

Sistema de Monitorização para Elevadores

Ana Margarida B. M. Sargento, Bernardo Cunha e Osvaldo da Rocha Pacheco

Resumo – Neste artigo é descrito o desenvolvimento de um Sistema de Monitorização para Elevadores, efectuado no âmbito da disciplina de Projecto do 5º ano do curso de Engenharia Electrónica e Telecomunicações da Universidade de Aveiro. Este projecto enquadra-se num conjunto de três projectos baseados em propostas de desenvolvimento provenientes de uma empresa exterior à Universidade, de nome LifTech, a qual desenvolve, fabrica e comercializa Sistemas de Comando para Elevadores.

O sistema desenvolvido é adaptável a qualquer tipo de elevador, tem capacidade para monitorizar um conjunto de sinais e para disponibilizar comandos actuáveis externamente. O sistema permite ainda fazer o registo de assistências técnicas. A monitorização propriamente dita é efectuada a partir de um Centro de Supervisão Remoto que está em contacto com vários sistemas de elevadores.

Abstract – In this paper we present the final result of the development of a general purpose Lift Monitoring System, which has been held in the context of a Project discipline in the last year of the graduated course in Engenharia Electrónica e Telecomunicações da Universidade de Aveiro. This project is part of a three project proposals block brought to us by a company called LifTech. This company is responsible for the production and commercialisation of Lift Command Systems.

The developed system can be adapted to any kind of lift, is able to monitor a significant amount of signals and provides a group of command signals that can be actuated on request. Technical assistance by authorized personnel can also be monitored. Remote monitoring is provided by a Remote Supervising Centre, which can be in contact with several distinct lift systems.

I. INTRODUÇÃO

O sistema desenvolvido 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.

Com o objectivo de desenvolver uma solução actual e competitiva foi efectuado um levantamento dos sistemas de monitorização de elevadores que existem no mercado. De entre os sistemas encontrados destacam-se os sistemas da OTIS [1], da ADVANTECH [2], da ELEVATOR WOLD [3] e da MITSUBISHI [4]. Alguns destes sistemas foram desenvolvidos exclusivamente para integração nos elevadores da própria empresa, como o da OTIS, enquanto

outros de destinam a ser integrados na generalidade dos elevadores já existentes. Alguns destes sistemas apresentam algumas funcionalidades de destaque como o facto de permitirem monitorizar a deterioração de componentes, prever anomalias, detectar ocorrências com duração parametrizável, gerar relatórios de ocorrências, permitirem a configuração remota do sistema, permitirem a comunicação por voz com passageiros presos dentro do elevador, possuírem uma bateria que assegura o funcionamento do sistema (por um curto período de tempo) em caso de falha de energia ou ainda o facto de assentarem numa arquitectura modular, facilitando a manutenção e a substituição de módulos danificados e permitindo uma fácil expansão do sistema.

O sistema desenvolvido visa reunir as funcionalidades encontradas nos demais sistemas existentes.

A arquitectura da solução adoptada baseia-se numa rede de Unidades Locais de Monitorização (ULMs) inteligentes, responsáveis pela monitorização do elevador ou conjunto de elevadores pertencentes a um mesmo edifício, e um Centro de Supervisão Remoto (CSR) constituído por um PC remoto. O *software* do PC localizado no CSR permite receber alarmes provenientes de múltiplas unidades e manter toda a informação relacionada com as mesmas. A comunicação entre ambos é efectuada através de rede fixa comutada ou rede GSM.

Ao longo do artigo, são descritas as várias fases de desenvolvimento do projecto. Inicialmente é apresentada uma secção inteiramente dedicada à descrição da Unidade Local de Monitorização (secção II), que engloba a solução adoptada em termos de arquitectura e funcionalidade, o protocolo de comunicação entre os módulos da ULM (Unidade Local de Monitorização) e a estrutura do *software* desenvolvido. De seguida é apresentada uma secção relativa ao Centro de Supervisão Remoto (CSR) (secção III), onde se descreve o protocolo de comunicação entre PC e ULM e a estrutura da base de dados que deve constar no PC do CSR. Na secção IV é então apresentada a aplicação de demonstração desenvolvida. Finalmente, na secção V são apresentadas as conclusões do trabalho.

II. UNIDADE LOCAL DE MONITORIZAÇÃO (ULM)

A. Arquitectura da Solução Encontrada

Cada ULM é responsável pela monitorização e registo local de eventos e tem capacidade de reportar alarmes para o Centro de Supervisão Remoto, actuar em saídas a pedido

do mesmo 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 Remoto, é enviado para o mesmo. Cada evento é acompanhado da data e hora da ocorrência, com resolução temporal de 250ms.

A sua implementação é baseada numa arquitectura distribuída que adopta um modelo funcional do tipo *master/slave*, podendo o sistema local (ULM) integrar:

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

interligados por uma *Rack* que suporta até 8 módulos.

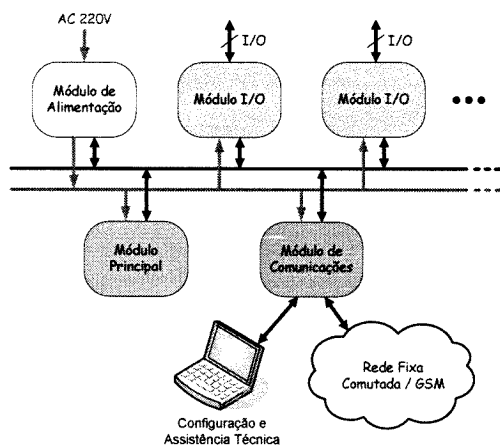


Figura 1 – Diagrama de blocos da Unidade Local de Monitorização

Os módulos comunicam através de um barramento CAN e possuem inteligência local baseada em microcontroladores da *Microchip* [6]. No Módulo de Comunicações é utilizado o PIC18F458 enquanto os restantes módulos se baseiam no PIC18F258.

A *Rack* fornece aos módulos uma identificação *hard-wired*, dependente da posição onde cada um se encontra, as alimentações desacopladas de 5V e 24V e as linhas de comunicação CAN [5].

O Módulo Principal é responsável por coordenar toda a ULM, assegurando a gestão de eventos, o armazenamento da configuração da ULM e do histórico de eventos e ainda o registo de assistências técnicas (Figura 2).

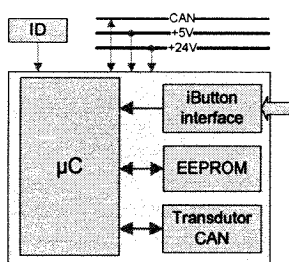


Figura 2 – Esquema do Módulo Principal

Os eventos a detectar podem ser:

- O início/fim de um período de manutenção local por parte de um técnico especializado;
- A ocorrência de erros na ULM;
- A ocupação da memória de histórico acima de um *threshold* predefinido;
- A detecção de ocorrências associadas a cada porto de entrada.

Cada evento pode conduzir ao registo de ocorrência do mesmo no histórico, ao envio de um alarme para o CSR, à actuação numa saída de um módulo I/O ou a qualquer combinação destas acções.

O armazenamento de configurações e histórico de eventos é efectuado numa memória externa não volátil, de tecnologia EEPROM.

O sistema de identificação do técnico especializado, usado para efectuar a detecção de início e fim das assistências técnicas em cada ULM é do tipo *iButton* [7] – dispositivo de armazenamento de dados, robusto, que actua como um registo de identificação electrónica, possuindo um número único associado.

O Módulo de Comunicações permite a comunicação com o exterior, ou seja, a comunicação entre a Unidade Local de Monitorização e o Centro de Supervisão e entre a Unidade Local de Monitorização e o PC de Assistência. A referida comunicação é efectuada a partir de uma interface USB para ligação a um PC de assistência e através de uma interface RS232 para ligação a um *modem* (Figura 3).

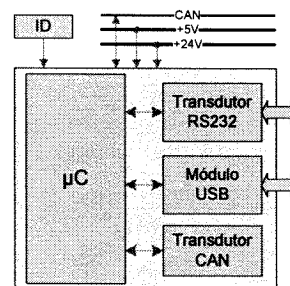


Figura 3 – Esquema do Módulo de Comunicações

O Módulo de Comunicações tem como função tornar o sistema independente do tipo de modem (GSM ou PSTN).

Finalmente, cada Módulo I/O é responsável pela interface física dos sinais a monitorizar, através de 8 entradas, isoladas opticamente para interface com contactos livres de tensão, e 2 saídas por relés (Figura 4).

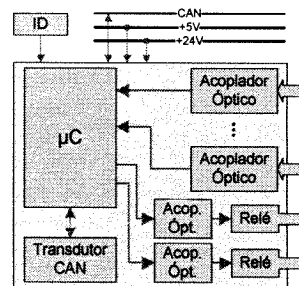


Figura 4 – Esquema do Módulo I/O

Cada um destes módulos é ainda responsável pela detecção de ocorrências com duração parametrizável de 250ms a 1 minuto e resolução temporal de 250ms.

B. Protocolo de Comunicação entre os Módulos

O Módulo Principal actua como *master* da comunicação com os restantes módulos da ULM, 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.

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.

C. Software do Módulo Principal

C.1. Algoritmo

A execução das várias tarefas a cargo deste módulo obedece à máquina de estados apresentada na figura 5.

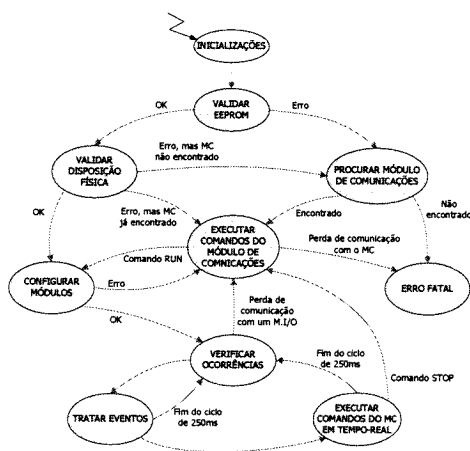


Figura 5 – Diagrama de estados do *software* do Módulo Principal

Na fase de arranque do Módulo Principal são efectuadas as inicializações dos vários periféricos, tanto internos como externos ao microcontrolador (inicialização dos portos, módulo da USART, módulo CAN, leitor de iButtons, EEPROM, tick do sistema e variáveis globais).

Seguidamente é efectuada uma validação do conteúdo da EEPROM. No caso de se verificar alguma incoerência da informação contida na EEPROM, a ULM terá de aguardar a intervenção de um técnico, pelo que tem necessidade de determinar a posição do Módulo de Comunicações para poder comunicar com o exterior.

Caso a informação contida na EEPROM seja válida, é efectuado um reconhecimento da disposição física dos restantes módulos. Se se verificar haver alteração,

relativamente à disposição física anterior, é necessária a intervenção de um técnico, pelo que o Módulo Principal fica a aguardar comandos do Módulo de Comunicações, passando para o estado correspondente.

Considerando a disposição física validada, é efectuada a (re)configuração dos módulos. Caso ocorra algum erro durante este procedimento o módulo passa para o estado *EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES*, aguardando intervenção de um técnico.

Após o reconhecimento da disposição física é efectuada a detecção de ocorrências relacionadas com o estado dos portos de entrada, detecção de iButton, percentagem de ocupação do histórico e ocorrência de erros (que não impeçam o funcionamento da ULM mas que sejam susceptíveis de gerar eventos). Os eventos disparados são então colocados numa fila de espera para posterior tratamento.

No estado *TRATAR EVENTOS* são processados os eventos que estão na lista de espera, ou seja, é levada a cabo a acção associada ao evento (actuar numa saída, registar evento, enviar alarme), tendo o cuidado de não ultrapassar o tempo de ciclo de amostragem do sistema. 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 o tempo de ciclo não se esgote, passa-se para o estado *EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES EM TEMPO-REAL*.

Na execução deste último estado são atendidos comandos, que não implicam a paragem da amostragem do sistema, vindos do Módulo de Comunicações.

O estado *EXECUTAR COMANDOS DO MÓDULO DE COMUNICAÇÕES* é em tudo semelhante ao estado descrito anteriormente, com a diferença de ser um estado bloqueante.

O estado *PROCURAR MÓDULO DE COMUNICAÇÕES* tem o objectivo de informar o Módulo Principal do ID do Módulo de Comunicações (e *vice-versa*) para que a comunicação entre a ULM e o exterior se possa processar, apesar da impossibilidade de funcionamento normal da mesma.

O estado *ERRO FATAL* é atingido quando existe uma falha de comunicação entre o Módulo Principal e o Módulo de Comunicações. Este estado é sinalizado com o LED multicolor a piscar vermelho à frequência de 2Hz.

C.2. Detalhes de Implementação

DEVICE DRIVER PARA INTERFACE COM A EEPROM

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, na EEPROM, da informação presente nos *buffers*.

DEVICE DRIVER PARA INTERFACE COM O MÓDULO CAN

Foi desenvolvida uma camada de *software* capaz de gerir tanto a recepção como a transmissão de tramas CAN

usando interrupções. Deste modo o código de transmissão e recepção de mensagens não é bloqueante [8].

DEVICE DRIVER PARA INTERFACE COM O IBUTTON

A interface com o *iButton* foi feita implementado em *software* o protocolo de comunicação descrito no *datasheet* do próprio componente [9].

D. Software do Módulo de Comunicações

D.1. Algoritmo

O Módulo de Comunicações actua como *gateway* entre o Módulo Principal e o CSR ou um PC local. O funcionamento deste módulo tem que obedecer a duas características difíceis de conciliar: a comunicação a ritmo elevado com o Módulo Principal, simultaneamente com a comunicação, a ritmo mais lento, com o *modem*. A solução adoptada passa por tratar os comandos de mais alta prioridade (mensagens CAN), vindos do Módulo Principal, numa rotina de interrupção.

O algoritmo de controlo associado ao Módulo de Comunicações engloba a execução de 3 processos: processamento de comandos vindos do CSR ou PC local quando está estabelecida uma ligação com um desses sistemas; envio de alarmes ao CSR e execução de comandos do CSR recebidos por SMS.

D.2. Detalhes de Implementação

COMUNICAÇÃO VIA USB

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* [10].

E. Software de cada Módulo I/O

E.1. Algoritmo

Os módulos I/O são responsáveis pela validação das ocorrências associadas aos respectivos portos de entrada e pela actuação nas saídas (sob a forma de comando). Os processos relacionados com a amostragem e processamento dos sinais de entrada são executados com um período de 10ms. O tempo que resta é usado para executar comandos do Módulo Principal, caso tenham sido recebidos.

Os sinais de entrada podem estar ligados a relés ou a dispositivos de comutação com actuação mecânica que têm associado ruído de bouncing derivado aos seus contactos. Para eliminar o ruído decorrente deste facto as entradas são amostradas a uma frequência de 100Hz, sendo a validação de cada estado feita apenas quando 25 amostras consecutivas assumem o mesmo valor. Deste modo só existe mudança de estado quando o sinal amostrado se mantém estável durante 250ms.

III. CENTRO DE SUPERVISÃO REMOTO (CSR)

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.

Do ponto de vista funcional, uma (re)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.

A. Protocolo de Comunicação entre PC e ULM

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. O *handshaking* praticado envolve, no entanto, a recepção de uma dupla confirmação. 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 6).

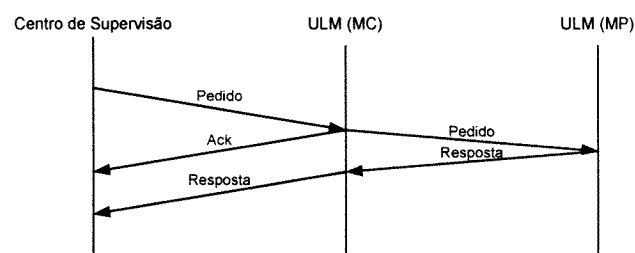


Figura 6 – Sequência de mensagens trocadas quando é efectuado um pedido de informação

As mensagens oriundas do CSR, que não constituem um pedido de informação, não recebem uma resposta mas apenas uma confirmação de comando transmitido com sucesso.

Cada mensagem trocada entre a ULM e o PC (quer seja do Centro de Supervisão quer seja de Assistência) é constituída por um carácter de início de trama, um comando (um byte), dados de tamanho variável (eventualmente nulo), o *checksum* da mensagem e um carácter de fim de trama. O *checksum* é aplicado ao conjunto (comando + dados) e é utilizado para detecção de erros na transmissão. Para garantir que os caracteres de início e fim de trama só surgem nessas situações (como

delimitadores da trama), toda a trama, excluindo esses caracteres, é codificada em ASCII.

A.1. Comunicação por GSM ou Rede Fixa Comutada

No modelo de comunicação utilizado 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, a transmissão de determinados fluxos de informação, como o *download* do histórico ou o envio de configuração, implica a transmissão de uma sequência longa de tramas, implicando *handshaking* para cada trama. Utilizando o envio de mensagens SMS sobre GSM, o tempo que medeia o envio de cada mensagem e a recepção da respectiva resposta é da ordem de alguns segundos levando a que a transmissão de sequências longas de pacotes seja 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, consequentemente, do Módulo Principal.

A utilização de SMS será assim apenas viável para uma ULM cuja configuração e *download* do histórico sejam efectuados localmente e que utilize a ligação GSM apenas para o envio de alarmes e recepção de comandos de actuação nas saídas. Estas comunicações envolvem 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 devem ser assumidos como limitações do sistema. Uma alternativa a esta tecnologia será a utilização GPRS, alternativa esta que não chegou a ser estudada no âmbito do projecto.

A utilização de rede fixa comutada permite o estabelecimento de uma chamada de dados que fica disponível aos 2 intervenientes até ser terminada por um deles. Assim é possível efectuar a transmissão de fluxos de informação com diferentes características, incluindo sequências longas de tramas.

B. Estrutura da Base de Dados

A base de dados armazenada no CSR 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 os seguintes campos:

- O número de telefone do elevador (modem) ⇒ Campo obrigatório;
- O número de telefone do Centro de Supervisão Remoto ⇒ Campo obrigatório;
- A morada do elevador de onde provém o alarme;

- O número do contrato;
- O número de telefone da empresa de manutenção;
- O número de telefone do porteiro do prédio;
- O tipo de modem da ULM ⇒ Campo obrigatório;
- O número de tentativas para estabelecer chamada entre o CSR e a ULM ⇒ Campo obrigatório;
- A percentagem de ocupação do histórico à qual deve ser enviado um alerta para a central;
- Uma cópia, datada, da configuração actual do sistema ⇒ Campo obrigatório;
- Uma cópia, datada, de cada uma das configurações que o sistema já teve ⇒ Campo obrigatório;
- Uma cópia do estado actual das saídas.

IV. APLICAÇÃO DE DEMONSTRAÇÃO

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, mais tarde, o *software* de monitorização e configuração. Embora não tendo havido possibilidade de desenvolver o *software* do CSR, surgiu a necessidade de ter um *software* de demonstração que permitisse por um lado fazer um *debug* mais profundo ao *software* da ULM, e por outro permitisse demonstrar as funcionalidades da ULM concebida. O *software* desenvolvido não explora profundamente as potencialidades da ULM, permitindo apenas ao utilizador um conjunto de actuações simples que emulam as transferências de informação que fazem parte do protocolo de comunicação implementado.

Este *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 poder ser recuperada cada vez que se utilizar a aplicação.

O *software* de demonstração permite ao utilizador configurar uma ULM e provocar as ocorrências previamente configuradas de modo a fazer despoletar os eventos (Figura 7). Finalmente, permite efectuar o *download* do histórico e analisar o seu conteúdo de forma a determinar se tudo se processou como era esperado.

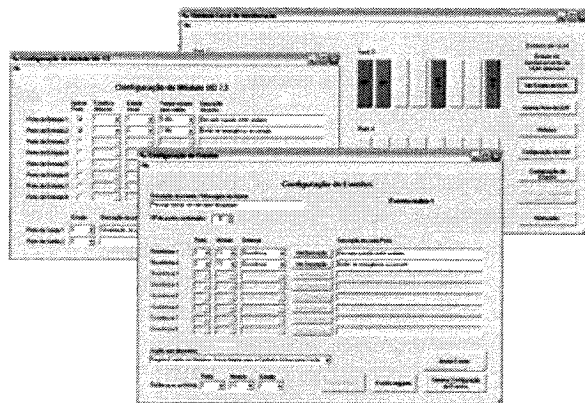


Figura 7 – Apresentação da Aplicação de Demonstração

V. CONCLUSÕES

Neste artigo são descritas as várias fases de desenvolvimento de um sistema de monitorização remota para elevadores bem como a sua arquitectura final. Este sistema permite a monitorização e registo de eventos, a actuação remota de comandos e o registo de assistências técnicas.

O sistema projectado baseia-se na existência de unidades de monitorização locais (ULMs), que fazem a monitorização directa dos sinais e a gestão dos eventos, e na existência de um Centro de Supervisão Remoto, o qual possui *software* que permite receber alarmes provenientes das ULMs e manter, numa base de dados, toda a informação relacionada com as mesmas. A comunicação entre ambos pode ser efectuada através de rede fixa comutada ou rede GSM. Foi no entanto possível concluir que a escolha da rede GSM acarreta algumas limitações relativamente às características da informação que pode ser transmitida, uma vez que não permite a transmissão de sequências longas de tramas.

O *software* desenvolvido para a ULM possui um elevada redundância ao nível das comunicações entre os módulos o que confere ao sistema uma elevada tolerância a falhas e esse nível. As funcionalidades disponibilizadas pela ULM permitem ainda uma elevada flexibilidade no desenvolvimento do *software* do CSR, uma vez que cada ULM pode responder aos comandos provenientes do CSR sem que a sua principal função de monitorização seja afectada.

O sistema desenvolvido é totalmente parametrizável pelo que pode ser utilizado noutras aplicações que não os elevadores. Para esta possível aplicação prática contribui o facto da resolução temporal do sistema ser de 250ms, permitindo uma maior reactividade do que a que seria estritamente necessária para um sistema de elevadores.

REFERÊNCIAS

- [1] http://www.otis.com/maintenancedetail/0,1422,CLH_MID7828_RES1,00.html, OTIS – Remote Elevator Monitoring
- [2] <http://www.advantech.com.tw/efms/appstories/12.pdf>, ADAM-5510 - Centralized Elevator Monitoring System
- [3] <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
- [4] http://global.mitsubishielectric.com/pdf/advance/vol88/vol88_tr8.pdf, Kiyoji Kawai and Hideki Shiozaki, A Remote Inspection System for Elevators
- [5] <http://www.can-cia.org>, CAN in Automation homepage
- [6] MicroChip PIC18FXX8 Datasheet
- [7] <http://www.ibutton.com/ibuttons/>
- [8] Alexandre Santos, Ana Sargento, Pedro Leite, Ricardo Marau - Developing APIs and Device Drivers for CAN Based Embedded System
- [9] <http://pdfserv.maxim-ic.com/en/ds/DS1990A-F3-DS1990A-F5.pdf>
- [10] <http://www.ftdichip.com/Documents/ds245b14.pdf>