulos e dá início ao normal funcionamento de todo o sistema).

O protocolo de comunicação entre módulos é do tipo *master-slave*, estando a cargo do Módulo Principal a tarefa de controlar todas as comunicações. Uma vez que cada módulo pode ser inserido em qualquer posição da RACK, numa fase inicial os vários módulos desconhecem o ID do Módulo Principal e o Módulo Principal desconhece o ID dos restantes módulos inseridos no sistema (o ID corresponde à posição do módulo na RACK, de 0 a 31, e é detectado pelo próprio módulo). Quando ocorre algum erro na fase de arranque do sistema, o Módulo Principal fica a aguardar a intervenção de um técnico, local ou remotamente. A comunicação com o exterior é efectuada através do Módulo de Comunicações. No entanto, esta comunicação só é possível após o Módulo

Principal efectuar a procura do Módulo de Comunicações, informando-o relativamente ao seu ID e obtendo o ID deste último (estado PROCURAR O MÓDULO DE COMUNICAÇÕES).

Neste estado são varridos todos os IDs até que seja encontrado o Módulo de Comunicações. Tendo sido correctamente detectado esse módulo passa-se para o estado EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES. Caso o Módulo de Comunicações não seja encontrado, o estado seguinte é o estado ERRO FATAL.

O estado ERRO FATAL corresponde a um estado em o Módulo Principal não consegue comunicar com o exterior por falha de comunicação com o Módulo de Comunicações ou ausência do mesmo. Este estado é sinalizado com o LED multicolor a piscar vermelho à frequência de 2Hz.

6.2.2 Detalhes de implementação

A EEPROM utilizada está internamente organizada em páginas de 32 bytes cada. Cada escrita numa página demora aproximadamente 5ms. Uma vez que o módulo principal tem de executar uma enorme quantidade de tarefas por ciclo de amostragem (período de 250ms), utilizar 5ms por cada escrita na EEPROM revela-se uma perda de eficiência muito grande. Para minimizar esse tempo, foi criada uma cache de escrita para a EEPROM: a informação a gravar na EEPROM é guardada numa primeira fase em *buffers*, e uma rotina de interrupção activada por um *timer* com período de 5ms é responsável pelo desencadear do processo de escrita efectiva da informação, presente nos *buffers*, na EEPROM.

O microcontrolador usado já tem integrado um módulo CAN, tendo este a particularidade de gerar interrupções de recepção e fim de transmissão. Este facto permitiu o desenvolvimento de uma camada de *software* capaz de gerir tanto a recepção como a transmissão de tramas CAN usando interrupções (ver referência [33]).

O uso de interrupções torna o código de transmissão de mensagens não bloqueante, e no caso da recepção, liberta a aplicação da responsabilidade de estar constantemente a verificar se recebeu alguma mensagem, para garantir que não ocorre *overflow* nos *buffers* de recepção (*hardware*).

A interface com o iButton foi feita implementado em *software* o protocolo de comunicação descrito no *datasheet* do próprio componente (ver referência [34]).

6.3 Módulo de Comunicações

6.3.1 Algoritmo geral

O Módulo de Comunicações actua como *gateway* entre o Módulo Principal e o Centro de Supervisão Remoto (CSR) ou um PC local. A interface com um PC local é feita através de um controlador USB, enquanto que a ligação ao CSR é feita através de um *modem* externo (rede fixa ou GSM) ligado a este módulo por uma interface RS232.

O funcionamento deste módulo tem que permitir conciliar a comunicação a ritmo elevado com o módulo principal com a comunicação a ritmo mais lento com o *modem*. A solução encontrada passa por tratar os comandos de mais alta prioridade vindos do Módulo Principal (mensagens CAN) na rotina de interrupção. Desta forma é garantido que esses comandos são tratados atempadamente e, uma vez que não exigem qualquer tipo de processamento, não sobrecarregam a rotina de interrupção.

O fluxograma que descreve a rotina de interrupção associada à recepção de tramas CAN é apresentado na Figura 30.

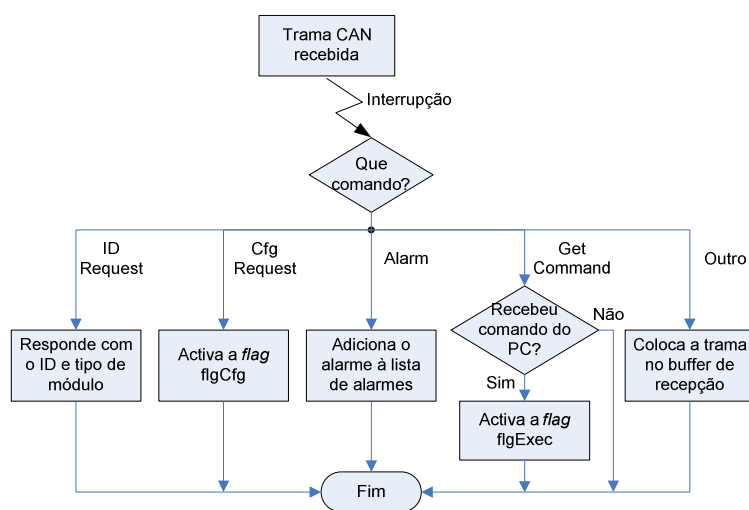


Figura 30 – Fluxograma da rotina de interrupção associada à recepção de tramas CAN

São 4 os comandos processados na rotina de interrupção, mas apenas 2 deles são executados na íntegra dentro dessa mesma rotina. Do diagrama da Figura 30 é possível verificar que o tratamento de 2 dos comandos corresponde à activação de 2 *flags*. Estas *flags* são usadas como sinalização para informar o código executado na rotina principal de quais os comandos que foram recebidos.

Na Figura 31 é apresentado o fluxograma do algoritmo de controlo deste módulo, executado na rotina principal.

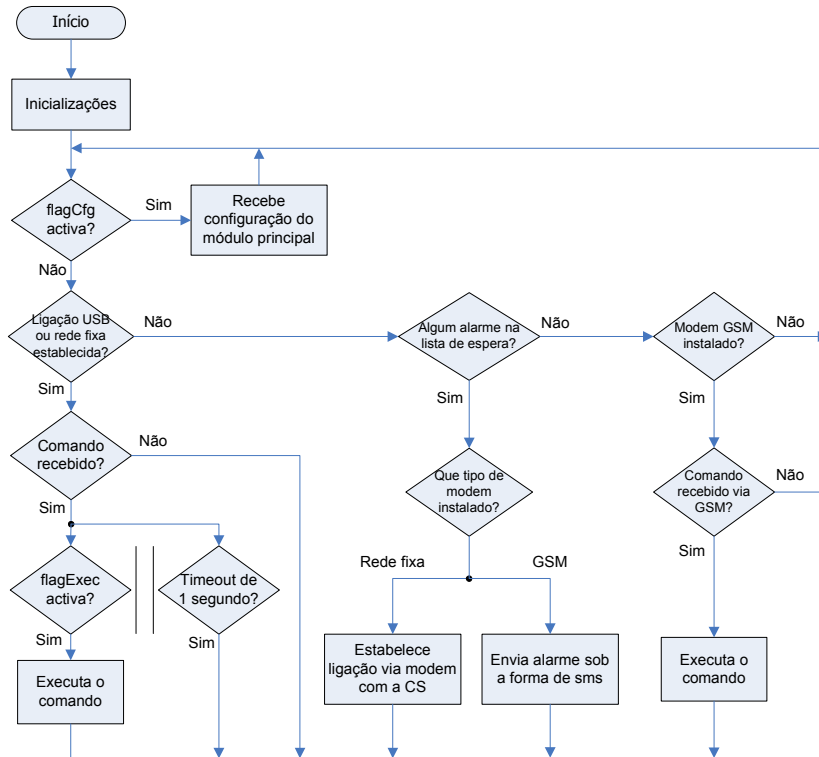


Figura 31 – Fluxograma do algoritmo de controlo do módulo de comunicações

O algoritmo consiste essencialmente no tratamento de 3 situações distintas.

A primeira está relacionada com o processamento de comandos vindos do CSR ou PC local, quando está estabelecida uma ligação com um desses sistemas. Sempre que é recebido um comando externo à ULM, o Módulo de Comunicações aguarda a recepção do comando *GetCommand* vindo do Módulo Principal o qual envia, como resposta, ao dispositivo externo. Quando o Módulo Principal responde, indicando que se encontra preparado para processar o comando, a *flagExec* é activada. Desde a recepção do comando *GetCommand* até à activação da *flagExec* tudo é processado na rotina de interrupção. Caso o MP não responda durante 1 segundo, o comando não é executado.

A segunda situação está relacionada com o envio de alarmes ao CSR. Este processamento depende do tipo de modem que está ligado à ULM. Caso esteja ligado um modem de rede fixa, é estabelecida uma ligação com o CSR, sendo este responsável por verificar a existência de alarmes através da execução do respectivo comando. Caso se trate de um modem GSM, é enviada uma SMS com o alarme ocorrido.

Finalmente, a terceira situação está relacionada com a execução de comandos do CSR recebidos por SMS. Caso se encontre ligado um modem GSM à ULM, é verificada a recepção de SMS's por *polling*, sendo posteriormente feito o processamento do seu conteúdo.

6.3.2 Detalhes de implementação

Para a comunicação via CAN foi usado o mesmo módulo de *software* desenvolvido para o Módulo Principal (ver secção 6.2.2).

A comunicação via USART foi implementada segundo a mesma filosofia usada na comunicação via CAN (ver secção 6.2.2), uma vez que o microcontrolador possui um módulo USART o qual gera interrupções de recepção e fim de transmissão de 1 carácter.

A comunicação via USB com o exterior é feita através de um módulo externo ao microcontrolador. Este módulo encapsula todo o protocolo USB, dispensando a implementação de qualquer tipo de *protocol stack* associado ao USB. O *software* de interface com este dispositivo baseia-se no protocolo de comunicação descrito no respectivo *datasheet* (ver referência [32]).

6.4 Módulo I/O

6.4.1 Algoritmo geral

Os módulos I/O são responsáveis pela validação das ocorrências associadas aos respectivos portos de entrada e pela actuação nas saídas. O fluxograma apresentado na Figura 32 descreve o código desenvolvido para este módulo.

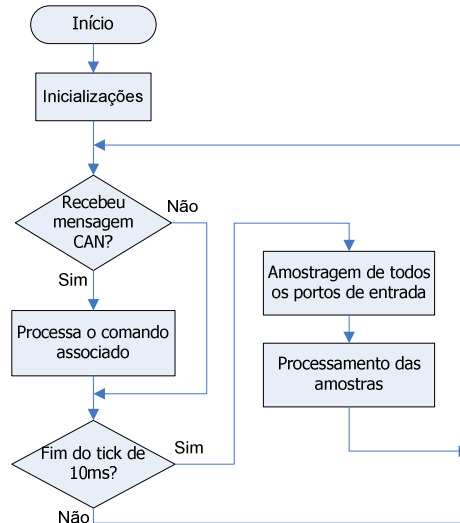


Figura 32 – Fluxograma do Módulo I/O

No fluxograma da Figura 32, é possível verificar que os processos relacionados com a amostragem e processamento dos sinais de entrada são executados com um período de 10ms, e que o tempo que resta é usado para executar comandos do módulo principal, caso tenham sido recebidos. Um problema inerente ao algoritmo adoptado tem a ver com o possível *jitter* provocado pelo processamento dos comandos vindos do módulo principal, reflectindo-se num atraso na execução dos restantes processos. No entanto, o valor desse *jitter*, quando existe, é praticamente insignificante. Os comandos que têm de ser processados durante o normal funcionamento do módulo implicam a recepção de uma trama CAN, construção e envio de uma trama de resposta. Uma vez que tanto o envio como a recepção de tramas CAN estão implementados usando interrupções, o tempo total de processamento de cada comando pode ser considerado desprezável (da ordem de poucos μ s). Adicionalmente, como o tempo de amostragem e processamento dos sinais de entrada é inferior a 1ms, sobram praticamente 10ms para executar comandos provenientes do Módulo Principal. Deste modo é possível concluir que só vai existir *jitter* quando é recebido um comando no instante imediatamente anterior à activação do *tick*. Nesse caso o *jitter* vai corresponder exactamente ao tempo de processamento do respectivo comando, o qual, tal como já foi referido, é pequeno e não tem impacto significativo no período de processamento dos sinais de entrada.

6.4.2 Algoritmo de amostragem dos portos de entrada

Uma ocorrência corresponde à detecção de um estado lógico ‘1’ ou ‘0’ com duração temporal mínima de 250ms. Uma vez que os sinais de entrada podem estar ligados a relés ou a outros dispositivos de comutação com actuação mecânica, existe um ruído de *bouncing* associado aos

contactos. A solução adoptada para efectuar a filtragem do ruído de *bouncing* consiste em amostrar as entradas a uma frequência de 100Hz, sendo a validação de cada estado feita apenas quando 25 amostras consecutivas assumem o mesmo valor (período de amostragem de 10ms → 10ms x 25 = 250ms). Na Figura 33 é apresentado o fluxograma do algoritmo adoptado.

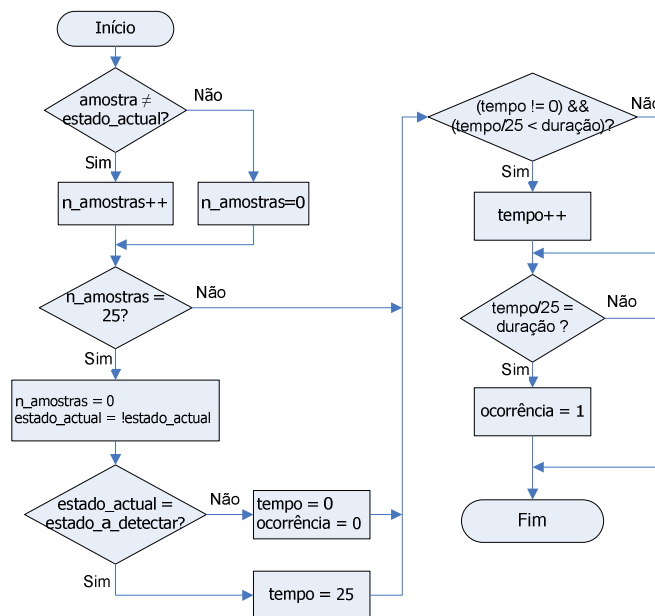


Figura 33 – Fluxograma do algoritmo de processamento dos sinais de entrada

Na Figura 34 é apresentado um exemplo de um sinal analógico em que é feita uma amostragem a 100Hz (período de 10ms). O sinal resultante da amostragem é apresentado na Figura 35.

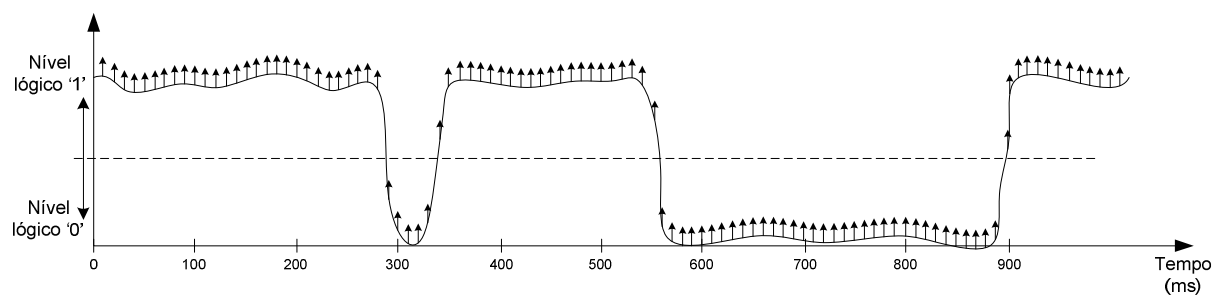


Figura 34 – Exemplo de um sinal à entrada de um dos portos

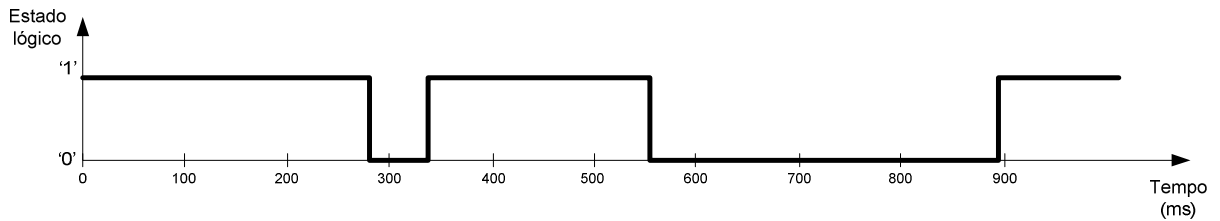


Figura 35 – Sinal resultante da amostragem do sinal apresentado na Figura 34

Se for aplicado o algoritmo descrito no fluxograma da Figura 33, o estado detectado pode assumir dois resultados consoante o valor inicial da variável estado_actual. Considerem-se as Figura 36 e Figura 37.

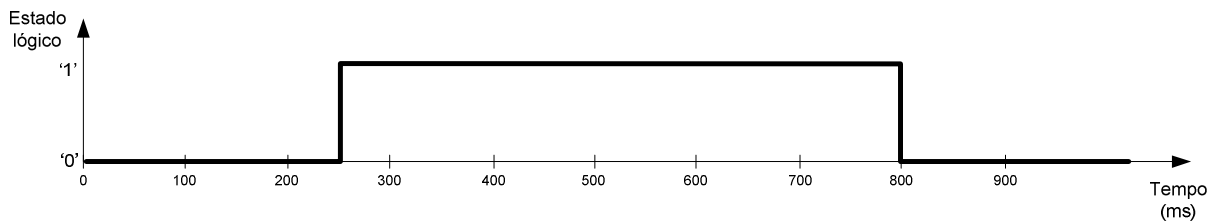


Figura 36 – detectado com “estado inicial=0”

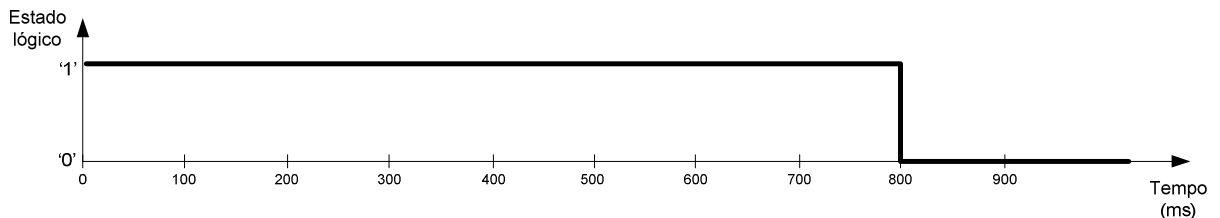


Figura 37 – Estado detectado com “estado inicial=1”

Se o estado_actual no instante anterior a 0ms era '0', só após 25 amostras a '1' é que o estado detectado é '1'. Por outro lado, se o estado_actual no instante anterior a 0ms era '1' o estado detectado mantém-se no valor '1' até ao aparecimento de 25 amostras consecutivas com o valor '0'. O estado inicial do porto é configurável.

Só existe mudança de estado quando o sinal amostrado se mantém estável durante 250ms. Este procedimento permite eliminar o ruído de *bouncing* associado aos contactos mecânicos.

O estado detectado não corresponde ainda à detecção da ocorrência associada ao porto. A ocorrência, com duração parametrizável de 250ms a 1min, é activada quando a variável estado detectado permanece com o valor pretendido durante o período de tempo pré configurado.

7 Protocolo de Comunicação entre PC e ULM

7.1 Introdução

A comunicação entre o PC (do Centro de Supervisão ou de Assistência) e uma ULM é efectuada em variadas situações que vão desde o pedido de informações por parte do PC, à actuação na ULM ou à recepção de alarmes. Toda a comunicação entre o PC e cada ULM implica *handshaking*. O envio de cada mensagem requer, portanto, a posterior recepção de uma mensagem de *Acknowledge* e/ou uma mensagem de resposta se se tratar de um pedido de informação. Neste capítulo é apresentado o protocolo que define a sequência de mensagens trocadas em cada situação.

A sequência de operações a efectuar, embora não sendo rígida, deve cumprir um conjunto mínimo de especificações, sob pena de, no caso de não serem cumpridas, poder comprometer dados relativos a uma determinada ULM ou até o correcto funcionamento da mesma. No final deste capítulo será apresentada uma relação de precedências necessária à salvaguarda do bom funcionamento do sistema e um conjunto de limitações inerentes ao modelo de comunicação.

Durante toda a descrição do protocolo vai-se considerar que se trata de comunicação entre uma ULM e um Centro de Supervisão Remoto (CSR) através de rede fixa comutada e que, portanto, há necessidade de estabelecer ligação. Todo o protocolo é, no entanto, válido também para o caso de se tratar de uma ligação directa de um PC através da interface UBS da ULM.

No final deste capítulo será ainda apresentada uma avaliação da viabilidade da comunicação por GSM e por rede fixa comutada.

7.2 Fluxos de Informação

Cada mensagem trocada entre a ULM e o PC (quer seja do Centro de Supervisão quer seja de Assistência) é constituída por um caracter de início de trama, um comando (um byte), dados de tamanho variável (eventualmente nulo), o *checksum* da mensagem e um caracter de fim de trama. O *checksum* é aplicado ao conjunto (comando + dados). Para garantir que os caracteres de início e fim de trama só surgem nessas situações, toda a trama, excluindo esses caracteres, é codificada em ASCII.

7.2.1 Recepção de alarme

Neste caso o CSR encontra-se à escuta e não está nenhuma ligação estabelecida, permitindo que uma qualquer ULM possa comunicar. O despoletar de um evento numa ULM desencadeia o estabelecimento de uma chamada desta para o CSR. A partir desse momento a ULM aguarda que o CSR solicite o envio da informação relativa ao evento ocorrido. Se por um lado podemos afirmar que o Módulo Principal actua como o *master* da comunicação entre os módulos ao nível de cada ULM, ao nível do conjunto de ULMs e do CSR que as monitoriza/controla, o papel de *master* está reservado ao CSR. Assim, cada ULM só envia a informação relativa ao/aos evento/os ocorridos quando o CSR solicita essa informação.

Quando na ULM x ocorre um evento susceptível de reportar alarme para o CSR, a ULM toma, pois, a iniciativa de estabelecer uma ligação com o mesmo (no caso de este último se encontrar disponível) (Figura 38). O CSR envia uma mensagem pedindo o identificador da ULM que estabeleceu a comunicação (Comando → *Get_ID*). A ULM envia, de seguida, a resposta pedida, a qual contém o comando *Resp* e 14 bytes correspondendo ao número de telefone da ULM. Como cada ULM só toma a iniciativa de estabelecer uma ligação com o CSR quando pretende enviar um alarme, este último, após receber a resposta solicitada anteriormente, pede à ULM o envio da informação relativa ao evento ocorrido (Comando → *Get_Alarm*), sem necessitar de perguntar o porquê da chamada. A ULM envia a resposta, a qual inclui o comando *Resp*, 1 byte contendo o índice do evento ocorrido (permitindo associar a uma determinada descrição e a uma mensagem de alarme) e 4 bytes com a data do evento (registo de tempo relativo a uma data fixa predeterminada, com resolução ao segundo). O CSR envia seguidamente uma mensagem de *Acknowledge* confirmando a recepção da informação pedida. A sequência *Get_Alarm*, *Resp*, *Acknowledge* é repetida até ao envio de uma mensagem *Resp* sem mais informação, indicando que a ULM não tem mais alarmes a enviar.

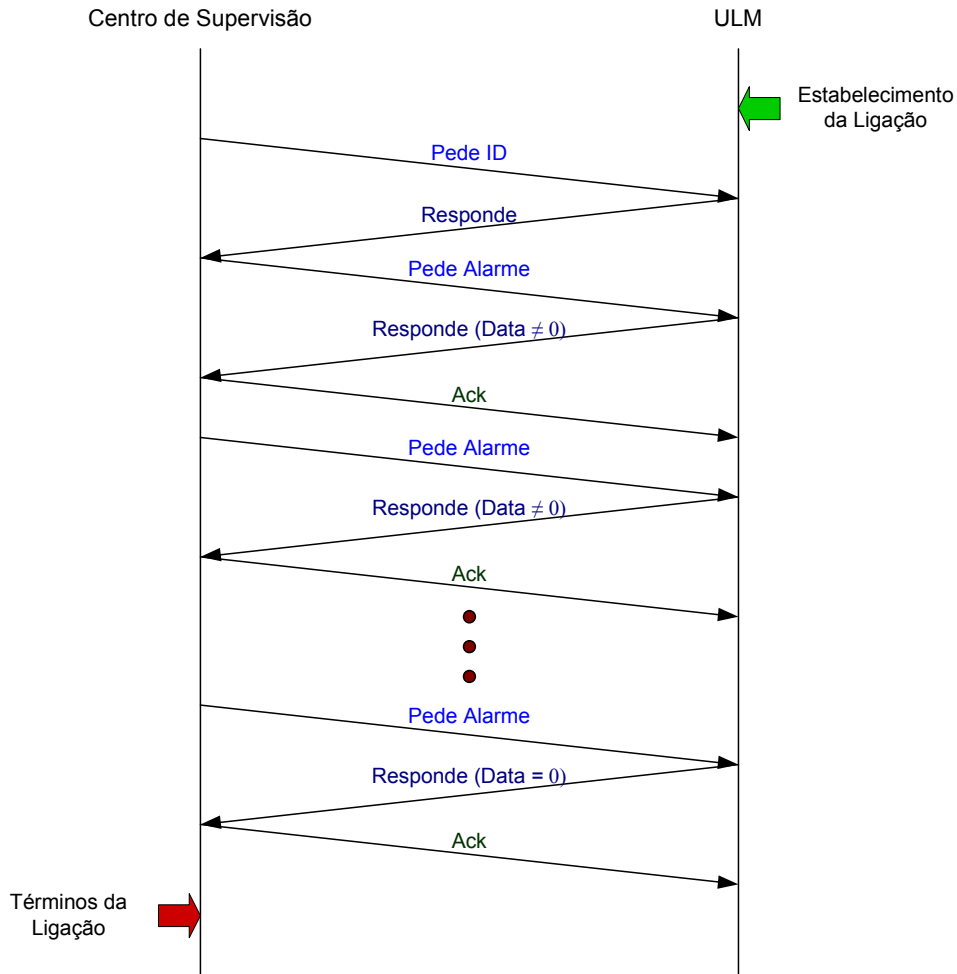


Figura 38 – Sequência de mensagens trocadas aquando da recepção de um alarme

No final desta sequência de mensagens o CSR termina a ligação que estava estabelecida desde o envio da primeira mensagem por parte da ULM. Embora tenha sido a ULM a estabelecer a ligação com o CSR, é este último que tem a responsabilidade de terminar a ligação, mantendo-se como *master* da comunicação.

7.2.2 Estabelecimento de Comunicação com uma nova ULM

O estabelecimento de uma comunicação com uma nova ULM inclui a transferência, para o CSR, da configuração física da ULM. O início da comunicação dá-se com o estabelecimento propriamente dito da chamada, por parte do CSR. Este último envia uma mensagem com o pedido da configuração física da ULM (Comando → *Get_FCFG*). Recebe de seguida uma mensagem de *Acknowledge* e a resposta ao pedido, Figura 39. A segunda mensagem recebida inclui o comando de resposta, *Resp*, e um campo de dados com 8 bytes com informação relativa a cada posição da ULM (2 bits para cada posição), indicando se se encontra algum módulo ligado nessa posição e,

no caso afirmativo, que tipo de módulo (00 → Posição vazia, 01 → Módulo I/O, 10 → Módulo de Comunicações e 11 → Módulo Principal).

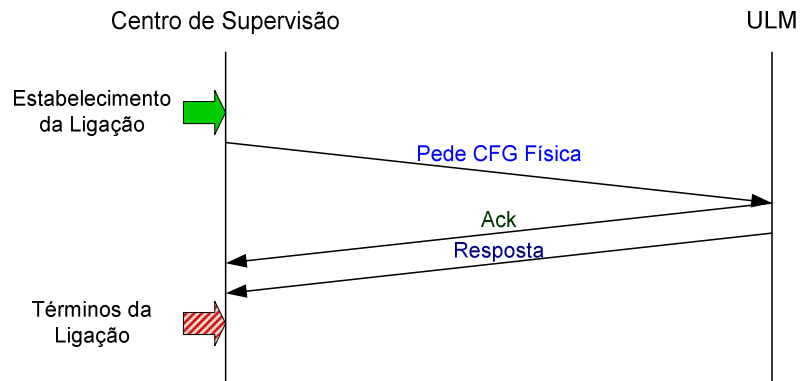


Figura 39 – Sequência de mensagens trocadas aquando da ligação a uma nova ULM

Na generalidade, cada pedido do CSR é seguido de uma mensagem de *Acknowledge* e de uma resposta ao pedido. Isto acontece porque a maior parte da informação susceptível de ser pedida tem que ser fornecida pelo Módulo Principal, o que requer comunicação entre os dois módulos e portanto implica algum *overhead*. Assim, o *Acknowledge* confirma a recepção da mensagem por parte do Módulo de Comunicações e, após a obtenção da informação, o Módulo de Comunicações envia a resposta correspondente para o CSR, Figura 40.

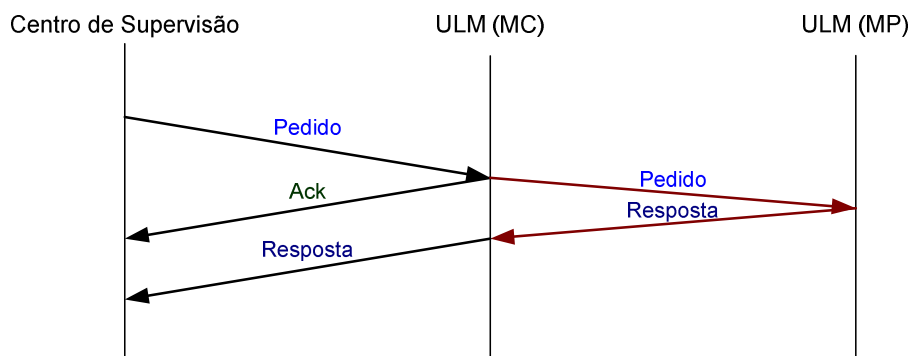


Figura 40 – Sequência de mensagens trocadas quando é efectuado um pedido de informação

O término da chamada após esta comunicação é facultativo. Como se trata de uma nova ULM, vai-se seguir, obrigatoriamente, uma fase de configuração (apenas ao nível do PC) que pode ser mais ou menos demorada, dependendo da dimensão da ULM (a (re)configuração de uma ULM é feita primeiramente ao nível da base de dados do PC e só depois enviada para a ULM

correspondente). Assim, o tempo que medeia entre a troca de mensagens relativa à configuração física e o envio da configuração, pode ser suficientemente grande para que se torne vantajoso terminar a chamada permitindo que a ligação do CSR esteja disponível para a comunicação de alarmes por parte de outras ULMs.

7.2.3 Estabelecimento de Comunicação com uma ULM Existente

O estabelecimento de uma comunicação com uma ULM existente compreende toda a troca de informação preliminar necessária à identificação da ULM e à actualização da informação relativa ao seu estado de funcionamento. Esta comunicação inicia-se com o estabelecimento propriamente dito da chamada, por parte do CSR (Figura 41), o qual envia de seguida a mensagem de pedido do estado da ULM (Comando → *Get_ULM_Status*) para determinar o seu estado de funcionamento e os erros que eventualmente possam ter ocorrido. A mensagem de resposta contém 2 bytes de dados que constituem a *flag* de erros (*Status*). Após reconhecimento da informação de estado é feito o *reset* ao registo correspondente para permitir detectar erros futuros. Seguidamente é enviada uma mensagem de pedido da identificação da ULM (Comando → *Get_ID*). O conteúdo da mensagem de resposta foi já descrito no parágrafo relativo ao envio de um alarme (7.2.1). Segue-se a mensagem de pedido da data da configuração actual (Comando → *Get_CFG_Date*) para confirmar a sincronização da configuração em funcionamento na ULM com a última configuração guardada na base de dados. A resposta a este pedido contém, para além do comando *Resp*, 4 bytes de dados com a data/hora da configuração representada em segundos a partir de uma data fixa predeterminada. Finalmente é enviada a mensagem de pedido de configuração física (Comando → *Get_FCFG*) para reconhecimento das posições dos vários módulos constituintes da ULM e cuja composição da mensagem de resposta já foi descrita na secção 7.2.2. Esta sequência de mensagens é apresentada na Figura 41.

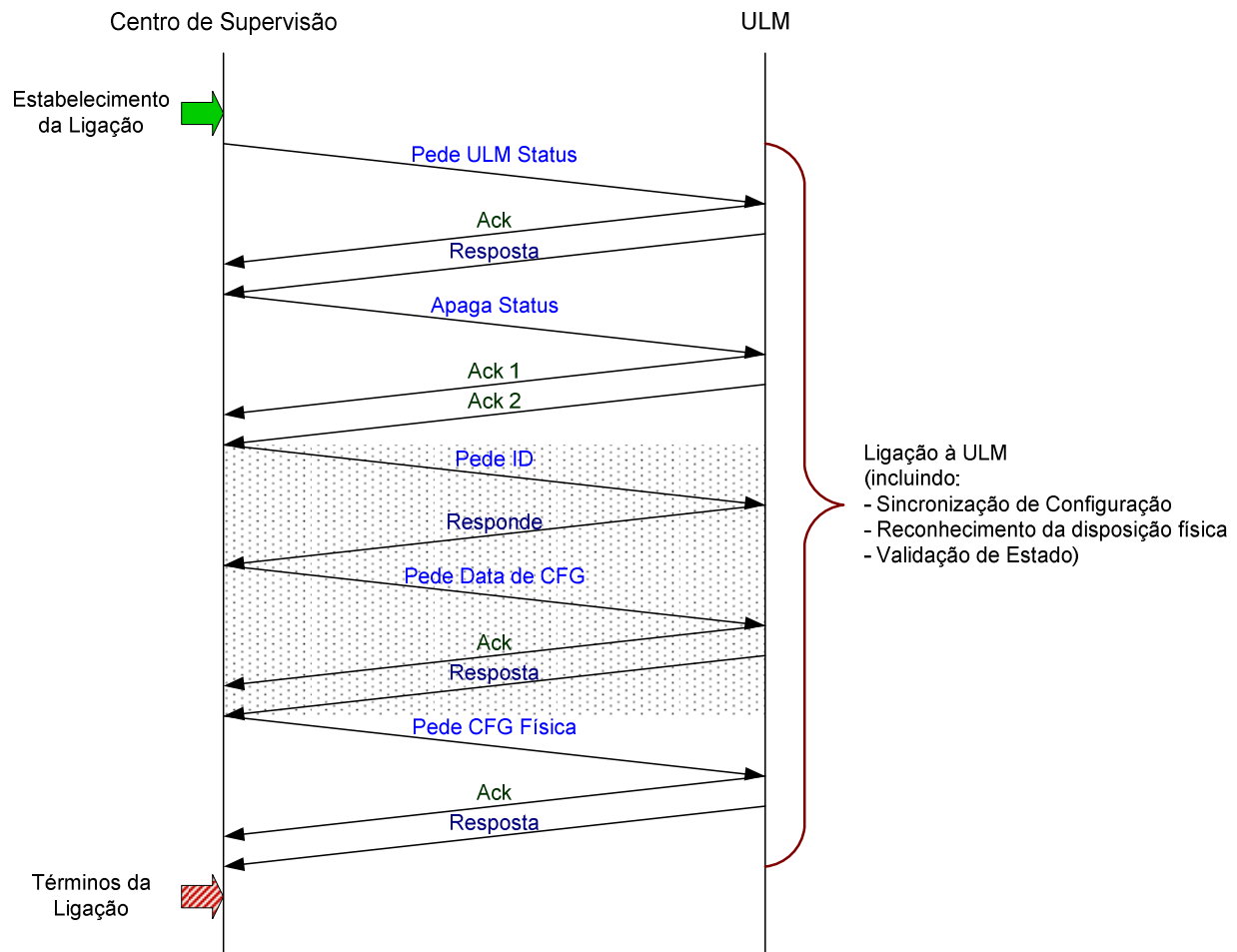


Figura 41 – Sequência de mensagens trocadas aquando da ligação a uma ULM existente

A mensagem, `Get_ID`, pode parecer desnecessária quando a comunicação é feita a partir do CSR, uma vez que o que identifica cada ULM é o seu número de telefone, necessário ao estabelecimento da chamada. No entanto pretende-se que o protocolo de comunicação seja único para PC do CSR e para o PC de Assistência e, quando a ligação ao PC é efectuada localmente através da interface USB, é necessário garantir que a base de dados que se está a manipular corresponde efectivamente à ULM ligada.

As mensagens de pedido de identificação e de pedido da data da configuração (assinaladas a sombreado na Figura 41) são enviadas apenas se o estado da ULM, recebido como resposta ao `Get_ULM_Status`, garantir que a configuração guardada em memória é válida. Caso contrário, a ULM não possui garantias quanto à integridade da informação e não fornece os referidos dados.

O modelo de comunicação anteriormente apresentado, em que um pedido do CSR é seguido da recepção de uma mensagem de *Acknowledge* e de uma resposta, é quebrado, na sequência de mensagens apresentada, relativamente ao pedido de identificação. Isto acontece porque o Módulo

de Comunicações possui o ID da ULM correspondente, o que lhe permite enviar a resposta de imediato, sem necessidade de comunicação com o Módulo Principal.

As mensagens oriundas do CSR, que não constituem um pedido de informação, não recebem uma resposta mas uma confirmação de cumprimento da “ordem” transmitida com sucesso. Como exemplo, a mensagem com comando para efectuar *reset* ao registo de estado da ULM é seguida de duas mensagens de *Acknowledge*. A primeira destas mensagens confirma a recepção do comando, a segunda confirma que o *reset* foi efectuado com sucesso.

O término da chamada é mais uma vez facultativo, pois o utilizador pode pretender continuar a comunicar com a ULM ou terminar a ligação enquanto depende algum tempo na reconfiguração da ULM ao nível da base de dados.

7.2.4 Acerto da Hora da ULM

Cada ULM possui um registo da hora actual relativamente a uma data fixa. No entanto, com o decorrer do tempo, esse “relógio” vai-se desacertando. Quando se efectua uma ligação à ULM pode haver interesse em acertar a hora para manter um certo sincronismo nas datas dos eventos e das configurações.

Para ter uma medida do desacerto actual e também para determinar se há necessidade de acertar o relógio é estabelecida a chamada, no caso de entretanto se ter desligado, e enviada uma mensagem de pedido da hora actual da ULM (Comando → *Get_Time*), tal como evidencia a Figura 42. A mensagem de resposta é equivalente à mensagem recebida ao pedir a data da configuração. Se se pretender enviar a hora presente no PC ligado, envia-se então uma mensagem com comando *Set_Time* e 4 bytes de dados que correspondem ao tempo relativo a uma data predeterminada, em segundos.

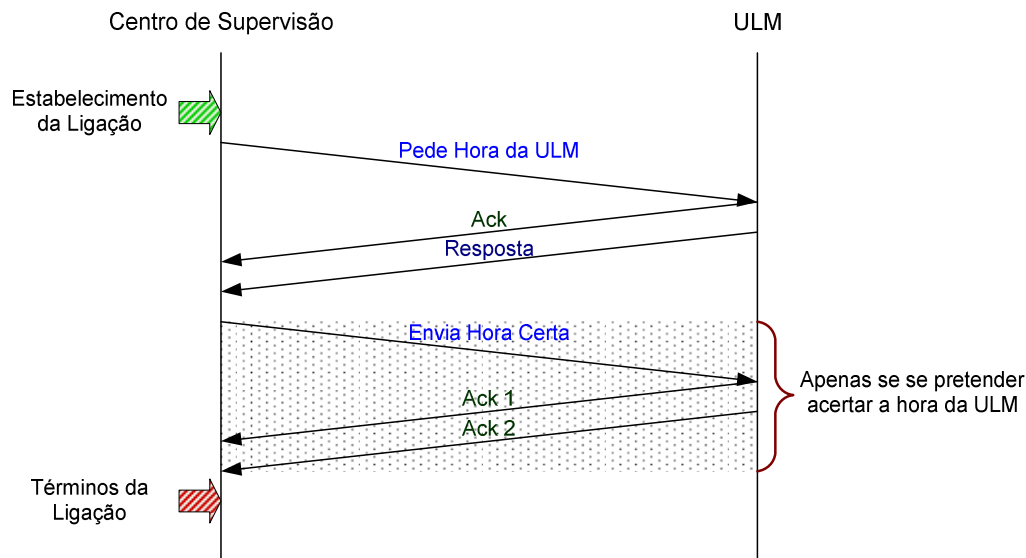


Figura 42 – Sequência de mensagens trocadas aquando do acerto da hora da ULM

O modelo de comunicação seguido cumpre, como já foi especificado, a transmissão de uma mensagem de *Acknowledge* e de uma resposta, a seguir a um pedido por parte do CSR, e a transmissão de duas mensagens de *Acknowledge* quando se trata de um comando.

7.2.5 Actuar numa saída da ULM

Quando se pretende actuar numa ou mais saídas sem efectuar uma reconfiguração da ULM, em princípio não se desligou a chamada previamente estabelecida. No entanto, caso isso tenha acontecido, deve sempre ser possível estabelecer a chamada de novo (Figura 43). De seguida é então enviada a mensagem de actuação na saída, a qual inclui o comando respectivo (Comando → Set_OutPort), 1 byte de identificação do porto de saída (ID do módulo + ID do porto) e 1 byte com o estado que se pretende que o porto assumira.

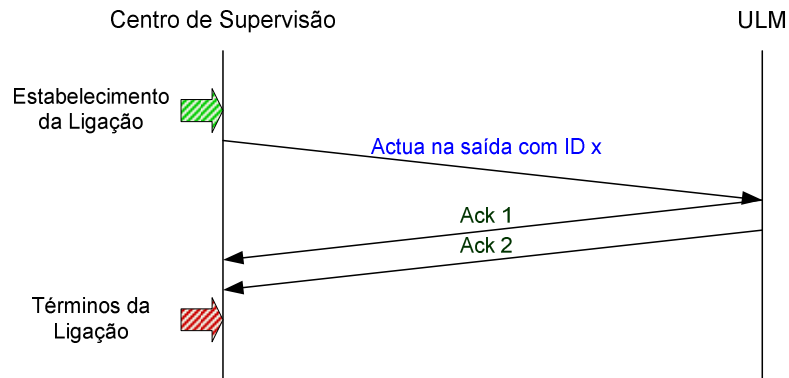


Figura 43 – Sequência de mensagens trocadas aquando da actuação numa saída

Cada uma destas mensagens corresponde à actuação numa única saída, sendo necessário repetir o envio para cada saída adicional, no caso de se pretender actuar em múltiplas saídas.

O término da ligação far-se-á então apenas após a actuação na última saída, caso não se pretenda continuar a comunicação com a ULM.

7.2.6 Análise do Estado da Memória de Histórico da ULM

A análise do estado da memória de histórico é efectuada através de uma comunicação semelhante à comunicação da hora da ULM. É estabelecida a chamada, apenas no caso de não estar já estabelecida, e é enviado um pedido do estado do histórico (Comando → Get_Hist_Stat), Figura 44.

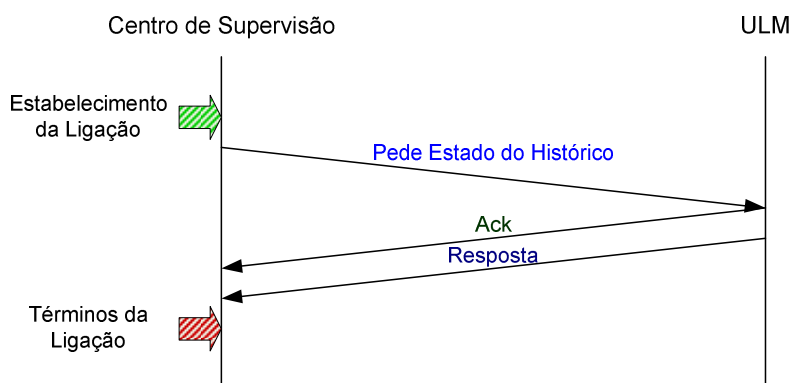


Figura 44 – Sequência de mensagens trocadas aquando do pedido de estado do Histórico na ULM

O pedido é seguido de uma mensagem de *Acknowledge* e uma resposta que contém o comando *Resp* e 4 bytes de dados, em que os 2 primeiros bytes contém o número de entradas de histórico

ocupadas e os últimos 2 bytes, o número total de entradas de histórico que a memória pode armazenar.

Após recepção desta informação pode haver interesse em efectuar o *download* do histórico, ou não, no caso de a percentagem de memória ocupada ser baixa. Nesta última situação, se nada mais houver a tratar pode-se terminar a chamada.

7.2.7 Download do Histórico

Quando se pretende efectuar o *download* do histórico, em princípio não se desligou a chamada previamente estabelecida para avaliar o estado do mesmo. No entanto, caso isso tenha acontecido, deve sempre ser possível estabelecer a chamada de novo (Figura 45). O *download* do histórico inicia-se com um pedido do CSR para efectuar o referido *download* (Comando → Download). A ULM responde com um *Acknowledge* e, assim que for possível, inicia a transferência das linhas de histórico, uma a uma, para o CSR. Cada mensagem que contém uma linha de histórico inclui o comando *Resp* e um conjunto de 5 bytes de dados (índice do evento (1 byte) + data de ocorrência (4 bytes)). A transmissão de cada linha de histórico da ULM para o CSR implica a transmissão de uma mensagem de *Acknowledge* no sentido contrário.

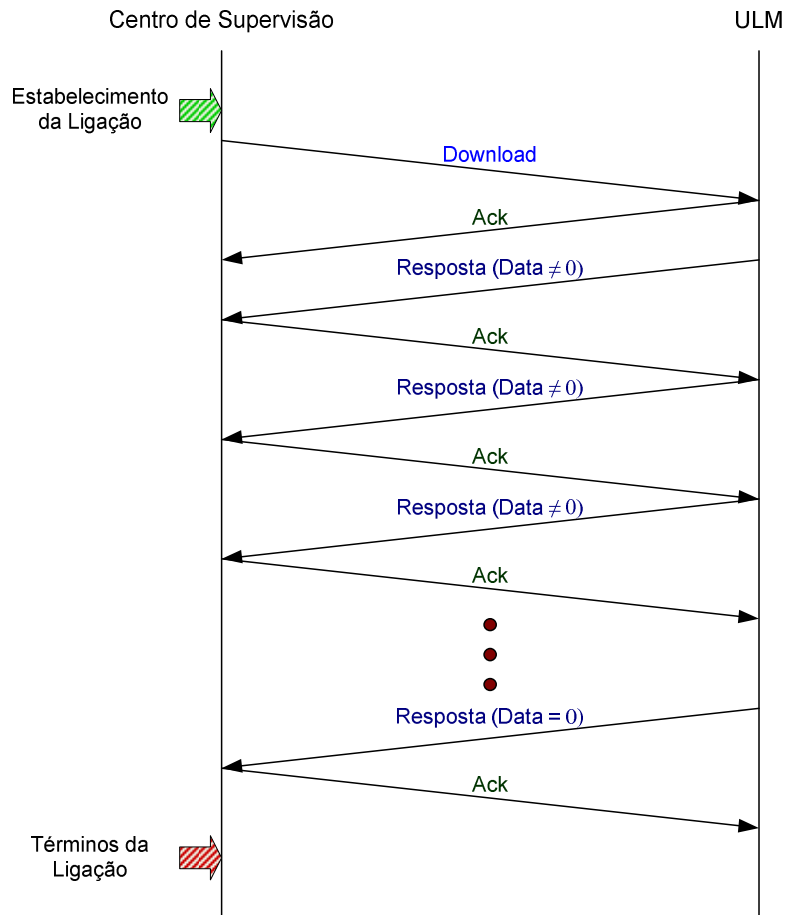


Figura 45 – Sequência de mensagens trocadas aquando do download do Histórico da ULM

O fim do *download* é assinalado pela transmissão de uma mensagem com comando *Resp* e 0 bytes de dados. A mensagem de *Acknowledge* transmitida de seguida pelo CSR acusa por um lado a recepção da última mensagem e por outro que “percebeu” que o *download* está terminado.

No caso de não se pretender continuar a comunicação com a ULM ou no caso de se pretender apenas libertar a ligação temporariamente é possível terminar a chamada.

7.2.8 Envio da Configuração para a ULM

Quando se pretende enviar para a ULM a configuração dos módulos efectua-se o estabelecimento da chamada, caso seja necessário, e de seguida envia-se uma mensagem de pedido para iniciar a transferência das referidas mensagens de configuração (Comando → EXT_ULM_CFG), Figura 46.

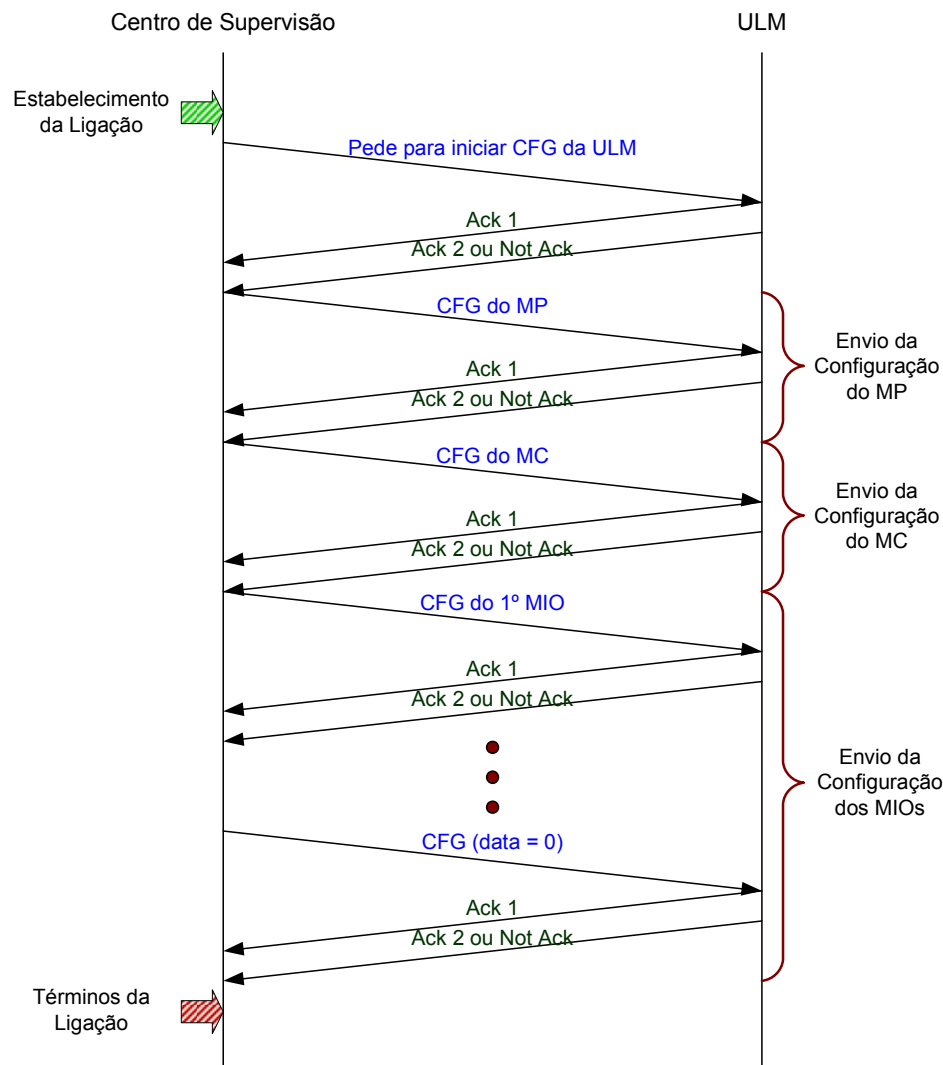


Figura 46 – Sequência de mensagens trocadas aquando da configuração dos módulos da ULM

A essa mensagem segue-se um *Acknowledge* de recepção por parte do Módulo de Comunicações e uma mensagem de *Acknowledge* ou *Not_Acknowledge*, consoante o pedido foi aceite e bem sucedido a nível do Módulo Principal ou não, respectivamente. A recepção de uma mensagem de *Not_Acknowledge* permite detectar erros de comunicação de uma forma mais rápida, sem necessidade de atingir o *timeout*. A recepção da segunda mensagem de *Acknowledge* dá início à transferência de mensagens com a configuração dos módulos constituintes da ULM, as quais são também seguidas de duas mensagens de *Acknowledge*, caso a transferência tenha sido bem feita. Começa por ser enviada uma mensagem com a configuração do Módulo Principal, seguida de uma mensagem com a configuração do Módulo de Comunicações, e depois, uma sucessão de mensagens com a configuração de cada um dos Módulos I/O. O conteúdo das mensagens que contém as configurações dos módulos é equivalente à estrutura de configuração dos mesmos.

A recepção, por parte da ULM, de uma mensagem de configuração com 0 bytes de dados indica o fim do envio de mensagens de configuração dos módulos, de modo que o Módulo de Comunicações da ULM deixa de estar à espera de receber mensagens específicas de configuração, voltando ao seu estado normal.

O envio da configuração de eventos, para a ULM, processa-se exactamente do mesmo modo que o envio da configuração dos módulos (Figura 47).

Inicialmente é enviado um pedido para iniciar o envio de mensagens contendo a configuração dos eventos (Comando → EXT_EVE_CFG), o qual recebe, como resposta, duas mensagens de *Acknowledge*, confirmando a recepção da mensagem e a aceitação ou não aceitação do pedido por parte do Módulo Principal. Segue-se a transmissão de mensagens com a configuração de eventos propriamente dita, intercalada pela recepção de 2 mensagens de *Acknowledge* indicando a recepção da configuração, primeiro ao nível do Módulo de Comunicações e depois ao nível do Módulo Principal. A mensagem de configuração de eventos contém a mesma estrutura que está guardada em memória (ver Figura 24).

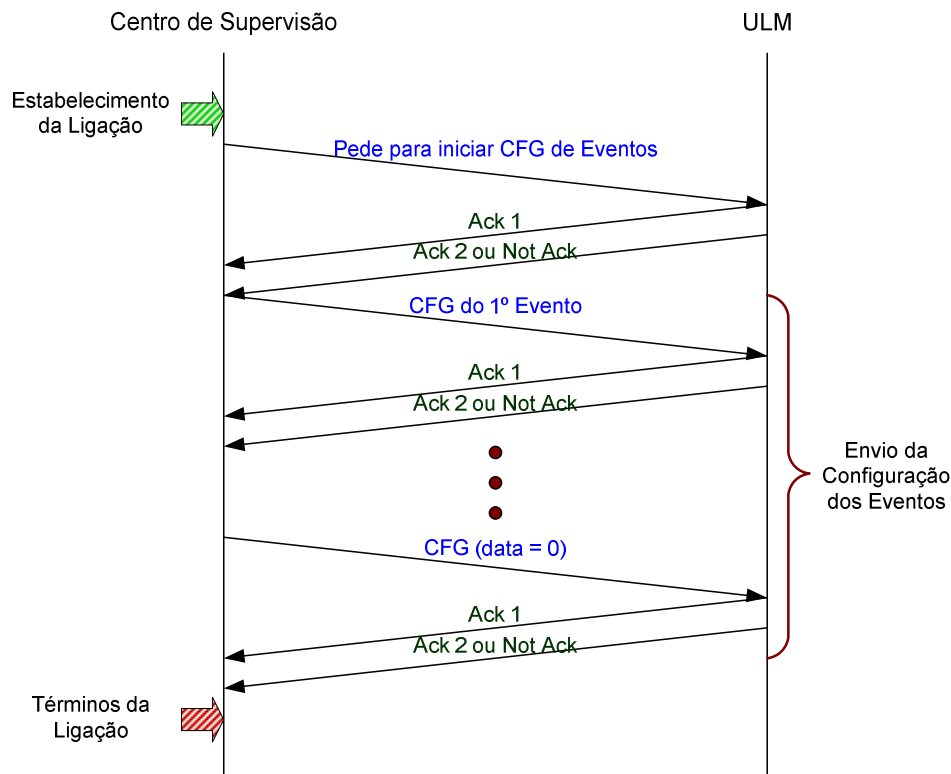


Figura 47 – Sequência de mensagens trocadas aquando da configuração dos eventos da ULM

O fim da configuração de eventos é assinalado também pelo envio de uma mensagem de configuração com 0 bytes de dados.

No final da transmissão a chamada deve poder ser terminada se assim se desejar.

7.3 Relações de Precedência e Limitações da Comunicação

Quando se pretende efectuar a configuração de uma nova ULM é necessário ter, do lado do CSR, uma imagem da configuração física da ULM, a qual deve ser obtida a partir da informação que o Módulo Principal tem, relativamente aos módulos que se encontram ligados à ULM. Assim, é absolutamente necessário que a comunicação se inicie pela transmissão desta informação, tal como foi indicado no ponto 7.2.2.

O efectuar de qualquer comunicação com uma ULM existente exige, em primeiro lugar, a confirmação da identificação da ULM, como já foi referido no ponto 7.2.3, seguida da confirmação da sincronização entre as datas da configuração guardada na base de dados e presente na ULM. Existe ainda, tal como na ligação a uma nova ULM, a necessidade de efectuar o reconhecimento da sua configuração física, para confirmar a correcta ligação de todos os módulos e permitir o acesso à configuração dos mesmos. A restante informação recolhida é ainda bastante importante para saber qual o estado de funcionamento actual da ULM. No entanto, a sequência de mensagens, não tem que ser efectuada necessariamente na ordem apresentada, para que a comunicação funcione. Apenas é estritamente necessário que toda esta informação seja do conhecimento do CSR antes de qualquer outra actuação, independentemente da ordem com que se obtém cada item de informação. É importante referir que, durante todo este processo de início da comunicação, a ULM pode responder a qualquer um destes comandos, ou até, a outros que iniciem transferência de outro tipo de informação. Esta sequência de mensagens constitui apenas uma possibilidade de início de comunicação com uma sequência de testes lógica que garante a obtenção da informação necessária e também a integridade da mesma.

A opção de efectuar a reconfiguração da ULM (mesmo ao nível da base de dados) pode, eventualmente, ser tomada sem que antes se tenha efectuado o *download* do histórico presente na ULM. Como a reconfiguração, em princípio, implica alteração da configuração de módulos ou eventos, e portanto a alteração da posição inicial da memória de histórico, o histórico que se mantiver na ULM considera-se irremediavelmente perdido uma vez que se perdem as referências. Por isso, sempre que o utilizador escolher a opção de efectuar a reconfiguração da ULM, é necessário efectuar um teste ao estado da memória de histórico e, no caso de o número de linhas de histórico presentes na ULM ser $\neq 0$, o utilizador deve ser informado desse facto podendo então optar por efectuar o *download* ou perder essas linhas de histórico. O mesmo deve acontecer sempre que se optar por fazer apenas a reconfiguração de eventos, pois, ao nível da ULM é

possível partir directamente para esta fase e as consequências da alteração correspondentes são equivalentes às já referidas para a reconfiguração total da ULM.

O Centro de Supervisão tem como objectivo principal fazer a monitorização de um conjunto de ULMs remotas. No entanto, sempre que a ligação com uma das ULMs estiver estabelecida, o CSR fica impedido de receber comunicação por parte de qualquer uma das outras, dado que a linha de comunicação está ocupada. A (re)configuração de ULMs a partir do CSR deve ser feita com o conhecimento de que implica um atraso na recepção de alarmes provenientes de outras ULMs. A possibilidade de estabelecer e terminar a chamada no início e no fim de cada fluxo de informação, respectivamente, permite libertar a ligação durante períodos de tempo em que efectivamente não é necessária, possibilitando que a ligação do CSR esteja mais disponível.

7.4 Comunicação por GSM ou Rede Fixa Comutada

Para avaliar a viabilidade da utilização destas tecnologias é necessário ter 2 pressupostos em atenção. Por um lado, que a transmissão de qualquer tipo de informação implica *handshaking*, pelo que, conforme a situação em concreto, uma das entidades, CSR ou ULM, se mantém, durante algum tempo, à espera de receber uma resposta por parte da outra entidade. Por outro lado, existem determinados fluxos de informação, como o *download* do histórico ou o envio de configuração, que implicam a transmissão de um conjunto de tramas de dados separadas e implicando, do mesmo modo, *handshaking*. Utilizando o envio de mensagens SMS sobre GSM, o tempo que mediará o envio de cada mensagem e a recepção da respectiva resposta seria da ordem de alguns segundos, na melhor das hipóteses, levando a que a transmissão de sequências longas de pacotes, como se trata do *download* de histórico ou do envio das configurações, fosse muito lenta. O tratamento de mensagens, tanto ao nível da ULM como ao nível do CSR, está sujeito a *timeouts* para permitir que o processamento do Módulo de Comunicações possua garantias temporais, não bloqueando indefinidamente à espera de uma mensagem. O dimensionamento dos *timeouts* para valores razoáveis em termos de SMSs inviabilizaria todo o restante processamento do Módulo de Comunicações e, conseqüentemente, do Módulo Principal.

Consideramos então que a utilização de SMS será apenas viável para uma ULM cuja configuração e *download* do histórico sejam sempre feitos localmente e que utilize a ligação GSM apenas para o envio de alarmes e recepção de comandos de actuação nas saídas. Estas comunicações envolveriam apenas o envio da mensagem pretendida, deixando a responsabilidade da entrega da mesma ao operador da rede GSM. O atraso inerente à comunicação, bem como a possibilidade de haver erros na transmissão teriam que ser assumidos como limitações do sistema. Uma alternativa a esta tecnologia seria a utilização de uma ligação de dados por GPRS, alternativa esta não estudada para integração nesta fase do projecto.

A utilização de rede fixa comutada parece ser uma boa opção uma vez que permite o estabelecimento de uma chamada de dados que fica disponível aos 2 intervenientes até ser terminada por um deles.

8 *Software* do PC

8.1 Introdução

Todo o projecto foi centrado, desde o início, na concepção da ULM, procurando desenvolver um sistema, o mais completo possível, ao qual pudesse ser facilmente associado, posteriormente, o *software* de monitorização e configuração. Este objectivo atingiu a fase final de desenvolvimento próximo da data limite para o término do trabalho, pelo que, foi desenvolvido apenas um *software* de demonstração das funcionalidades da ULM.

Neste capítulo será inicialmente apresentada a estrutura da base de dados que deve estar associada a cada ULM. Posteriormente será apresentada uma proposta para as funcionalidades do *software* do PC do Centro de Supervisão bem como do PC de Assistência. Finalmente será apresentada a estrutura do *software* de demonstração desenvolvido.

8.2 Estrutura da Base de Dados

A base de dados deve ter capacidade para armazenar a informação recolhida da monitorização e, simultaneamente, a informação necessária para controlar um conjunto de ULMs. Assim, cada entrada da base de dados representa uma ULM e contém toda a informação relacionada com esta. O Diagrama de Entidades e Relacionamentos que descreve a base de dados utilizada é apresentado na Figura 48.

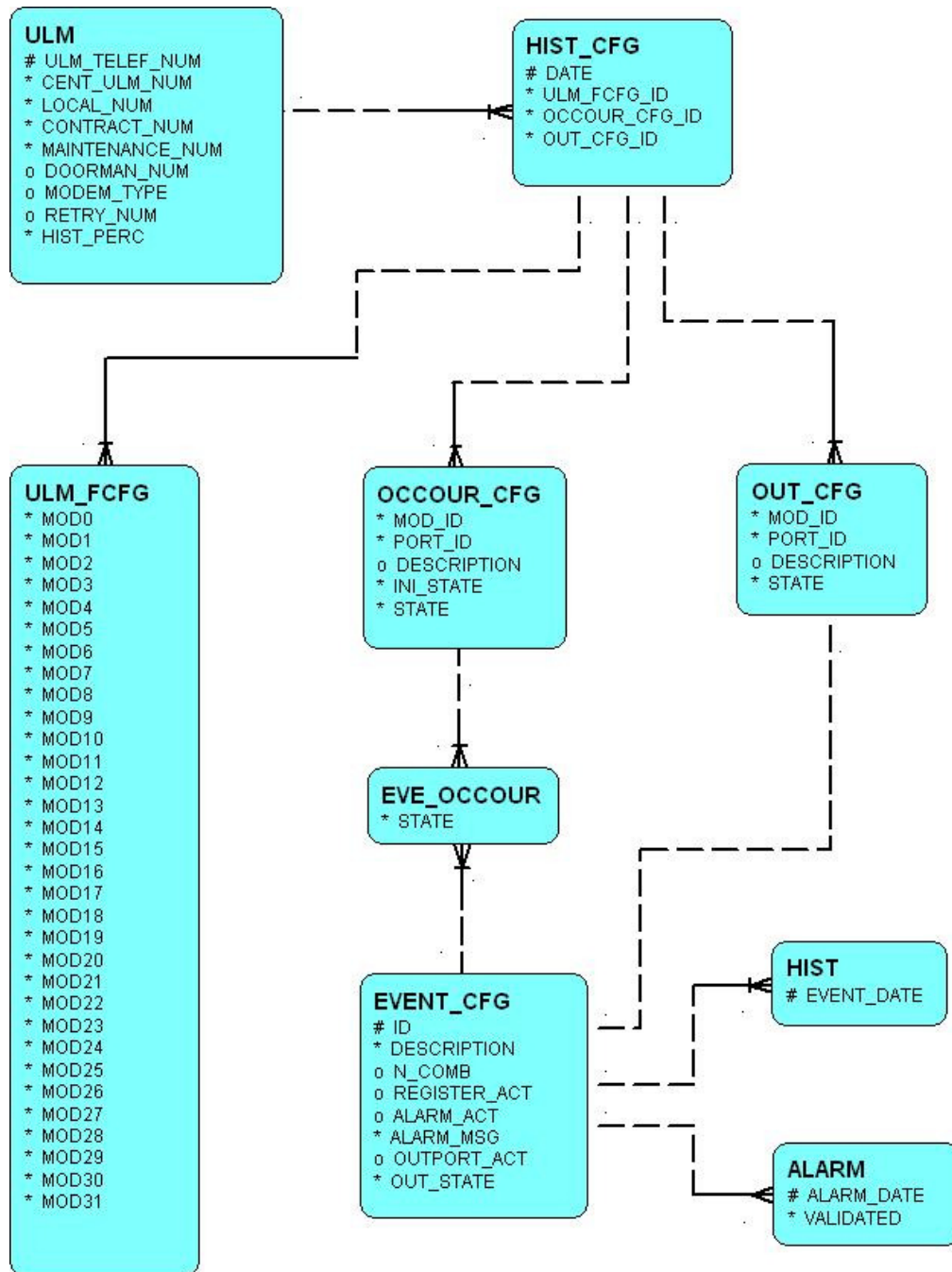


Figura 48 – Diagrama de Entidades Relacionais

A entidade ULM contém os campos que permitem identificar a ULM: número de telefone da ULM, número de telefone do CSR, local onde se encontra a ULM, número do contrato, número de telefone da empresa de manutenção, número de telefone do porteiro do prédio, tipo de modem ligado à ULM, número de tentativas para estabelecer a ligação e percentagem de ocupação do histórico à qual a ULM deve enviar um alerta para o CSR.

O Número de Telefone da ULM (NUM_TELEF_ULM) constitui o identificador da mesma e faz parte da chave de todas as tabelas a ela relacionadas.

Cada ULM terá sucessivas configurações, de acordo com as necessidades de monitorização, as quais serão identificadas pela data de configuração. Cada configuração é composta pela configuração física da ULM, a configuração das ocorrências, a configuração dos eventos e a configuração de estado dos portos de saída.

A configuração física da ULM indica se em cada posição está algum módulo ligado nessa e que tipo de módulo se encontra ligado (00 → Posição vazia, 01 → Módulo I/O, 10 → Módulo de Comunicações e 11 → Módulo Principal (em bits)).

A configuração das ocorrências é constituída pelo ID do módulo, ID do porto, descrição da ocorrência, estado inicial, estado a detectar e duração mínima para validar a activação da ocorrência.

A configuração dos portos de saída é constituída pelo ID do módulo, ID do porto, descrição do porto e estado.

A configuração de eventos contém um identificador, a descrição e o número de combinações (este último campo representa o número de portos que, combinados constituem o evento). Contém também campos relativos à acção que a ULM deve tomar aquando do despoletar do evento respectivo: se deve ser registado no histórico, se deve reportar alarme e qual a mensagem de alarme correspondente, e se deve actuar nalguma saída e o estado que se pretende que adopte.

Cada evento pode estar deste modo relacionado com um porto de saída para actuar no mesmo. Cada evento está ainda relacionado com uma ou mais ocorrências, até um máximo de 8. Esta relação inclui um atributo, o estado, que pretende identificar se se pretende detectar a ocorrência ou a não ocorrência relativamente a cada porto de entrada.

Existe ainda um repositório de alarmes despoletados pela detecção de eventos, que contém uma data de ocorrência e um atributo que indica se o alarme já foi validado.

A tabela de histórico contém o registo de todos os eventos ocorridos, incluindo também o registo dos alarmes, já contemplado, e que se encontra aqui repetido. Estes registos têm associada uma data de ocorrência.

8.3 PC do Centro de Supervisão

8.3.1 Cenários de Utilização

O *software* do PC do Centro de Supervisão deve, em primeiro lugar, permitir efectuar a monitorização de um conjunto de ULMs. Adicionalmente deve permitir actuar na ULM, o que compreende, entre outras acções, o pedido de informações relativas ao estado de funcionamento da ULM, actuação em saídas, (re)configuração e *download* do histórico. Assim, existem diferentes cenários de utilização do *software* do CSR, com diferentes capacidades e limitações.

Num cenário em que o CSR se encontra a efectuar a monitorização de um conjunto de ULMs o *software* encontra-se à “espera” de uma nova comunicação. Esta situação mantém-se a menos que o CSR pretenda efectuar qualquer tipo de acção sobre uma das ULMs a monitorizar que implique o estabelecimento de ligação com esta última e impeça o estabelecimento de qualquer outra ligação durante esse período.

Como a recepção de alarmes é o fim mais importante do CSR, deve ser possível efectua-lo sempre que a ligação estiver disponível, ou seja, este cenário de utilização deve ser permanente e estar a funcionar, em *background*, sempre que os modems do CSR estiverem disponíveis.

A actuação numa ULM implica o estabelecimento de comunicação com a mesma. Esta acção compreende toda a troca de informação preliminar necessária à identificação da ULM e à actualização da informação relativa ao seu estado de funcionamento. Requer, portanto, que se troque toda a informação que faz parte do estabelecimento de comunicação com uma ULM existente, 7.2.3. Posteriormente deve ser permitido fazer todo o tipo de actuação, respeitando as relações de precedência já indicadas.

Durante o período de actuação, enquanto a chamada estiver estabelecida, o primeiro cenário apresentado (monitorização) fica “bloqueado”, por indisponibilidade da ligação.

8.3.2 Estrutura do *Software*

Na Figura 49 é apresentado um esboço das funcionalidades que o *software* do CSR deve disponibilizar ao utilizador. As opções assinaladas a verde fazem surgir uma nova janela.

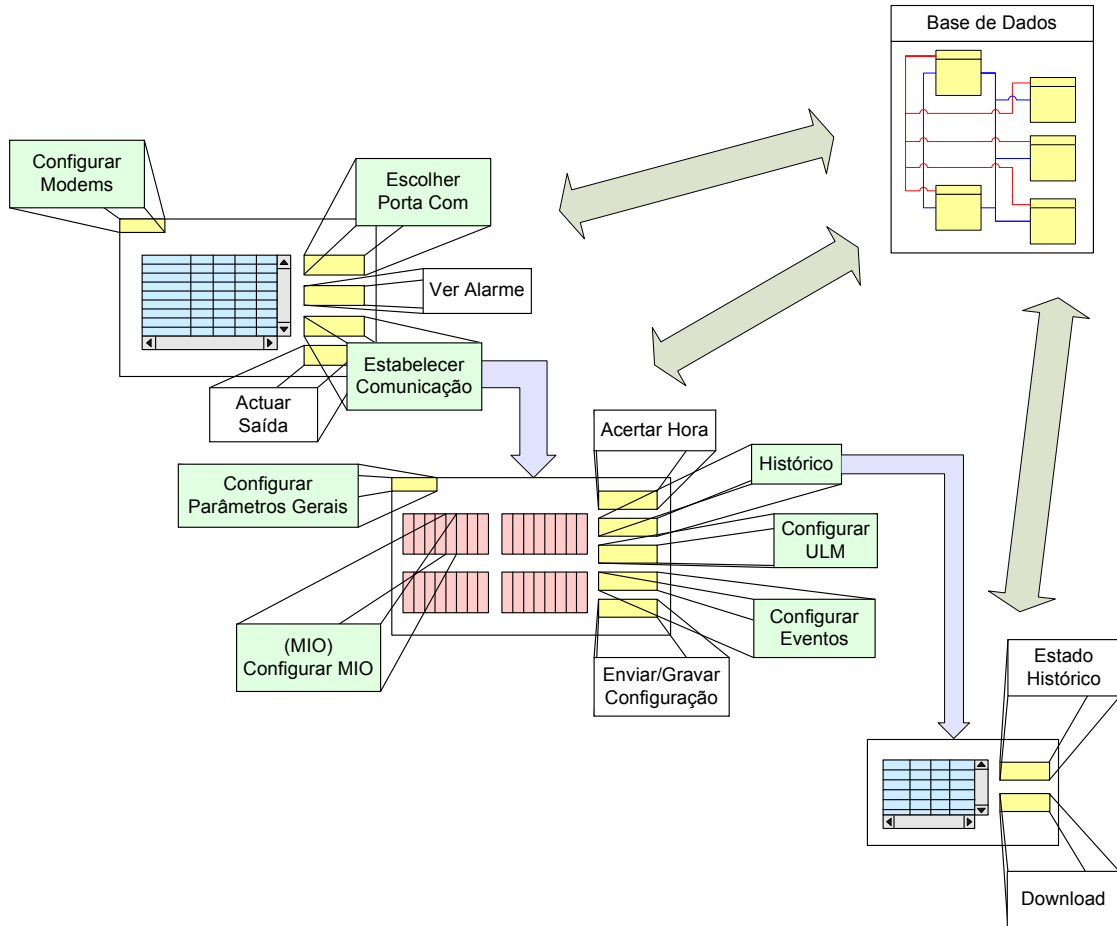


Figura 49 – Estrutura do Software do Centro de Supervisão

A janela principal da aplicação deve solicitar, inicialmente, que o utilizador configure as portas de comunicação com o exterior. Deve ficar definida qual a porta ligada ao modem rede fixa, qual a porta ligada ao modem GSM e qual a porta que pode ser utilizada para ligação directa a uma ULM (considera-se que o PC pode funcionar como PC do Centro de Supervisão Remota ou como PC de Assistência).

Considerando que esses parâmetros estão configurados, como a principal funcionalidade do *software* é efectuar a monitorização do conjunto de ULMs que lhe foram delegadas, a janela principal da aplicação deve apresentar uma tabela em que cada linha representa uma ULM e contém os campos mais relevantes para a monitorização da mesma (nome identificador atribuído à ULM, número de telefone, estado de funcionamento, último alarme recebido). Quando o CSR recebe um alarme (quer seja por rede fixa comutada ou por GSM) a ULM em questão deve ser posta, de alguma forma, em destaque, assinalando a recepção (auto-scroll de modo a mostrar a ULM afectada, colorir a linha correspondente).

Nesta primeira janela deve ser possível aceder a qualquer informação da base de dados, lista de alarmes, histórico e configuração, de qualquer ULM que conste da mesma. A visualização é

efectuada *off-line*, ou seja, não necessita de estabelecer ligação entre o CSR e a ULM. Deve também ser possível actuar numa saída de ULM com ligação por GSM (a actuação numa saída de uma ULM com ligação por rede fixa comutada implica o estabelecimento de comunicação e pode ser efectuada actuando na configuração da mesma).

Nesta janela deve ser disponibilizada a opção de estabelecer comunicação com uma ULM escolhida na tabela. A partir da opção de estabelecimento de comunicação, todas as funcionalidades do *software* são exclusivas para ULMs com ligação por rede fixa comutada. A partir do momento que se estabelece a referida comunicação deve ser apresentada ao utilizador uma janela específica da referida ULM.

Como durante o estabelecimento da comunicação o CSR recebeu informação relativa à configuração física da ULM, esta informação deve ser apresentada ao utilizador, de uma forma gráfica e o mais parecida possível com uma imagem da ULM, indicando possíveis alterações relativamente à disposição física guardada na base de dados. Deve ser ainda apresentado ao utilizador o estado de funcionamento da ULM, também recebido aquando do estabelecimento da comunicação.

Após o estabelecimento de comunicação deve ser possível, tal como antes, aceder a toda a informação que está na base de dados relativa à ULM. No entanto, a visualização da configuração de cada módulo deve também poder ser vista individualmente, escolhendo o módulo que se pretende no gráfico apresentado, permitindo ao utilizador ter uma visão mais nítida da configuração de toda a ULM. Apesar de ser desejável que o utilizador tenha acesso à configuração dos módulos, não é desejável que nesta fase seja possível efectuar qualquer alteração nessa configuração.

Esta segunda janela deve permitir ao utilizador obter algumas informações sobre a ULM como sejam a hora presente na mesma ou o estado da memória de histórico. A partir desta última informação, o utilizador deve poder decidir efectuar o *download* de todo o histórico presente na ULM.

O utilizador deve poder também optar por efectuar uma reconfiguração completa da ULM ou apenas uma reconfiguração dos eventos associados à presente configuração da ULM. Se o utilizador tomar uma destas opções sem que antes tenha sido efectuado o *download* do histórico deve ser informado que, se não efectuar o *download* antes da reconfiguração, todo o histórico presente na ULM se perde irremediavelmente.

A opção de efectuar a reconfiguração total da ULM deve começar por disponibilizar uma janela que contém os actuais parâmetros gerais de configuração, permitindo a sua alteração. De seguida deve ser possível alterar os parâmetros de configuração das ocorrências e dos portos de saída de cada Módulo I/O, escolhendo o módulo correspondente a partir do gráfico apresentado. Após esta reconfiguração estar completa, deve ser possível efectuar uma reconfiguração dos eventos. Estes são apresentados sequencialmente, uma vez que é a ordem pela qual são configurados que lhes

confere um índice que, mais tarde, actua com identificador. Apesar da reconfiguração de eventos dever ser efectuada sequencialmente, deve ser sempre possível voltar atrás para alterar alguns parâmetros. Para facilitar a configuração dos eventos, relativamente à escolha das ocorrências que combinadas os compõem, em cada janela de configuração de eventos deve ser disponibilizada uma tabela com a localização de cada ocorrência (ID do módulo e ID do porto) e a sua descrição. É importante referir ainda que, uma vez tomada a decisão de efectuar uma reconfiguração total da ULM, e portanto das ocorrências, não faz sentido efectuar a reconfiguração de eventos antes da restante reconfiguração, dado que a primeira depende das ocorrências previamente configuradas. É, no entanto, possível efectuar apenas a reconfiguração de eventos desde que se mantenham todas as características das ocorrências já configuradas.

No final da reconfiguração, seja esta total ou apenas ao nível dos eventos, deve ser possível enviar a mesma para a ULM correspondente, cumprindo o protocolo de comunicação já apresentado. A nova configuração deve ser também actualizada na base de dados. A actualização da configuração na base de dados deve incluir a adição de uma nova configuração à lista de configurações que a ULM assumiu durante todo o seu funcionamento até ao momento.

Após a actuação na ULM deve ser possível regressar à janela principal, continuando a efectuar a monitorização.

Como já foi referido, durante a actuação na ULM pode-se optar por desligar a ligação durante os períodos em que não é precisa para permitir que a monitorização do conjunto de ULMs continue a ser feita em *background*.

8.4 PC de Assistência

8.4.1 Cenários de Utilização

O *software* do PC de Assistência é muito semelhante ao *software* do PC de Supervisão. No entanto, o PC de Assistência não tem necessidade de se preocupar com a configuração dos *modems*, uma vez que comunica directamente com a ULM. Para além disso, o PC de Assistência não tem como principal objectivo a monitorização de um conjunto de ULMs, mas, como o próprio nome indica, prestar assistência, a qual compreende gestão do histórico e (re)configuração. À parte o cenário de monitorização, o *software* do PC de Assistência deve permitir efectuar todas as acções que são executadas pelo *software* do PC de Supervisão, como sejam actuar na ULM, o que inclui o pedido de informações relativas ao estado de funcionamento da ULM, actuação em saídas, (re)configuração e *download* do histórico. O *software* do PC de Assistência poderá, eventualmente, efectuar a monitorização temporária da ULM a que está ligado, apenas para

auxiliar a assistência prestada. Deste modo o PC de Assistência actua essencialmente num cenário de ligação permanente com a ULM permitindo efectuar qualquer actuação.

A actuação na ULM implica o estabelecimento de comunicação com a mesma, para permitir, em primeiro lugar, validar a identificação da ULM e, após esta validação, obter a informação necessária à prestação da assistência. O estabelecimento de comunicação com a ULM corresponde ao mesmo estabelecimento de comunicação apresentado já na Figura 41, excluindo, obviamente, o estabelecimento e o término da chamada, uma vez que neste caso se trata de uma ligação directa a partir da interface USB do Módulo de Comunicações. Depois deste procedimento deve ser permitido fazer todo o tipo de actuação, respeitando as relações de precedência já apresentadas.

8.4.2 Estrutura do Software

A primeira janela que o *software* do PC de Assistência deve disponibilizar deve ser, em tudo, semelhante à janela apresentada pela aplicação do PC do Centro de Supervisão. Do mesmo modo, as janelas seguintes, de apresentação e (re)configuração da ULM, devem ser equivalentes. As diferenças situam-se então apenas na ausência de capacidade para receber chamadas para transmissão de alarmes, e no facto de não necessitar de estabelecer e terminar chamadas para toda e qualquer troca de informação com a ULM.

O *software* pode, portanto, ser o mesmo do Centro de Supervisão. A escolha da porta série de ligação directa a uma ULM deve constituir informação suficiente para que o *software* conduza à actuação pretendida.

A base de dados que o PC de Assistência possui pode não ser completa, podendo constituir apenas uma fracção da base de dados presente no PC do CSR. Poderá, portanto, conter apenas a informação referente a um sub-conjunto de ULMs ao qual se vai efectuar a assistência.

8.5 Software de Demonstração

8.5.1 Introdução

A necessidade de ter um *software* de demonstração surgiu, por um lado, da necessidade de fazer um *debug* mais profundo ao *software* da ULM, mas também pelo facto de permitir demonstrar as funcionalidades da ULM concebida. O *software* desenvolvido não explora profundamente as potencialidades da ULM, permite apenas ao utilizador um conjunto de actuações simples que

emulam as transferências de informação que fazem parte do protocolo de comunicação implementado.

Esta aplicação foi desenvolvida utilizando o *Visual Basic .NET*. A escolha da linguagem foi feita tendo em consideração o factor tempo de aprendizagem, procurando otimizar ao máximo o tempo disponível para a concepção da aplicação.

O *software* não utiliza a base de dados descrita, mas sim uma estrutura baseada nessa base de dados e que contém toda a informação de uma ULM. A estrutura referente a cada ULM é guardada num ficheiro de modo a que possa estar disponível para cada utilização da aplicação.

8.5.2 Cenários de Utilização

Um utilizador do *software* de demonstração pretende, à partida, configurar uma ULM e provocar as ocorrências previamente configuradas de modo a fazer despoletar os eventos. Finalmente, deve efectuar o *download* do histórico e analisar o seu conteúdo de forma a determinar se tudo se processou como era esperado. Este foi o cenário de utilização considerado no desenvolvimento.

8.5.3 Estrutura do Software

A aplicação de demonstração começa por apresentar ao utilizador uma janela que permite: configurar os dispositivos ligados a cada porta de comunicação, escolher a porta de comunicação que se vai utilizar, estabelecer comunicação com uma nova ULM ou estabelecer uma comunicação com uma ULM existente. O primeiro estabelecimento de comunicação referido corresponde à troca de informação previamente apresentada na Figura 39, o segundo corresponde à troca de informação apresentada na Figura 41. Nenhuma das duas opções pode ser tomada sem que antes tenha sido escolhida a porta de comunicação. E, embora não seja obrigatório, convém configurar, em primeiro lugar, o tipo de dispositivo que está ligado à porta que vai ser utilizada (pelo menos), uma vez que dessa configuração depende a activação de alguns campos de configuração. A primeira janela constitui portanto uma janela de apresentação e inicialização, a qual é apresentada na Figura 50.

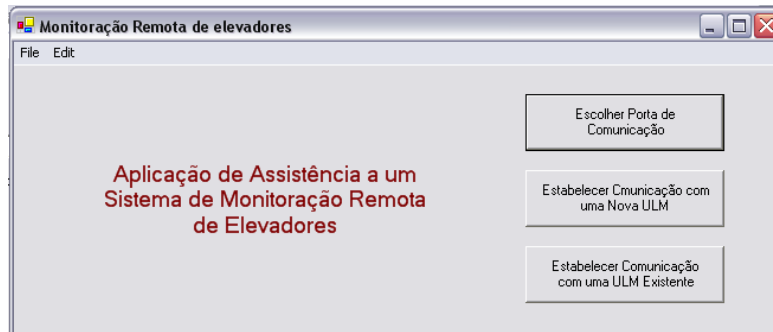


Figura 50 – Janela Principal da Aplicação de Demonstração

Na opção *Edit* do *MainMenu* existe um item de configuração (*Config*) que remete para a janela apresentada na Figura 51. Nesta janela é dada a possibilidade de configurar uma porta para ligação a um modem rede fixa, uma para ligação a um modem GSM e uma outra para ligação directa a uma ULM.

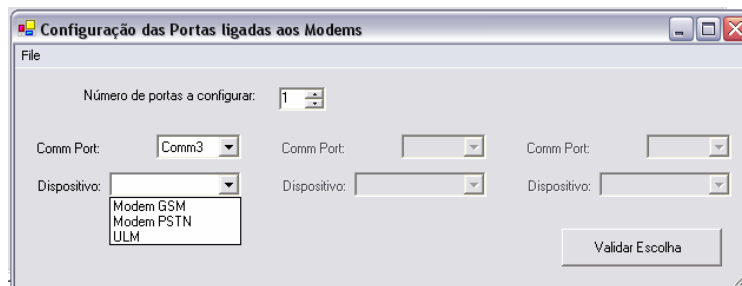


Figura 51 – Janela de configuração dos modems

A opção de escolher a porta de comunicação remete o utilizador para uma outra janela que permite escolher a porta e o *baud rate* que se vai utilizar na comunicação (Figura 52). O utilizador será então informado da disponibilidade da porta escolhida e, caso esteja disponível, pode regressar à primeira janela para iniciar a monitorização ou actuar na ULM. O utilizador, ao validar a escolha, é informado do tipo de dispositivo que foi configurado na porta de comunicação escolhida.

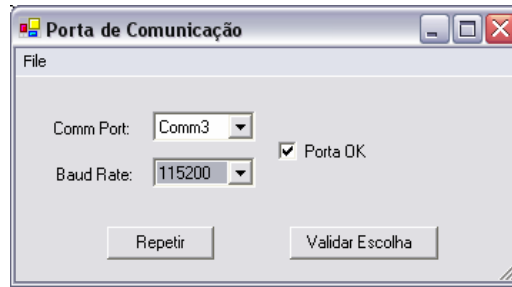


Figura 52 – Janela de Escolha da Porta de Comunicação

Se o utilizador escolher, na primeira janela, a opção de estabelecer comunicação com a ULM, o *software* remete toda a acção para uma janela que fornece uma imagem da ULM em questão, apresentando a posição dos vários módulos. No decorrer da descrição da estrutura do *software* esta janela será denominada *Janela da ULM*, Figura 53.

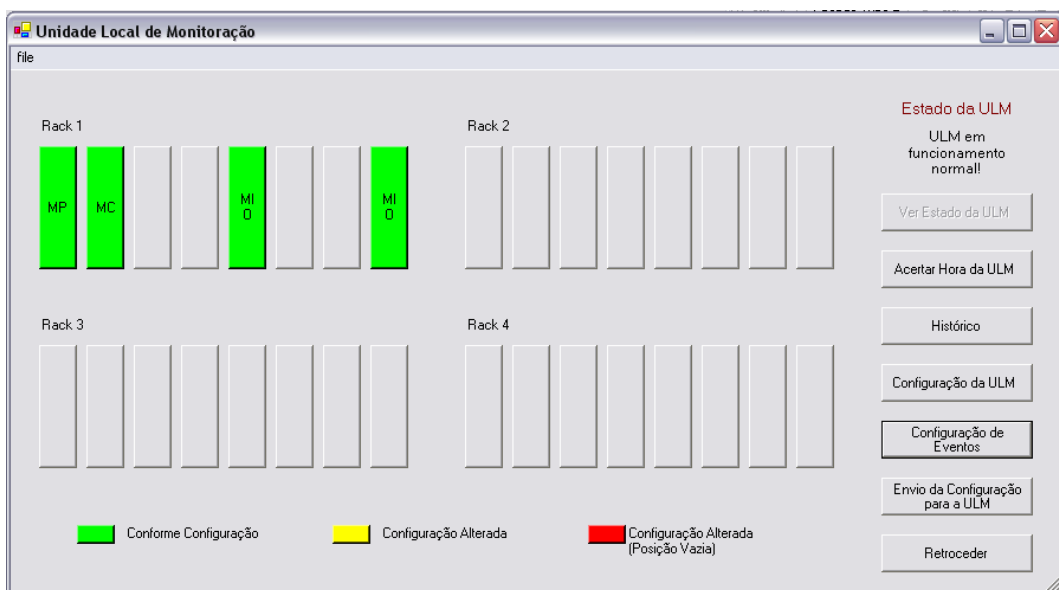


Figura 53 – Janela da ULM

Na transição para esta janela há uma questão que deverá ser tida em consideração: o facto de se estar a estabelecer a comunicação com uma nova ULM (o que significa que não possui ainda configuração) ou com uma ULM já existente (que no contexto deste *software* significa que já tinha sido configurada e estaria, eventualmente, em funcionamento). No primeiro caso, não há necessidade de verificar a data da configuração presente na ULM, assim como não há necessidade de verificar qual o estado de funcionamento da ULM. No segundo caso, não só é necessário fazer as verificações referidas como se deve ir buscar a antiga configuração ao ficheiro correspondente e tornar essa configuração acessível à aplicação. A janela apresentada, neste

segundo caso, apresenta, além da eventual existência de uma alteração do estado de funcionamento normal da ULM, uma estrutura gráfica das posições dos módulos, indicando as que sofreram alterações relativamente à configuração anterior (Figura 53). O estado de funcionamento, quando fora do normal, é visível através de um clique no botão correspondente. A configuração de cada módulo I/O é visível através de um clique sobre cada um deles, não sendo, no entanto, possível alterar qualquer parâmetro. A configuração de um conjunto de parâmetros gerais da ULM é visível a partir de um clique sobre o Módulo Principal.

Esta *Janela da ULM* permite acertar a hora da ULM, acção esta em que o PC começa por obter a hora actual da ULM e, por comparação com a hora do PC pode optar por acertar ou deixar estar a hora que estava.

A *Janela da ULM* permite ainda analisar o histórico da ULM. A escolha desta opção abre uma outra janela que permite visualizar a linhas de histórico presentes na estrutura, verificar o estado da memória de histórico da ULM e efectuar o *download* (Figura 54).

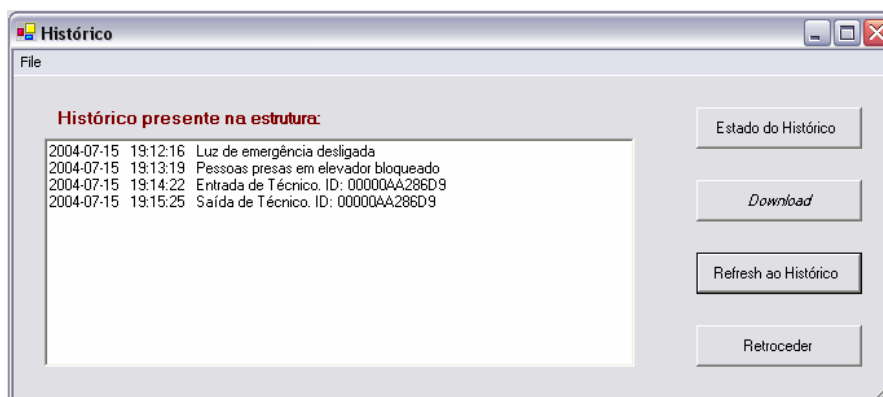


Figura 54 – Janela de Visualização do Histórico

Voltando à *Janela da ULM*, esta permite ainda, ao utilizador, optar por efectuar a (re)configuração da ULM. Ao escolher esta opção, a aplicação apresenta uma janela de configuração dos parâmetros gerais da ULM, como os números de telefone da ULM e da Central, o número de tentativas para estabelecer uma ligação e a percentagem de utilização do histórico à qual deve ser enviado um alerta para o Centro de Supervisão (Figura 55).

Figura 55 – Janela de Configuração dos Parâmetros Gerais da ULM

Após validação da informação introduzida nesta janela, torna-se possível efectuar ou alterar a configuração da cada módulo I/O, que se encontrava, até aqui, bloqueada para escrita (Figura 56). Em cada janela de (re)configuração de módulos I/O é possível activar ou desactivar cada um dos portos de entrada e, para cada um deles, definir o seu estado inicial, o estado a detectar, o tempo mínimo que o sinal se deve manter no estado que se pretende para validar a ocorrência e definir uma descrição para melhor identificar a ocorrência associada. Relativamente aos portos de saída, é possível definir um estado e uma descrição para cada um deles.

Para efectuar uma possível alteração dos parâmetros gerais da ULM, pode-se efectuar um *clique* sobre o Módulo Principal e ter de novo acesso à janela correspondente.

Activar Porto	Estado a detectar	Estado inicial	Tempo mínimo para validar	Descrição do porto
<input checked="" type="checkbox"/>	1	0	0.50s	Elevador parado entre andares
<input checked="" type="checkbox"/>	1	0	0.50s	Botão de emergência accionado
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

Porto de Saída	Estado	Descrição do porto
Porto de Saída 1	0	Sinalização de pessoas presas no elevador
Porto de Saída 2	0	

Figura 56 – Janela de Configuração de um Módulo I/O

Após concluir a (re)configuração dos módulos e, estando ainda na *Janela da ULM*, é possível optar por efectuar a (re)configuração dos eventos. É importante referir que, já antes do concluir da configuração dos módulos era possível tomar esta opção, no entanto, como a configuração dos eventos depende das ocorrências configuradas, só faz sentido efectuar uma (re)configuração dos eventos ou após completa (re)configuração dos módulos ou antes de a efectuar, se não se pretender, de todo, reconfigurar os módulos I/O.

Quando a opção de efectuar a (re)configuração dos eventos é tomada, surge, mais uma vez, uma nova janela, apresentada na Figura 57.

	Porto	Módulo	Detectar	Ver Descrição	Descrição de cada Porto:
Ocorrência 1	1	13	Ocorrência	Ver Descrição	Elevador parado entre andares
Ocorrência 2	2	13	Ocorrência	Ver Descrição	Botão de emergência accionado
Ocorrência 3				Ver Descrição	
Ocorrência 4				Ver Descrição	
Ocorrência 5				Ver Descrição	
Ocorrência 6				Ver Descrição	
Ocorrência 7				Ver Descrição	
Ocorrência 8				Ver Descrição	

Figura 57 – Janela de Configuração de um Evento

A janela apresentada representa o evento índice 1. Nesta janela é possível associar ao evento uma descrição, definir o número de ocorrências que se vão combinar para formar o evento (até um máximo de 8), identificar cada uma das ocorrências (ID do módulo + ID do porto) e definir se o evento é despoletado quando a ocorrência se manifesta ou quando, por outro lado, está inactiva. Para confirmar se a ocorrência que se está a considerar é a pretendida é possível visualizar a sua descrição. A janela permite também definir a acção que está associada ao despoletar do evento (registo no histórico e/ou envio de alarme e/ou actuação numa saída) bem como a identificação e o estado do porto de saída em que se vai actuar (se for caso disso) e a mensagem de alarme a mostrar aquando do despoletar desse evento. A janela permite uma navegação através dos

eventos, permitindo efectuar possíveis alterações da respectiva configuração, e ainda a anulação de um qualquer evento de uma lista já configurada.

9 Conclusões

9.1 Especificação do sistema

O sistema desenvolvido possui uma capacidade de expansão até um número máximo de 30 módulos I/O, cada um deles com 8 entradas e 2 duas saídas. Consequentemente, o *software* desenvolvido terá capacidade para fazer a monitorização de 240 eventos diferentes (30 módulos * 8 portos de entrada). Estes limites terão porventura sido sobredimensionados, uma vez que o sistema dificilmente será utilizado no seu formato máximo, e será também difícil encontrar 240 situações susceptíveis de ser sinalizadas para que constituam eventos. Esta especificação teve consequências a nível do desenvolvimento do *software*, tendo dificultado o cumprimento das especificações temporais. Tiveram assim que ser encontradas alternativas para agendar os processos a executar, as filas de espera.

A resolução temporal do sistema (250ms) pode também ser considerada elevada relativamente às necessidades da aplicação final, dado que se pretende, numa primeira fase, a integração num conjunto de elevadores, o qual não tem elevada reactividade. Para esta aplicação uma resolução temporal de 1s seria, eventualmente, suficiente. No entanto, como se trata de um sistema de monitorização que poderia ser integrado num outro sistema que não um sistema de elevadores, possuindo esta resolução temporal, a sua aplicação não está tão limitada aos sistemas com baixa reactividade.

Na especificação do sistema o processamento foi distribuído por vários módulos. Os módulos principal e de comunicações foram, por isso, projectados de forma separada, tendo já em vista uma solução de *Hardware* com baixa capacidade de processamento. Esta solução tinha como objectivo diminuir o custo do sistema. No entanto a escolha de um microcontrolador mais potente que permitisse a integração de todo o processamento principal e das comunicações num mesmo módulo teria facilitado bastante a implementação das comunicações quer ao nível da ULM, quer ao nível das comunicações exteriores. Para além disso, como a maior parte do custo do sistema vai estar centrado no custo dos PCBs, a eliminação de um módulo poderia dar origem a um custo global inferior.

9.2 Hardware Escolhido

Relativamente ao microcontrolador escolhido, embora se tenha procurado uma solução barata e que possuísse a maior parte dos periféricos necessários já integrados, o custo de desenvolvimento que a escolha representou revelou-se enorme. As soluções encontradas estiveram sempre dependentes das limitações do microcontrolador, tendo-se revelado muito mais complexas do que seria desejável. Se se tratasse de uma empresa, teria provavelmente perdido mais dinheiro com o tempo adicional despendido do que perderia com a integração de um microcontrolador com maior capacidade de processamento e maiores recursos, tendo em vista um volume de produção relativamente baixo (100 unidades).

O protótipo desenvolvido no projecto possui uma muito baixa densidade de componentes por placa, o que permite, num cenário de produção final, o desenvolvimento de um sistema bastante compacto.

Além do microcontrolador, os restantes componentes são essencialmente passivos e por isso, de baixo custo. Decorrente deste facto, do baixo preço do microcontrolador, da pequena quantidade de componentes por módulo e do pequeno tamanho necessário para os PCBs, tudo contribuiu para que no seu conjunto, o sistema final tenha um custo baixo.

9.3 Software da ULM

O *software* da ULM centra-se essencialmente na comunicação entre os módulos, uma vez que se trata de um sistema distribuído. A principal característica do *software* desenvolvido é a existência de uma elevada redundância ao nível das comunicações, de modo que possui, portanto, uma tolerância a falhas bastante elevada a esse nível.

9.4 Comunicação GSM vs. PSTN vs. GPRS

A utilização de rede fixa comutada para a comunicação entre o CSR e as ULMs surge como uma opção viável uma vez que permite o estabelecimento de uma chamada de dados que os dois intervenientes na comunicação podem utilizar para transferir os dados que pretendem, até que um dos quais tome a iniciativa de a terminar.

A utilização de GSM, para efectuar a comunicação entre a ULM e o Centro de Supervisão Remoto, não permite usufruir de todas as funcionalidades do sistema de Supervisão. Toda a comunicação tem que estar sempre sujeita a restrições temporais, e o envio de mensagens SMS, para além de ser um processo lento, não oferece garantias ao nível do tempo de entrega. Concluimos então que apenas a comunicação de alarmes e a actuação em saídas pode ser efectuada utilizando SMSs sobre GSM, deixando, no entanto, a responsabilidade da entrega da mesma ao operador da rede GSM.

A ligação de uma ULM que não possua uma linha de rede fixa comutada a um CSR poderá eventualmente ser feita utilizando um modem GPRS e efectuando uma ligação de dados. Esta possibilidade não foi estudada no âmbito do projecto.

9.5 Software do PC

O *software* de demonstração desenvolvido utiliza uma estrutura interna à aplicação para manipular todos os dados relativos à ULM, o que limita as operações que se podem efectuar sobre os referidos dados. No entanto, mesmo que o *software* tivesse sido desenvolvido com a capacidade de manipular uma base de dados, não seria possível desenvolver uma aplicação que pudesse ser, na prática, utilizada por uma empresa de manutenção. Esta limitação deve-se, por um lado, ao facto de haver um tempo limitado para o desenvolvimento do projecto que não permitiu o investimento necessário na aquisição de conhecimento que permitiria o desenvolvimento de uma aplicação comercial. Por outro lado, para ter conhecimento das funcionalidades que o *software* deveria oferecer seria de extrema importância ter informações provenientes de um utilizador real do equipamento, que teria sensibilidade relativamente às necessidades que seriam sentidas.

Como já foi referido, a interface disponibilizada pela ULM permite desenvolver o referido *software* à imagem do que for idealizado pelo cliente.

O desenvolvimento do *software* do PC constitui uma actividade que pode atingir elevados níveis de complexidade, dependendo, como vimos, do que o utilizador pretender e do tempo que se puder investir, de modo que pode constituir em si, um futuro projecto de 5º ano.

9.6 Actualidade do Trabalho Desenvolvido

As principais diferenças encontradas entre os sistemas que se encontravam no mercado em 2003 relativamente aos sistemas actuais centram-se na interface com o utilizador, apresentando uma maior possibilidade de visualização gráfica dos resultados e relatórios, mecanismos de correlação

de resultados, permitindo tirar o maior partido possível das funcionalidades do sistema de monitorização.

Adicionalmente, o acesso à informação remotamente, bem como a comunicação e actuação nos dispositivos distantes assentam agora em tecnologias *web*, permitindo que o controlo e monitorização dos sistemas possam ser efectuados de qualquer lugar, através de ligações seguras, que garantem autenticação no acesso e possibilitam que o trabalho de monitorização não esteja confinado às centrais.

Como as principais diferenças se centram na comunicação e na aplicação de monitorização, considera-se que o trabalho realizado não perdeu actualidade, constituindo ainda um projecto de um sistema de monitorização que poderia ser integrado tanto em conjuntos de elevadores como noutros sistemas que, do mesmo modo, não exigissem uma grande reactividade. De qualquer modo, a aplicação deste sistema como sistema de monitorização remota implicaria o desenvolvimento da aplicação de monitorização, o qual foi apenas especificado no decorrer do trabalho desenvolvido.

Anexos

A Características e Preços de Memórias

A.1 EEPROMs

A.1.1 Catalyst

A *Catalyst* [26] disponibiliza um conjunto de memórias EEPROM com interface SPI que variam em capacidade e protecção de escrita. Na Tabela 2 estão apresentadas as principais características de cada uma destas memórias.

Dimensão (linhas e colunas)	Interface (modos)	"Package"	Protecção de escrita	Preço
1K - 16K ($2^{(n-3)}$ x 8)	SPI (0,0 e 1,1)	8-pin DIP/SOIC, 8/14-pin TSSOP e 8-pinMSOP	1/4, 1/2 ou todo o array da EEPROM	
1K - 16K ($2^{(n-3)}$ x 8)	SPI (0,0 e 1,1)	8-pin DIP/SOIC, 8/14-pin TSSOP e 8-pinMSOP	Primeira página, última, qualquer 1/4 do array ou o 1/2 inferior do array	
32K - 64K ($2^{(n-3)}$ x 8)	SPI (0,0 e 1,1)	8-pin DIP/SOIC, 14-pin TSSOP e 20-pinMSOP	1/4, 1/2 ou todo o array da EEPROM	
32K - 64K ($2^{(n-3)}$ x 8)	SPI (0,0 e 1,1)	8-pin DIP/SOIC e 14-pin TSSOP	Primeira página, última, qualquer 1/4 do array ou o 1/2 inferior do array	
128K / 256K ($2^{(n-3)}$ x 8)	SPI (0,0 e 1,1)	8-pin DIP/SOIC, 16-pin SOIC, 14-pin TSSOP e 20-pin TSSOP	1/4, 1/2 ou todo o array da EEPROM	

Tabela 2 – Memórias EEPROM da Catalyst

A.1.2 Farnell

No catálogo da *Farnell* [27] (que constituí um fornecedor acreditado) podemos encontrar a lista de EEPROMs apresentada na Tabela 3.

Fabricante	Número do fabricante	Detalhes	"Package"	Dimensão (linhas e colunas)	Vcc (V)	Interface	T(°C)	Preço 100+
ST Microelectronics	M95080-WMN6	IC-SM EEPROM	PDIP8 / SO8 / TSSOP8 / SBGA5	8K (1K x 8)	2,5	SPI	-40 a 85	0,67
ST Microelectronics	M95080-WBN6	IC-EEPROM	PDIP8 / SO8 / TSSOP8 / SBGA6	8K (1K x 8)	2,5	SPI	-40 a 85	0,67
ST Microelectronics	M95160-WMN6	IC-SM EEPROM	PDIP8 / SO8 / TSSOP8 / TSSOP14	16K (2K x 8)	2,5	SPI	-40 a 85	1,02
ST Microelectronics	M95160-WBN6	IC-EEPROM	PDIP8 / SO8 / TSSOP8 / TSSOP14	16K (2K x 8)	2,5	SPI	-40 a 85	1,02
ST Microelectronics	M95320-WMN6	IC-SM EEPROM	PDIP8 / SO8 / TSSOP8 / TSSOP14	32K (4K x 8)	2,5	SPI	-40 a 85	2,18
ST Microelectronics	M95320-WBN6	IC-EEPROM	PDIP8 / SO8 / TSSOP8 / TSSOP14	32K (4K x 8)	2,5	SPI	-40 a 85	2,18
ST Microelectronics	M95640-WMN6	IC-SM EEPROM	PDIP8 / SO8 / TSSOP8 / TSSOP14	64K (8K x 8)	2,5	SPI	-40 a 85	1,89
ST Microelectronics	M95640-WBN6	IC-EEPROM	PDIP8 / SO8 / TSSOP8 / TSSOP14	64K (8K x 8)	2,5	SPI	-40 a 85	1,89

Tabela 3 – EEPROMs (ST Microelectronics)

A.1.3 Mouser Electronics

A *Mouser Electronics* [28] apresenta a lista de EEPROMs apresentada na Tabela 4.

Fabricante	Número do fabricante	Detalhes	Dimensão (linhas e colunas)	"Package"	Interface	Preço 100+
Microchip	25AA080/P	Serial EEPROM	8K (1K x 8)	PDIP	SPI	0,74
Microchip	25AA080/SN	Serial EEPROM	8K (1K x 8)	SOIC-8	SPI	0,74
Microchip	25C080/P	Serial EEPROM	8K (1K x 8)	PDIP-8	SPI	0,74
Microchip	25C080/SN	Serial EEPROM	8K (1K x 8)	SOIC-8	SPI	0,74
Microchip	25LC080/P	Serial EEPROM	8K (1K x 8)	PDIP-8	SPI	0,74
Microchip	25LC080SN	Serial EEPROM	8K (1K x 8)	SOIC-8	SPI	0,74
Microchip	25C160/P	Serial EEPROM	16K (2K x 8)	PDIP-8	SPI	0,74
Microchip	25C160/SN	Serial EEPROM	16K (2K x 8)	SOIC-8	SPI	0,74
Microchip	25LC160/P	Serial EEPROM	16K (2K x 8)	PDIP-8	SPI	0,74
Microchip	25LC160/SN	Serial EEPROM	16K (2K x 8)	SOIC-8	SPI	0,74
Microchip	25C320/P	Serial EEPROM	32K (4K x 8)	PDIP-8	SPI	0,62
Microchip	25C320/SN	Serial EEPROM	32K (4K x 8)	SOIC-8	SPI	0,62
Microchip	25LC320/P	Serial EEPROM	32K (4K x 8)	PDIP-8	SPI	0,62
Microchip	25LC320/SN	Serial EEPROM	32K (4K x 8)	SOIC-8	SPI	0,62

Tabela 4 – EEPROMs (Microchip)

A.1.4 RS

No catálogo da RS [29] (que constituí um fornecedor acreditado) podemos encontrar a lista de EEPROMs apresentada na Tabela 5.

Fabricante	Referência	Dimensão (linhas e colunas)	"Package"	Vcc(V)	Iccstby(µA)	Icc(mA)	Interface	T(°C)	price 100+
STM	M95160-WBN6	16K (2K x 8)	DIP8	2,5 a 5,5	10	4	SPI	-40 a 125	0,94
STM	M95160-WMN6	16K (2K x 8)	SO8	2,5 a 5,5	10	4	SPI	-40 a 125	0,94
STM	M95320-WBN6	32K (4K x 8)	DIP8	2,5 a 5,5	10	4	SPI	-40 a 125	1,99
STM	M95320-WMN6	32K (4K x 8)	SO8	2,5 a 5,5	10	4	SPI	-40 a 125	1,99
STM	M95640-WBN6	64K (8K x 8)	DIP8	2,5 a 5,5	10	4	SPI	-40 a 85	2,83
STM	M95640-WMN6	64K (8K x 8)	SO8	2,5 a 5,5	10	4	SPI	-40 a 85	2,83
FAIR	FM25C640ULEN	64K (8K x 8)	DIP8	2,5 a 5,5	50	3	SPI	-40 a 85	1,28

Tabela 5 – EEPROMs (STM & FAIR)

A.2 Memórias Flash

A.2.1 ChipCatalog

A *Chip Catalog* [30] apresenta memórias *flash* de 512K e 1M de capacidade. Na Tabela 6 são apresentadas as suas principais características.

Fabricante	Dimensão (linhas e colunas)	Vcc (V)	Interface	"Package"
Atmel	512K (65536 x 8)	2,7 a 3,6	SPI (0,0 e 1,1)	8-lead JEDEC SOIC e 8-lead SAP Package
Atmel	1M (131072 x 8)	2,7 a 3,7	SPI (0,0 e 1,1)	8-lead JEDEC SOIC e 8-lead SAP Package

Tabela 6 – Memórias Flash (Atmel)

A.2.2 Farnell

No catálogo da *Farnell* [27] podemos encontrar a lista de memórias *flash* apresentada na Tabela 7.

Fabricante	Número do fabricante	Detalhes	Vcc (V)	Interface	T(°C)	Preço
Atmel	AT45DCB002	2M byte (16M-bit) Flash Card	2,7	SPI	max 85	9,07

Tabela 7 – Memória Flash (Atmel)

A.2.3 SST

A SST [31] apresenta uma memória *flash* com as características apresentadas na Tabela 8.

Fabricante	Referência	Dimensão	Interface	"Package"	Byte-Program Time
SST	SST25VF512	512K	SPI (0,0 e 1,1)	8-lead SOIC / 8-contact WSON	14us

Tabela 8 – Memória Flash (SST)

B Esquemáticos Completos

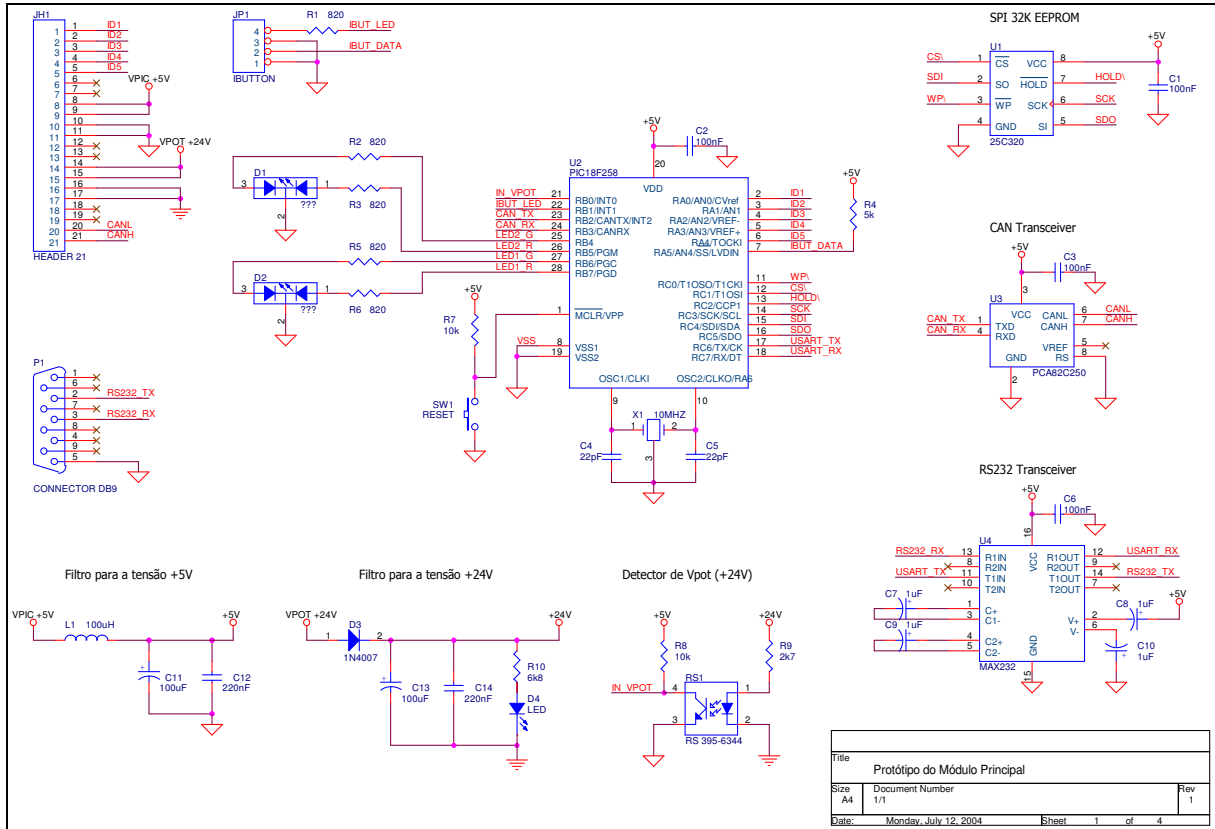


Figura 58 – Esquemático do protótipo do Módulo Principal

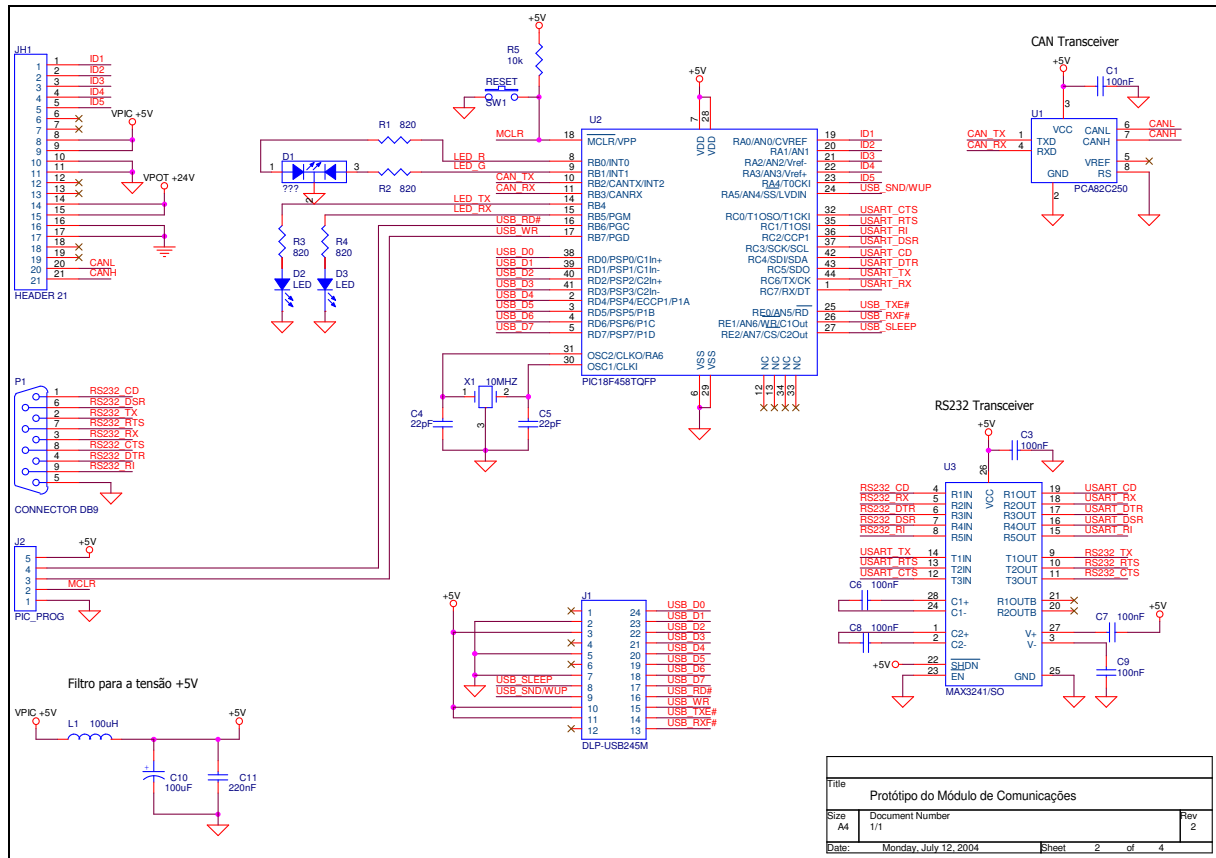


Figura 59 – Esquemático do protótipo do Módulo de Comunicações

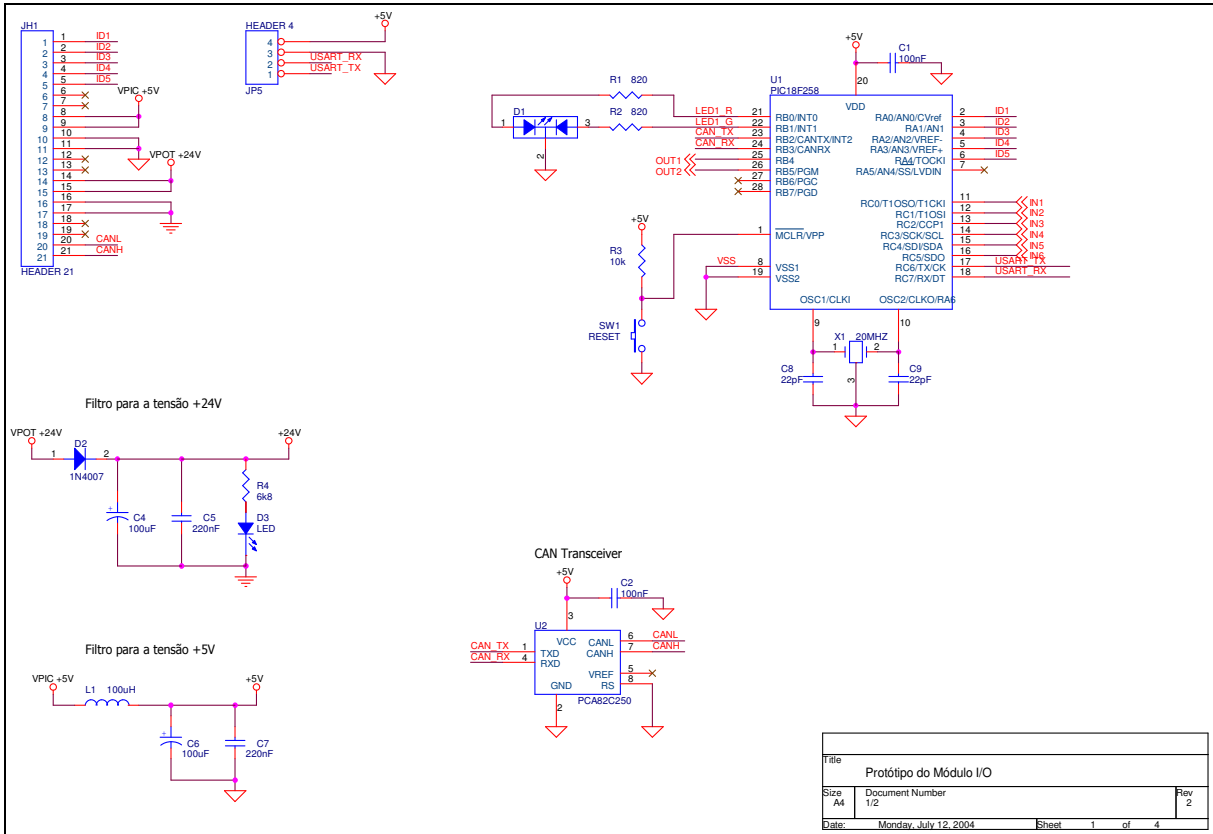


Figura 60 – Esquemático do protótipo do Módulo de I/O (parte 1)

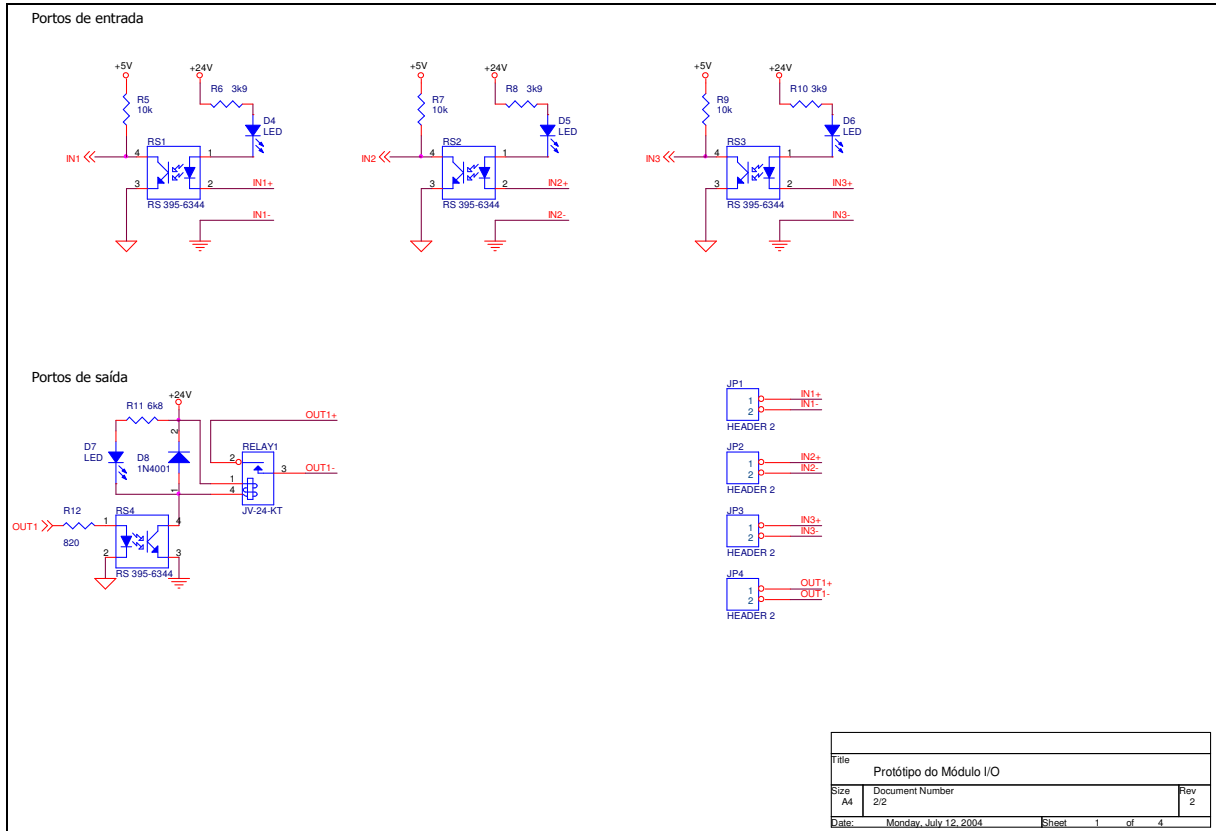


Figura 61 – Esquemático do protótipo do Módulo de I/O (parte 2)

C Listagem dos comandos de comunicação entre os módulos da ULM

Comando	Valor	Direcção	Descrição
ID_REQ	0x01	MP → Outros	Pedido de ID e tipo de módulo
ID_RESP	0x02	Outros → MP	Resposta relativa ao comando ID_REQ
CFG_REQ	0x03	MP → MC	Pedido de configuração ao MC
CFG	0x04	MP → Outros	Trama com informação relativa a configurações
CFG_ACK	0x05	Outros → MP	<i>Acknowledge</i> relativo aos comandos CFG_REQ e CFG
COMM_REQ	0x06	MP → MC	Pergunta ao MC se existem comandos externos para executar
COMM_YES	0x07	MC → MP	Resposta afirmativa do comando COMM_REQ
COMM_NO	0x08	MC → MP	Resposta negativa do comando COMM_REQ
ULM_STAT_REQ	0x09	MC → MP	Pedido do <i>status</i> da ULM
ULM_STAT_RESP	0x0A	MP → MC	Resposta relativa ao comando ULM_STAT_REQ
ULM_STAT_RESP_ACK	0x0B	MC → MP	<i>Acknowledge</i> relativo ao comando ULM_STAT_RESP
FCFG_ULM_REQ	0x0C	MC → MP	Pedido da disposição física dos módulos da ULM
FCFG_ULM_RESP	0x0D	MP → MC	Resposta relativa ao comando FCFG_ULM_REQ
FCFG_ULM_RESP_ACK	0x0E	MC → MP	<i>Acknowledge</i> relativo ao comando FCFG_ULM_RESP
HIST_DOWN_REQ	0x0F	MC → MP	Pedido para efectuar o <i>download</i> do histórico
HIST_DOWN_RESP	0x10	MP → MC	Resposta relativa ao <i>download</i> do histórico
HIST_DOWN_RESP_ACK	0x11	MC → MP	<i>Acknowledge</i> do comando HIST_DOWN_RESP
HIST_DOWN_END	0x12	MP → MC	Sinalização de fim de <i>download</i>
HIST_DOWN_END_ACK	0x13	MC → MP	<i>Acknowledge</i> relativo ao comando HIST_DOWN_END
EXT_ULM_CFG	0x14	MC → MP	Pedido ao MP para dar início à recepção de uma nova configuração da ULM
EXT_ULM_CFG_ACK	0x15	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_ULM_CFG
EXT_MP_CFG	0x16	MC → MP	Trama com informação relativa à configuração do MP
EXT_MP_CFG_ACK	0x17	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_MP_CFG
EXT_MC_CFG	0x18	MC → MP	Trama com informação relativa à configuração do MC
EXT_MC_CFG_ACK	0x19	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_MC_CFG
EXT_MIO_CFG	0x1A	MC → MP	Trama com informação relativa às configurações dos M.I/O
EXT_MIO_CFG_ACK	0x1B	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_MIO_CFG
EXT_ULM_CFG_END	0x1C	MC → MP	Sinalização de fim de configuração
EXT_ULM_CFG_END_ACK	0x1D	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_ULM_CFG_END
EXT_EVE_CFG	0x1E	MC → MP	Pedido ao MP para dar início à recepção de uma nova configuração dos eventos
EXT_EVE_CFG_ACK	0x1F	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_EVE_CFG
EVE_CFG	0x20	MC → MP	Trama com informação relativa ao eventos
EVE_CFG_ACK	0x21	MP → MC	<i>Acknowledge</i> relativo ao comando EVE_CFG
EXT_EVE_CFG_END	0x22	MC → MP	Sinalização de fim de configuração de eventos
EXT_EVE_CFG_END_ACK	0x23	MP → MC	<i>Acknowledge</i> relativo ao comando EXT_EVE_CFG_END
RUN	0x24	MC → MP	Colocar a ULM no estado normal de funcionamento
RUN_ACK	0x25	MP → MC	<i>Acknowledge</i> relativo ao comando RUN
STOP	0x26	MC → MP	Colocar a ULM em pausa para configuração
STOP_ACK	0x27	MP → MC	<i>Acknowledge</i> relativo ao comando STOP
GET_TIME	0x28	MC → MP	Obter hora actual da ULM
GET_TIME_RESP	0x29	MP → MC	Resposta relativa ao comando GET_TIME
PUT_TIME	0x2A	MC → MP	Actualizar hora da ULM
PUT_TIME_ACK	0x2B	MP → MC	<i>Acknowledge</i> relativo ao comando PUT_TIME
GET_HISTSTAT	0x2C	MC → MP	Obter o estado do histórico
GET_HISTSTAT_RESP	0x2D	MP → MC	Resposta relativa ao comando GET_HISTSTAT
MIO_IN_REQ	0x2E	MC → MP	Obter estado das ocorrências de um Módulo I/O
MIO_IN_RESP	0x2F	MP → MC	Resposta relativa ao comando MIO_IN_REQ
MIO_OUT	0x30	MC → MP	Actuação nas saídas de um Módulo I/O
MIO_OUT_ACK	0x31	MP → MC	<i>Acknowledge</i> relativo ao comando MIO_OUT
ALARM	0x32	MP → MC	Envio de um alarme
ALARM_ACK	0x33	MC → MP	<i>Acknowledge</i> relativo ao comando ALARM
GET_CFGDATE	0x34	MC → MP	Obter data da configuração actual da ULM
GET_CFGDATE_RESP	0x35	MP → MC	Reposta relativa ao comando GET_CFGDATE
CLR_ERRORS	0x36	MC → MP	Apagar <i>flag</i> de erros
CLR_ERRORS_ACK	0x37	MP → MC	<i>Acknowledge</i> relativo ao comando CLR_ERRORS
SET_OUTPORT	0x38	MC → MP	Actuação externa numa saída da ULM
SET_OUTPORT_ACK	0x39	MP → MC	<i>Acknowledge</i> relativo ao comando Set_OUTPORT

Tabela 9 – Comandos de Comunicação entre os módulos da ULM

D Listagem dos comandos de comunicação entre PC e ULM

Comando	Valor	Direcção	Descrição
ULM_STAT_REQ	0x09	PC → MC	Pedido do <i>status</i> da ULM
FCFG_ULM_REQ	0x0C	PC → MC	Pedido da disposição física dos módulos da ULM
HIST_DOWN_REQ	0x0F	PC → MC	Pedido para efecturar o <i>download</i> do histórico
EXT_ULM_CFG	0x14	PC → MC	Upload de uma nova configuração para a ULM
EXT_EVE_CFG	0x1E	PC → MC	Upload de uma nova lista de eventos para a ULM
RUN	0x24	PC → MC	Colocar a ULM no estado normal de funcionamento
STOP	0x26	PC → MC	Colocar a ULM em pausa para configuração
GET_TIME	0x28	PC → MC	Obter hora actual da ULM
PUT_TIME	0x2A	PC → MC	Actualizar hora da ULM
GET_HISTSTAT	0x2C	PC → MC	Obter o estado do histórico
GET_CFGDATE	0x34	PC → MC	Obter data da configuração actual da ULM
CLR_ERRORS	0x36	PC → MC	Apagar <i>flag</i> de erros
SET_OUTPORT	0x38	PC → MC	Actuação externa numa saída da ULM
PC_GET_ID	0x63	PC → MC	Obter o ID da ULM
PC_GET_ALARMS	0x64	PC → MC	Download dos alarmes ocorridos
PC_ACK_OK	0x60	PC ↔ MC	<i>Acknowledge</i> positivo relativo à comunicação entre o PC e Módulo de Comunicações
PC_ACK_KO	0x61	PC → MC	<i>Acknowledge</i> negativo relativo à comunicação entre o PC e Módulo de Comunicações
PC_RESP	0x62	PC ↔ MC	Trama com informação relativa aos comando executados

Tabela 10 – Comandos de Comunicação entre o PC e a ULM

Publicações do autor no âmbito do trabalho exposto na Dissertação

Ana Margarida B. M. Sargento, Bernardo Cunha, Osvaldo Rocha Pacheco, “Sistema de Monitorização para Elevadores”, Revista do Departamento de Electrónica e Telecomunicações, Universidade de Aveiro, Vol. 4, nº 3, Setembro de 2004

Lista de Acrónimos

ASCII	<i>American Standard Code for Information Interchange</i>
CAN	<i>Controller Area Network</i>
CSMA/CA	<i>Carrier Sense Multiple Access with Collision Avoidance</i>
CSMA/CD	<i>Carrier Sense Multiple Access with Collision Detection</i>
CSR	Centro de Supervisão Remoto
DC	<i>Direct Current</i>
EEPROM	<i>Electrically Erasable Programmable Read-Only Memory</i>
GPRS	<i>General Packet Radio Service</i>
GSM	<i>Global System for Mobile Communication</i>
LED	<i>Light-emitting diode</i>
NRZ	<i>Non-Return-to-Zero</i>
PSTN	<i>Public switched telephone network</i>
RAM	<i>Random access memory</i>
SMS	<i>Short Message Service</i>
SPI	<i>Serial Peripheral Interface</i>
TCP/IP	<i>Transmission Control Protocol/Internet Protocol</i>
TDMA	<i>Time Division Multiple Access</i>
ULM	Unidade Local de Monitorização
USART	<i>Universal Synchronous Asynchronous Receiver Transmitter</i>
USB	<i>Universal Serial Bus</i>

Referências

- [1] http://www.otis.com/maintenancedetail/0,1422,CL11_MID7828_RES1,00.html, OTIS – Remote Elevator Monitoring (2003 - 2006)
- [2] <http://www.advantech.com.tw/efms/appstories/12.pdf>, ADAM-5510 - Centralized Elevator Monitoring System (2003)
- [3] http://www.advantech.com.tw/eAutomation/CMScontent-apps.asp?doc_id=%7BFEA4B107-084A-49D3-921B-F49E0A45E18, ADAM-5510 - Centralized Elevator Monitoring System (2006)
- [4] <http://www.elevator-world.com/magazine/archive01/0009-001.html-ssi>, W.L. Chan, Albert T.P. So and S.K. Liu, A Cost-effective Remote Monitoring and Communication System (2003)
- [5] http://global.mitsubishielectric.com/pdf/advance/vol88/vol88_tr8.pdf, Kiyoji Kawai and Hideki Shiozaki, A Remote Inspection System for Elevators (2003)
- [6] http://global.mitsubishielectric.com/pdf/advance/vol115/115_TR4.pdf Mitsubishi Electric Advance, September 2006 (2006)
- [7] <http://www.fujitecamerica.com/cms/Modernization/Performance/ElevatorMonitoringSystem/Overview.html> Fujitec Elevator Monitoring System (EMS™) (2006)
- [8] <http://www.qameleon.com/protected/ems.htm> QarVision™ Elevator Performance Monitor (2006)
- [9] <http://www.wurtec.com/S3Analyzer.htm> Wurtec Remote elevator Monitoring (2006)
- [10] <http://www.barcodetech.com/magnetic-reader/index.htm> (2003)
- [11] http://www.avantinc.com/pc-based_id/pvc_cards.html (2003)
- [12] <http://www.aurorabarcode.com/> (2003)
- [13] <http://www.didya.com/> (2003)
- [14] <http://www.omron.com/card/cardreader/> (2003)
- [15] <http://www.ti.com/tiris/> (2003)
- [16] <http://www.ibutton.com/ibuttons/> (2003)
- [17] “Industrial automation and control”, Foundation fieldbus, 1993 (2003)
- [18] Perry Green, Bill Moss, and Mike Miclot, “How fast is your network?”, <http://www.pyrasol.com/DataCom/byline.htm>, 2000. (2003)
- [19] http://icat.snu.ac.kr:3333/ww/pdf/ww_2002_8.pdf (2003)
- [20] <http://www.semiconductors.philips.com/buses/i2c/facts/index.html#whatis> (2003)
- [21] <http://itcofe.web.cern.ch/itcofe/Services/Fieldbus/Profibus/welcome.html> (2003)
- [22] http://www.profibus.com/imperia/md/content/pisc/technicaldescription/4002_Draft_April_2002.pdf (2003)
- [23] <http://www.worldfip.org/SHWebClass.ASP?WCI=ShowDoc&DocID=433> (2003)
- [24] <http://itcofe.web.cern.ch/itcofe/Services/Fieldbus/WorldFIP/welcome.html> (2003)
- [25] <http://www.can-cia.org>, CAN in Automation homepage (2003)
- [26] <http://www.catsemi.com/datasheets/memory.html> (2003)
- [27] <http://uk.farnell.com/jsp/home/homepage.jsp> (2003)
- [28] http://www.mouser.com/index.cfm?handler=listproduct&categoryid=95&lst_pcode=57901 (2003)
- [29] <http://rswww.com> (2003)
- [30] <http://www.chipcatalog.com/Cat/512.htm> (2003)

- [31] http://www.sst.com/products.xhtml/serial_flash/25/ (2003)
- [32] <http://www.ftdichip.com/Documents/ds245b14.pdf> (2003)
- [33] Alexandre Santos, Ana Sargento, Pedro Leite, Ricardo Marau - Developing APIs and Device Drivers for CAN Based Embedded System
- [34] <http://pdfserv.maxim-ic.com/en/ds/DS1990A-F3-DS1990A-F5.pdf> (2003)