Universidade da Beira Interior

Departamento de Engenharia Electromecânica

UBI
Covilhã
Portugal

CONCEPÇÃO DE UM MODELO DE SISTEMA PARA A GESTÃO DE INTRUSÃO EM ESPAÇOS RESIDENCIAIS

Rui Jorge da Fonseca Filipe Vaz

Dissertação para Obtenção de Grau de Mestre do Segundo Ciclo em Engenharia Electrotécnica

Covilhã, Outubro de 2010

Dissertação realizada sob a orientação do

Prof. Doutor Bruno Jorge Ferreira Ribeiro
Departamento de Engenharia Electromecânica
Universidade da Beira Interior

“O único homem que está livre de erros
é aquele que não arrisca acertar”
(Albert Einstein)

Dedico este trabalho aos meus pais, irmã e namorada pelo incentivo e paciência que tiveram para comigo.

Também devo lembrar e agradecer todos os amigos que de uma forma ou de outra estiveram sempre presentes.

CONCEPÇÃO DE UM MODELO DE SISTEMA PARA A GESTÃO DE INTRUSÃO EM ESPAÇOS RESIDENCIAIS

Resumo

A Domótica existe para simplificar a vida diária dos utilizadores, satisfazendo três necessidades básicas: conforto, segurança e comunicações. Trata-se de uma tecnologia em pleno desenvolvimento cuja evolução passará certamente pela integração sistemática de novos serviços e pela generalização das aplicações a todos os segmentos do mercado imobiliário. A banalização da Domótica está em curso e, se esta realidade não é suficientemente visível, é porque existe uma grande falta de informação sobre o assunto. Sendo um mercado em expansão dado a exageros, convém esclarecer quem está interessado em investir em Domótica, para que as expectativas, quer de quem as vende, quer de quem as compra, não sejam defraudadas.

Com o desenvolvimento desta dissertação, pretende-se demonstrar um modelo de sistema de gestão de intrusão baseado num sistema automatizado, com a capacidade de estabelecer ligações lógicas entre dispositivos, e acima de tudo suportar a definição de "cenários" de funcionamento.

Os modelos aqui apresentados são baseados através da norma de comunicações CANopen. Para cada um dos dispositivos apresenta-se o respectivo perfil funcional que identifica a sua função específica, (sensores que podem ser utilizados e respectiva central de comando com modos e zonas de funcionamento bem definidos) onde estão definidos todos os parâmetros relativos à sua configuração.

Palavras-Chave: Domótica, Sistema de gestão de intrusão; Sistema automatizado; sensores; central de comando; CANopen.

Abstract

The Home automation exists to simplify daily life of users, satisfying three basic needs: comfort, security and communications. This is a rapidly developing technology whose evolution will surely by the systematic integration of new services and the growth of applications to all segments of the housing market. The banality of Home Automation is underway, and if this reality is not visible enough, because there is a lack of information on the subject. Being a growing market given to exaggeration, it is clear who is interested in investing in Domotics to expectations, both of who sells or who buys them, are not disappointed.

With the development of this thesis is intended to demonstrate a model of intrusion management system based on an automated system, with the ability to establish logical connections between devices, and above all support the definition of "scenarios" of operation.

The models presented here are based communications via the CANopen standard. For each device presents its functional profile that identifies their specific function (sensors that can be used and its control unit with modes and well-defined areas of operation) which defines all parameters relating to its setting.

Keywords: Home automation, intrusion management system, automated system, sensors, electronic control unit, CANopen.

Agradecimentos

Ao longo do meu trabalho, alguns foram os que contribuíram com a sua ajuda e motivação, sem as quais o presente trabalho nao teria sido possível.

A realização desta Dissertação de Mestrado contou em primeiro lugar com a preciosa ajuda e orientação do Professor Doutor Bruno Ribeiro, que com a sua dedicação, disponibilidade, enorme experiência, constante motivação, numerosas sugestões e críticas, foram uma mais valia para que esta dissertação , marco importante na minha vida académica, se tornasse uma realidade. Por tudo isto, Professor – Bem-haja.

Quero também agradecer ao Departamento de Engenharia Electromecânica da Faculdade de Engenharias Da Universidade da Beira Interior, pela cedência da sala onde desenvolvi o meu trabalho.

A todos aqueles que de alguma forma contribuiram para a realização deste trabalho, quero deixar aqui os meus mais sinceros agradecimentos.

Finalmente, quero agradecer à minha família, namorada e amigos, o apoio e motivação que sempre me deram desde a primeira hora e por terem compreendido os vários momentos em que nao pude estar presente.

Covilhã, Outubro de 2010

Rui Jorge da Fonseca Filipe Vaz

1 Índice

2 Índice de Figuras

3 Índice de Tabelas

GLOSSÁRIO

AC Acceptance Code

AM *Acceptance Mask*

BSP *Bit Stream Processor*

BTL *Bit Timing Logic*

CAL *CAN Application Layer*

CAN *Controller Area Network*

CiA *CAN in Automation*

CMS *CAN-based Message Specification*

COB *Communication Object*

CRC *Cyclic Redundancy Check*

DLC *Data Lenght Code*

DU *Data Unit*

EIB *European Instalation Bus*

EDS *Electronic Data Sheet*

EHS *European Home Systems*

EML *Error Management Logic*

EOF *End of Frame*

IDE *Identifier Extension*

ISO *International Standards Organization*

LLC *Logical Link Control*

MAC *Medium Acess Control*

MDI *Medium Dependent Interface*

NBR *Nominal Bit Rate*

NBT *Nominal Bit Time*

OD *Object Dictionary*

OSI *Open Systems Interconnection*

PCI *Protocol Control Information*

PDO *Process Data Object*

PDU *Protocol Data Unit*

PLS *Physical Signaling*

PMA *Physical Medium Attachment*

RFD *Reduced Function Device*

RPDO *Receive Process Data Object*

RTR *Remote Transmition Request*

SCI Subunidade Controladora Integradora

SDO *Service Data Object*

SDU *Service Data Unit*

SOF *Start of Frame*

SPI *Serial Peripheral Interface*

SYNC *Synchronization Object*

TCL *Transceiver Control Logic*

TPDO *Transmit Process Data Object*

TQ Time Quantum

UC Unidade Controladora

UIS Unidade Integradora Supervisora

X-10 Protocolo baseado em linha eléctrica

1 INTRODUÇÃO

1.1 Enquadramento e motivação

A Domótica, junção da palavra *Domus* e Robótica, consiste no uso de uma tecnologia recente que abrange em simultâneo o uso da electricidade, electrónica e informática tendo em vista a gestão técnica de edifícios. Este conceito proporciona soluções ao nível do conforto, segurança e economia, podendo efectuar a maior parte das actividades de rotina humana.

Na base da Domótica, está sempre associada a palavra segurança. O automatismo mais simples para uma casa é o alarme, que, ao longo dos últimos anos, tem incorporado uma série de funções e *interfaces* que lhe permitem interagir com vários outros automatismos. Face à necessidade de soluções da Domótica, criada pelo ramo imobiliário, as empresas de segurança estão finalmente a valorizar este mercado e mantêm-se particularmente atentas à evolução natural dos sistemas electrónicos de segurança para soluções técnicas integradas de valor acrescentado.

O modelo de sistema de intrusão referido neste trabalho não é mais que uma plataforma de suporte já existente, que utiliza norma de comunicações CANopen. Do ponto de vista prático, este sistema é conhecido pela sua simplicidade e capacidade de definição de cenários de funcionamento, o que oferece uma grande flexibilidade de adaptação às necessidades dos utilizadores.

1.2 Objectivos

O objectivo deste trabalho é conceber um modelo de sistema para a gestão de intrusão em espaços residenciais. Com a concepção do modelo aqui estruturado, serão apresentados cenários de funcionamento, os respectivos dicionários de objectos , tendo em vista os diferentes parâmetros de configuração de cada um dos dispositivos. Serão também estruturados os respectivos *electronic data sheets,* necessários para descrever todos os objectos que um dispositivo implementa.

1.3 Estrutura da dissertação

Esta dissertação encontra-se estruturada em 4 capítulos, bibliografia e referência a anexos, da seguinte forma:

1ª parte - correspondente ao capítulo 2, apresenta-se uma definição global de um sistema de intrusão, dando exemplos dos mais variados e importantes sensores utilizados em caso de emergência.

2ª parte - é constituida pelos capítulos 3 e 4. Nestes capítulos, é abordada a concepção do modelo de sistemas de gestão de intrusão. São apresentados esquemas correspondentes ao modelo utilizado, assim como as respectivas funções, enquanto que no capítulo 4 se trata de definir o Dicionário de Objectos e o respectivo *Electronic Data Sheets*.

Na terceira parte da dissertação, que corresponde ao capítulo 5, descrevem-se os detalhes de funcionamento do sistema, incluíndo as unidades de ensaio, harware e software de configuração com a rede CAN.

Por fim apresentam-se as conclusões a que se chegou com o desenvolvimento deste trabalho, assim como algumas sugestões para trabalho futuro.

No sentido de não transformar a leitura da dissertação cansativa, foi colocado em anexo as diferentes tecnologias utilizadas de comunicação, e as respectivas tabelas dos dicionários de objectos e um exemplo de uma *electronic data sheet*. É ainda apresentado nos CD´s a respectiva configuração do controlador MCP2510, e ainda as respectivas *electronic data sheet* dos restantes dispositivos.

2 DOMÓTICA E SEGURANÇA ELECTRÓNICA

2.1 Sistema de Intrusão

Hoje, um sistema de segurança, além da detecção de intrusão, pode perfeitamente integrar a função de controlo de acessos de uma ou mais portas, com cartão ou etiquetas de proximidade, detecção de incêndio, detecção de fugas de gás, corte automático de gás e de água em caso de alarme técnico, accionamento automático de circuitos de iluminação (quer por programação horária quer por detecão de presença coincidente com falta de luz), accionamento de aparelhos (termoacumuladores, bombas de água, radiadores eléctricos, estores motorizados, etc).

O automatismo mais simples e conhecido do público em geral é o sistema de alarme, que tem, na sua programação e no seu desempenho, uma série de funções especificamente estudadas para reagir a informações transmitidas pelos seus periféricos, sejam eles contactos magnéticos de porta, detectores de movimento, detectores de quebra de vidro ou outros. Esta reacção prevê várias formas de assinal uma occorència, que podem ir desde o disparo automático de uma sirene exterior, ao envio de uma mensagem de voz pré-programada para um ou mais números de telefone ou uma mensagem de dados oara uma central de alarme.

Para interagir com o alarme, o utilizador dispõe de um teclado com uma série de menus intuitivos, que permitem executar as operações normais de uma utilização do sistema. Para o cliente, a evolução destes sistemas electrónicos tornou executável determinadas aplicações, bem como tornou acessíveis os seus consequentes benefícios, que, de outra forma quer por uma questão de custos, quer por oportunidade de instalação, estariam totalmente fora de questão. A banalização do uso de computadores e telemóveis no dia-a-dia tornou acessíveis para o comum dos utilizadores, determinadas soluções electrónicas de uma certa complexidade e de um curto grau de sofisticação.

Por exemplo, nas vivendas, passou a fazer sentido pensar-se num ou mais circuitos de iluminação de segurança, comandados pelo sistema de alarme, conforme as situações. Com a instalação de uma electroválvula de gás e outra de água como actuadores dos alarmes técnicos, cria-se o princípio de reacção/acção no sistema de segurança: além da detecção (reacção), passa-se também a actuar (acção) em caso de necessidade, cortando o gás ou

água, conforme o necessário. O mesmo se passa ao dotar-se determinadados detectores de movimento com dupla funcionalidade: de segurança (se o alarme estiver ligado) e de detecção de presença para accionamento automático de uma luz ou de um grupo de luzes (se o nível de luminosidade não for suficiente e se for detectado movimento no local).

O desenvolvimento das telecomunicações, móveis ou fixas, vieram também proporcionar novas formas de interacção com os sistemas de alarme e, consequentemente, com os restantes sistemas integrados. Assim, passou a ser banal o uso de um telemóvel para controlar o ligar ou desligar um alarme remotamente, assim como para se accionar uma saída do alarme com vista à execução de uma função complementar. Sistemas que acendem luzes assim que o dia começa a escurecer proporcionam conforto, mas também segurança, uma vez que fazem crer que a casa está ocupada. Se não se quiser deixar nenhuma margem de dúvida na mente do ladrão, poder-se-á mesmo instalar um sistema de simulação de presença, pois através desta tecnologia, poderá fazer-se uma simulação de luzes e estores, como se estivesse gente em casa.

2.2 Sensores utilizados num sistema de intrusão

2.2.1 Contactos magnéticos

Os sensores de portas e janelas mais comuns são os contactos magnéticos (duas peças) que indicam se a porta ou a janela está aberta oufechada. Uma parte destes contactos é instalada na porta ou janela e a outra parte na ombreira da porta ou janela. Quando a porta está fechada, os dois elementos (ímanes) estão alinhados.
Quando o alarme está ligado e os contactos alinhados, não há interrupção da ligação, pois existe um curto-circuito entre os dois contactos), pelo que o alarme se mantém em silêncio. Quando esta ligação é interrompida, o alarme toca.

2.2.2 Contactos magnéticos balanceados

Devido à facilidade em sabotar um contacto magnético com um simples íman, existem os contactos magnéticos balanceados, que além do simples contacto entre dois magnetos, têm, no seu interior, um contacto aberto que se fechará, caso se aproxime um campo magnético externo (aproximação de um íman). Sempre que uma porta esteja fechada, o campo magnético entre os dois ímanes dos contactos está fechado e o segundo está aberto. Quando a porta se abre ou se aproxima um campo magnetico externo, o segundo contacto fecha-se sobre um dos magnetos, gerando um sinal de alarme.

2.2.3 Detectores de quebra de vidro

Os sensores de vibração e de choque actuam apenas se o intruso partir um vidro para entrar (em vez de abrir a janela). Os sensores de pressão são os mais acessíveis a nível monetário e instalam-se colando-se directamente no vidro a proteger. Para controlar várias janelas, é possível escolher sensores que detectam o som de vidro a partir. Estes sofisticados sensores, que geralmente incluem até um microprocessador, distinguem os sons com frequências similares ao quebrar de um vidro (chocalhar de chaves, brinquedos, ladrar de cães, etc.). Alguns sensores distinguem o som pela análise de pontos de frequència específicos do som de um vidro a partir, outros pelo processamento de frequências no som que ocorre imediatamente depois do impacto no vidro, mas antes dele partir. No sentido de se optimizar a instalação, os detectores de choque são usualmente instalados juntamente com sensores de vibração.

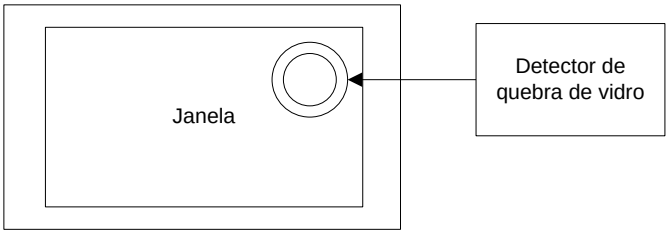

Figura 2- 1 Exemplo de um detector de quebra de vidro.

2.2.4 Detectores de movimento

Os detectores de movimento constituem outro nível de protecção. Assim que detectam movimentos numa determinada área protegida, fazem tocar o alarme. Os PIR (*passive infrared detector*) são os mais comuns em instalações do tipo residencial. Os avanços tecnológicos e a redução dos custos do fabrico dos PIR fez com que os antigos sensores ultra-sónicos fossem praticamente postos de lado em caso de instalações de segurança. OS PIR detectam a energia infravermelha radiada pelo corpo do intruso ao entrar numa zona protegida. Dado que podem reagir a outras fontes de calor como, por exemplo, animais, luz do sol e até coorentes de ar, este tipo de sensores não é totalmente fiável fora de uma habitação (neste caso, os ideais são os que utilizam a detecção por microondas). Mesmo dentro de uma habitação, a intalação dos PIR requer alguns cuidados:

- Não devem ser virados para fontes de calor, ventoinhas, aparelhos de ar condicionado, janelas expostas ao sol ou mesmo locais com variações bruscas de temparatura.
- Devem estar direccionados para locais de passagem que o intruso possa eventualmente utilizar;

- Devem estar direccionados a uma altura superior a um animal doméstico;
- Deve-se ter em conta o raio de cobertura do sensor, de forma a garantir a protecção de toda a zona.

Em áreas como cozinhas e garagens, onde as temperaturas mudam bruscamente, devem-se escolher detectores que combinem a tecnologia PIR e as microondas. Enquanto que os detectores PIR detectam o calor infravermelho, os detectores por microondas detectam cortes no freixe de emissões de ondas de altas frequências, provocadas pela intromissão de uma pessoa no seu campo de acção. Aos contrário dos PIR, os detectores por microondas não são sensíveis a correntes de ar. No entanto, podem criar falsos alarmes por reflectirem superfíceis metálicas, ou ainda reagir a movimentos fora da área a proteger, uma vez que as ondas de radiofrequência atravessam paredes.
A escolha de sensores certos para os diferentes locais é absolutamente essencial para garantir a eficácia do sistema de intrusão. Os sensores são peças fundamentais na constante vigilância do estado da casa. É graças a este tipo de sensores especiais que é possível detectar e eventualmente resolver automaticamente tentativas de assalto ou de emergência. Um exemplo de um detector PIR é mostrado na figura 2-2.

Figura 2- 2 Exemplo de um detector PIR de cortina [3].

2.2.5 Detector termovelocimétrico

Um detector térmico monitoriza metade da área abrangida por um detector óptico, que deve ser colocado à mesma altura. A sua principal função é responder como alarme a um aumento rápido da temperatura, ou quando é atingida uma temperatura predefinida. Não é afectado pelo pó, e apropriado para fogos com combustão total (fogos com base em álcool em que não há fumo).

2.2.6 Sistema automático de extinção por gases – SAEG

A central de detecção de gás, deverá ter a capacidade para organizar e classificar as informações transmitidas pelos sensores em três níveis diferentes de perigos, os quais são classificados em PPM (partículas por milhão), no caso dos gases tóxicos, e em LEL (limite inferior de explosão), no que diz respeito aos gases explosivos. Qualquer sistema dedicado à detecção de gases deverá estar sempre interligado ao sistema de detecção de incèndios oara que haja interacção e operação conciliada, podendo através deste ser também realizada a respectiva integração no sistema de gestão e supervisão, caso exista [4].

2.2.7 Sistema automático de evacuação de emergência

Um sistema de evacuação de emergência visa não só preservar os ocupantes através de evacuação orientada e segura, mas também reunir as condições para que as equipas de intervenção possam combater e conter o foco do incêndio de forma mais eficaz. O sistema sonoro a implementar para este fim inclui os seguintes dispositivos:

- Altifalantes;
- Dispositivos ópticos de sinalização (*strobes*);
- Dispositivos sonoros de sinalização (sirenes);
- Dispostivos óptico-sonoros de sinalização (sirenes com *strobe*);
- Telefones de bombeiros;
- Unidades de controlo por satélites;
- Painéis repetidores de alarme;
- Redes eléctricas associadas.

Todos os dispositivos de sinalização, altifalantes, sirenes e *strobes*, devem ser organizados em circuitos de acordo com a configuração de cada edifício. Deve corresponder a cada um, uma zona de evacuação.

3 O MODELO DO SISTEMA

3.1 Os dispositivos e as suas funções

Dado que o principal objectivo deste trabalho consiste na projecção de um modelo de sistema para a gestão de um sistema de intrusão em espaços residenciais, que possibilite o aumento de segurança, que através da sua programação, permite criar uma série de funções especificamente pensadas para reagir a informações transmitidas pelos seus periféricos, sejam eles contactos magnéticos de porta, detectores de movimento, detectores de quebra de vidro ou outros, como já foi explicado no capítulo anterior . Neste capítulo irão ser dados exemplos de características e funcionalidades envolventes de um alarme de intrusão, com funções bem definidas (modos de funcionamento, zonas de funcionamento do alarme, funcionamento de sensores e teclado) que comunicam entre si através de um barramento de comunicações, utilizando a norma CANopen.

Esta reacção prevê várias formas de assinalar a ocorrência, desde o disparo automático de uma sirene exterior, ao envio de uma mensagem de voz pré-programada para um ou mais números de telefone e / ou uma mensagem de dados para uma central receptora de alarme.

3.1.1 Painel de Alarme

O cérebro de um sistema de alarme será, obviamente o painel. Controla, envia e recebe sinais dos sensores espalhados no imóvel. O painel de alarme será dividido por “Zonas” que identificam com precisão o ambiente violado. E por ser micro-processado é totalmente programável, como opção por zonas 24 horas e modificação da temporização de entrada e saída. Controlado por um teclado, permite inclusão de várias opções descritas ainda neste capítulo.

3.1.2 Sensores

São os dispositivos que identificam alguma movimentação. Existem diversos tipos, dependendo da necessidade de segurança. Devem ser supervisionados, avisando a central caso alguma manutenção seja necessária, como queda de equipamento ou alteração do ambiente em que se encontra. Sem o perfeito funcionamento dos mesmos, não é possível a

identificação de movimentações, podendo até ser um transtorno para o sistema com disparos em falso.

3.1.3 Os Modelos do Dispositivo

De seguida, serão apresentados os modelos do sistema de intrusão incluindo os seus parâmetros de configuração, assim como um modelo base de funcionamento do sistema. A cada modo de funcionamento, estarão associados os respectivos grupos com as zonas de funcionamento. Estão assim apresentados na tabela 3.1 as respectivas funcionalidades.

Tabela 3. 1 Modos e zonas de funcionamento do Alarme.

MODO FUNCIONAMENTO ALARME	
Modo 1	Permite saber qual o Modo de Funcionamento que está activo.
Modo 2	
Modo 3	
Modo n	
ZONAS	
FUNÇÃO	Permite escolher a função da zona seleccionada.
DESCRIPTOR	Seleccionar descriptor para uma determinada zona.
PERMISSÕES	Define zonas permissíveis através do utilizador.
GRUPOS	Selecciona o grupo a que cada zona pertence.
TECLADO	
	Permite mudar definições predefinidas pelo utilizador manualmente.
SENSORES	
S1,…, Sn	Sensores conjugados com o alarme.

3.1.4 Os modos de funcionamento

Iniciando a exposição, os cenário de cada modo de funcionamento são hipoteticamente descarregados para todos os dispositivos sempre que necessário, definido previamente pelos utilizadores.
Através do modo de funcionamento, se os grupos estiverem carregados e o utilizador tenha feito a escolha do grupo, então o estado do grupo é demonstrado previamente no display.

3.2 Zonas de funcionamento

3.2.1 Omitir Zonas

Esta opção permite que as zonas sejam temporariamente removidas (omitidas) do sistema. Uma vez que a zona foi omitida, não será gerado um alarme. As zonas omitidas são restabelecidas quando o sistema for desactivado, ou manualmente quando a opção de omissão de zona for desactiva.

3.2.2 Selecção de Zonas

Esta opção permite que o usuário seleccione qualquer zona, independentemente do tipo de função. Através deste teste, todas os detalhes de cada zona serão mostrados no display do teclado.

3.2.3 Função

Esta opção permite seleccionar a zona pretendida. A cada zona pode ser atribuída uma função diferente para cada modo de funcionamento para permitir um máximo de flexibilidade.

3.3 Funcionamento Interno do Sistema de Intrusão e Respectiva Comunicação

Através da Figura 3-1, podemos observar no diagrama de blocos, como será efectuada a comunicação do microcontrolador com a rede CAN. Esta arquitectura mostra-se algo simples.

O micontroladador é responsável não só pela leitura de dados, mas sim pelo empacotamento desses mesmos dados no formato determinado pela rede CAN, fazendo a transmissão dos dados pela rede (Transceiver CAN) para outro dispositivo. A comunicação entre o MSP430 e o controlador CAN, é feito através de um barramento SPI. Os mecanismos Rx e Tx, mais não são que os respectivos meios de recepção e transmissão COB (*Communication Object*), ou seja, é através do COB que é possível ser efectuada uma comunicação Cliente/Servidor com outro dispositivo. Um tipo de COB que segue este tipo de relação, tendo um acesso directo aos *Application Objects*, são os PDO (Process Data Object), que são utilizados para transmissão de dados em tempo real. O conteúdo de cada PDO é definido previamente no dicionário de objectos (OD), que mais não é que a *interface* entre a informação contida nos dispositivos e a rede atribuindo um endereço lógico a cada endereço físico da memória interna do dispositivo, e pode ser configurado utilizando mensagens do tipo SDO (*Service Data Objects*).

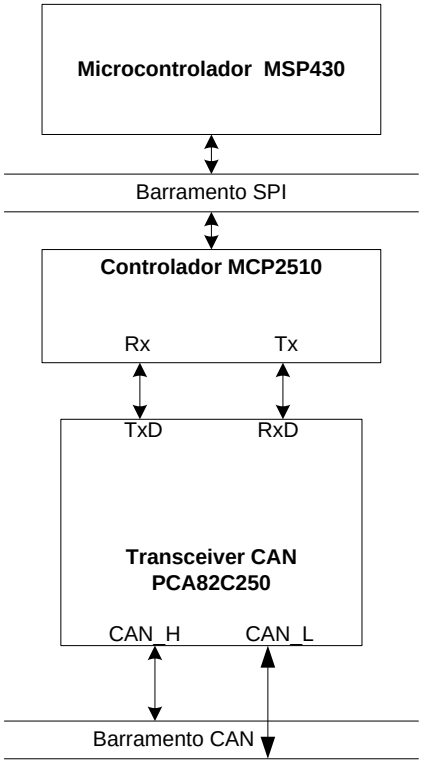

Figura 3-1 Comunicação das placas de ensaio.

Na figura 3.2 está representado um exemplo do funcionamento do sistema de intrusão presente neste trabalho. O sistema é composto pela central de alarme, que tem como funcões internas os respectivos modos e zonas de funcionamento, teclado e respectivos sensores. Os modos de funcionamento mais não são que as funções que o alarme pode tomar escolhidas previamente pelo utilizador. A cada modo de funcionamento irá corresponder a respectiva zona, zona essa que pode ser também seleccionada arbitráriamente pelo utilizador consoante as suas necessidades. Todo o funcionamento exterior, ou seja, a comunicação da central para os sensores e respectivo teclado, é feita por uma barramento CAN. Sempre que ocorra alguma anomalia ou caso seja necessário haver troca de informação, essa comunicação será feita recorrendo aos COB (*Communication Objects*), mais concretamente utilizadando PDOs. O conteúdo de cada PDO é definido previamente no dicionário de objectos e pode ser configurado recorrendo a mensagens do tipo SDO. Todos os objectos do dispositivo serão acedidos recorrendo ao já falado dicionário de objectos, pois é a partir dele que se faz o mapeamento da estrutura interna do nosso modelo. Cada objecto dentro do OD será endereçado com um índice de 16 bits, e um sub-índice de 8 bits. Através do dicionário de objectos, é possível ver todos os parâmetros que descrevem o dispositivo.

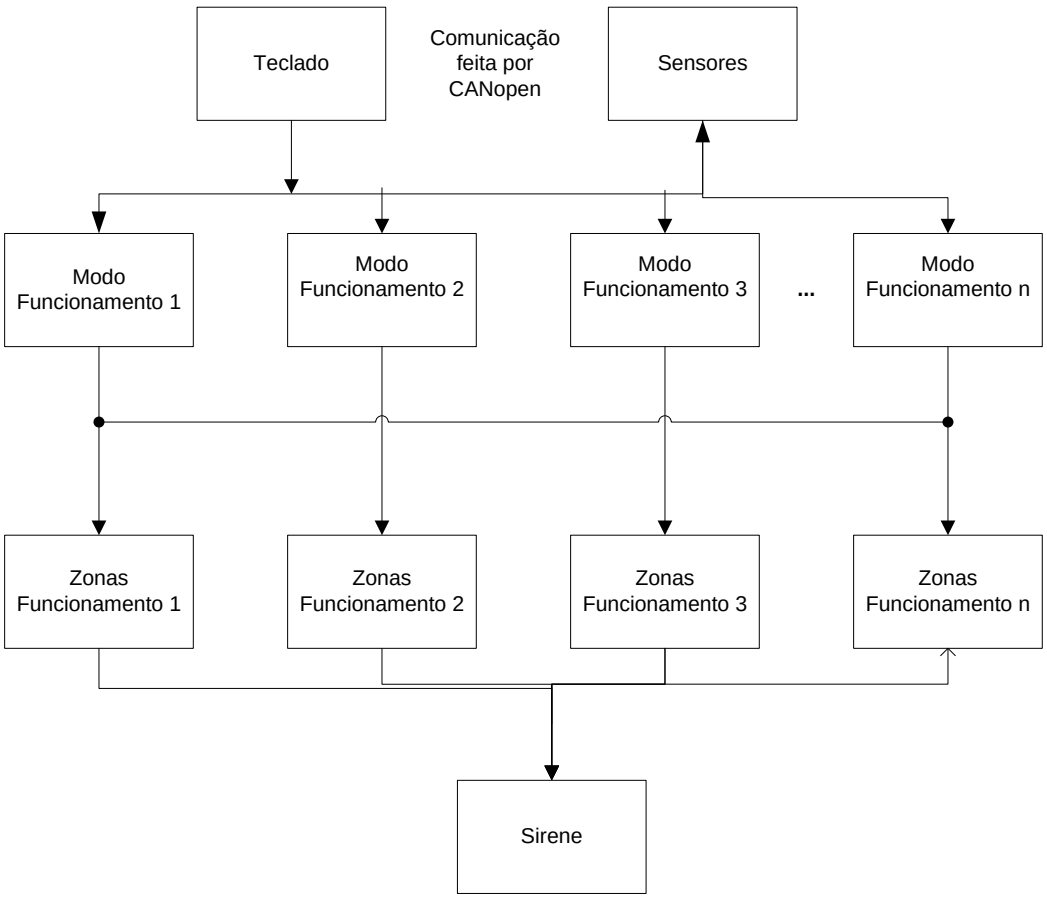

Figura 3-2 Exemplo prático do funcionamento do sistema de intrusão.

A figura 3.3 representa o modo como a comunicação do modelo de dispositivo é feita. Como já foi explicado anteriormente, os principais componentes para esta comunicação são o

dicionário de objectos, PDOs, que é previamente definido no OD, e os *Service Data Objects* que permitem as ligações *MASTER/SLAVE* presentes na figura. Os SDOs e respectivos PDOs terão aqui duas funções distintas:

Consoante a recepção ou envio de informação na rede, os SDOs serão do tipo rx (*receive*) ou tx (*transmit*), assim como os PDOs serão configurados como TPDOs, em que o 'T' significa transmissão, ou RPDOs como modo de recepção. O Controlo de Erros serve para que sempre quando ocorra um erro de transmissão, este irá ter a capacidade de cancelar a transmissão enviando um aviso de erro (*Error Flag*) para os outros nós do barramento [8].

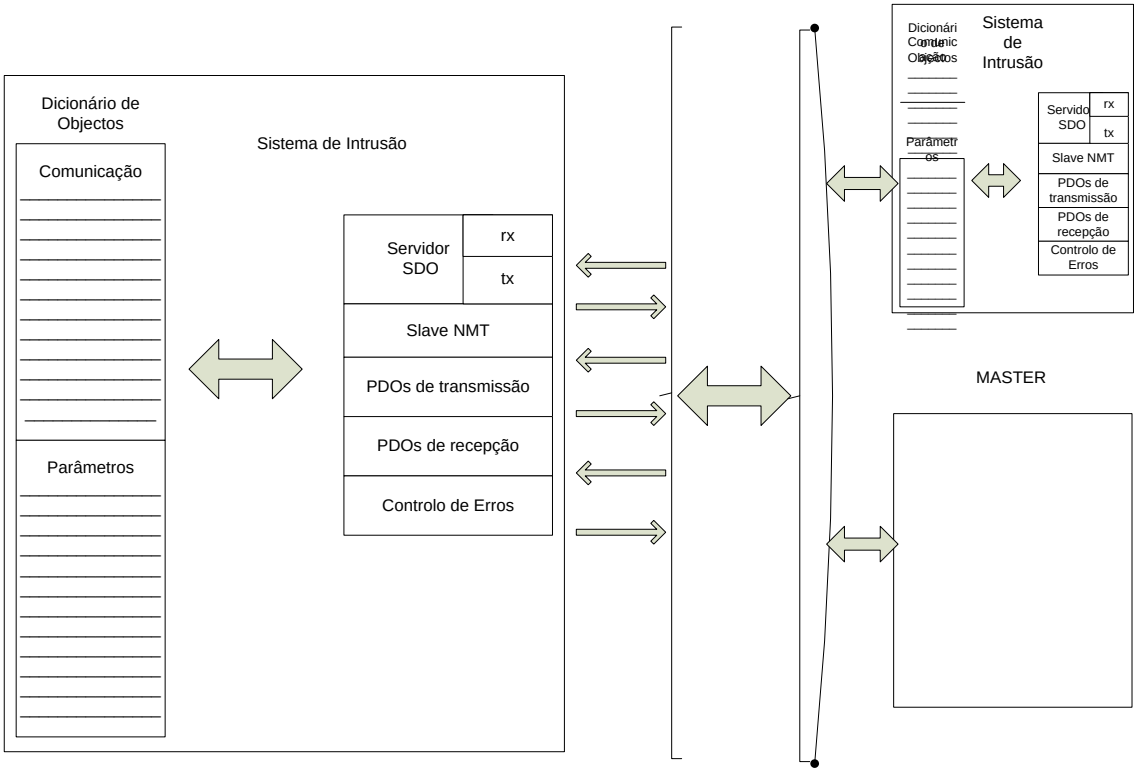

Figura 3-3 Comunicação CANopen entre o modelo de dispositivo.

O conceito definido na figura 3.4 como opção superior para a configuração de funcionamento de um alarme de intrusão consiste, certamente, numa forma interessante de se encarar a domótica. Uma prévia definição de modos e zonas de funcionamento (cenários), reflectindo as vontades do utilizador, com a consequente activação por eventos temporizados ou externos, conduz a uma simplificação de manuseamento do sistema domótico, escondendo a complexidade no seu interior. O sistema presente neste trabalho apresenta uma vocação intrinseca para este modo de operação, pois baseia-se num mecanismo de partilha de dados

entre os elementos da rede, os quais poderão ser acedidos de uma forma expedita por meio de PDOs ou SDO, como já foi explicado nas figuras antecedentes.

Todas as unidades Canopen funcionarão como servidores para todos os objectos mapeados pelo dicionário de objectos, cabendo o papel de cliente à unidade integradora e supervisora (UIS). Os cenários são programados e mantidos na unidade UIS, sendo esta unidade um repositório de cenários que, mediante um evento específico, fará o *download* de novo cenário para os objectos residentes nas unidades CANopen dispersas, para um novo comportamento destas unidades, do qual resultará num novo estado de funcionamento da habitação [5].

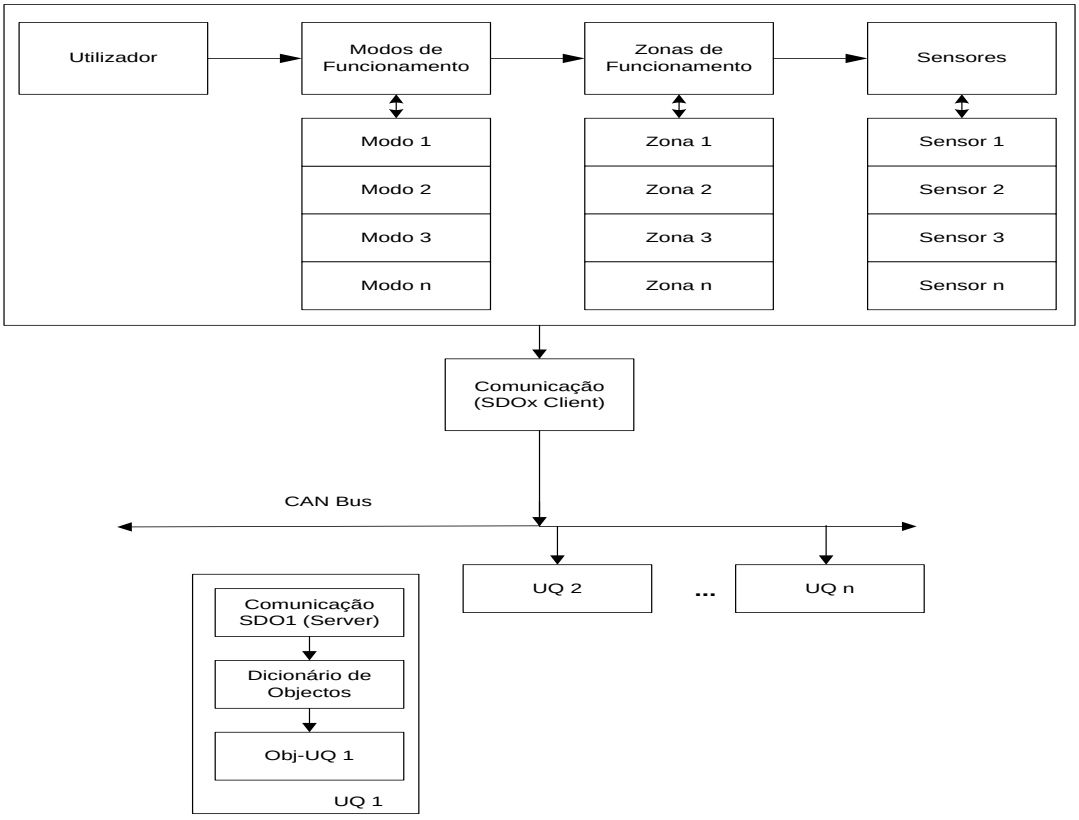

Figura 3-4 Ilustração de cenários de funcionamento. [5]

As figuras 3-5 e 3-6 mais não são que uma representação esquemática da arquitectura interna do modelo de intrusão presente neste trabalho. É constuído então pelo bloco de utilizadores, que têm o papel fundamental de mudar as configurações dos modos e zonas de funcionamento consoante o desejado. Volta-se a referir através do esquema da figura 3-6, que toda a transmissão de dados entre a central de alarme e os respectivos sensores e teclado, será feita através das normas de comunicação CANopen.

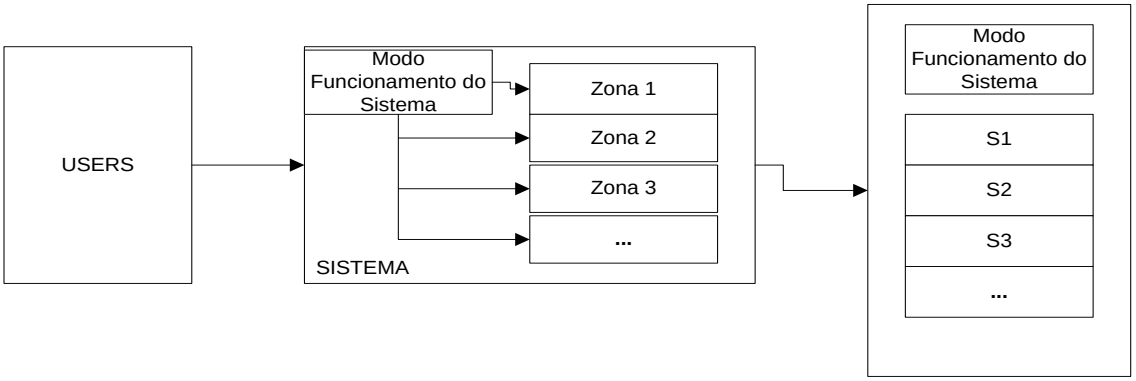

Figura 3-5 Representação esquemática da arquitectura do modelo de intrusão.

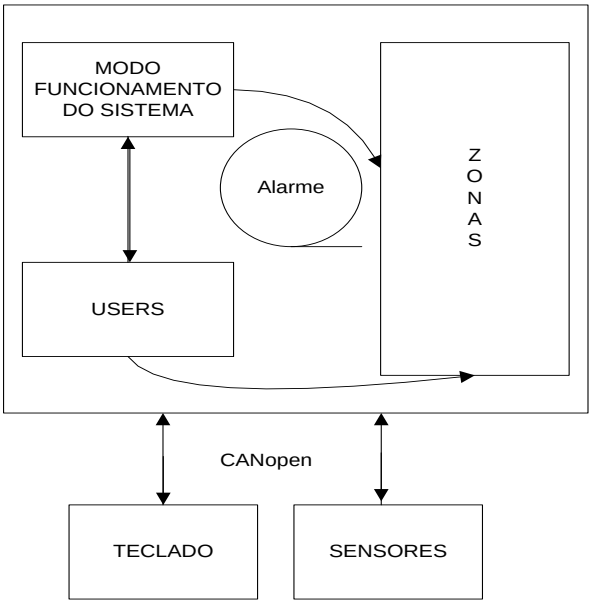

Figura 3-6 Representação esquemática da arquitectura interna do modelo de intrusão.

A figura 3.7, mostra o funcionamento interno da central de alarme. A cada modo de funcionamento irão corresponder as respectivas zonas. Como os sensores devem informar sempre que haja alguma alteração ou anomalia do sistema, foi optado por se usarem portas AND, pois basta um sensor ficar em circuito aberto, para se conseguir enviar imediatamente a respectiva informação (basta que uma entrada de uma porta AND seja '0', para a sua saída ser '0' também). Em relação à ligação entre os modos de funcionamento e as zonas, resolveu-se utilizar um circuito aberto, pois será apenas da responsabilidade do utilizador optar pelas

zonas que quer associadas ao respectivo modo (ex: é possível termos associado um modo de funcionamento a diferentes zonas de funcionamento), não limitando assim as respectivas funções, como pôde ser visto no capítulo 3.2.3.

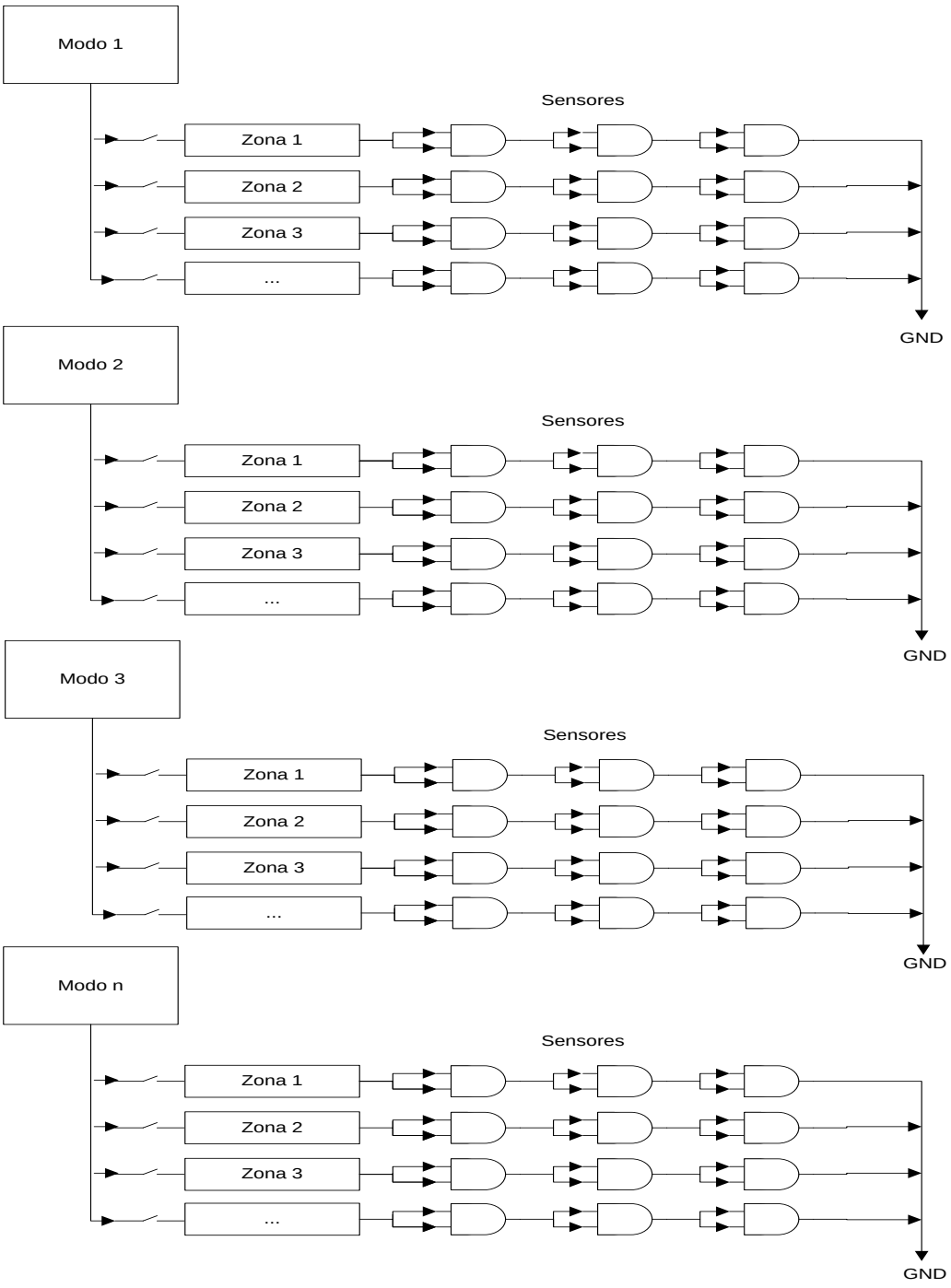

Figura 3-7 Funcionamento interno da central de alarme.

4 DICIONÁRIO DE OBJECTOS E OS EDS

4.1 Os Dicionários de Objectos

O bloco mais importante de um dispositivo CANOpen é o dicionário de objectos, composto por um grupo ordenado de definições de objectos acessíveis pela rede. Cada nó da rede tem o seu dicionário que utiliza para interagir com os outros nós, através de objectos de comunicação já descritos. Cada dicionário tem grupos de objectos *standard*, grupos de objectos definidos para o ambiente de utilização pretendido (geralmente definidos pelos fabricantes em *standards* públicos), e ainda grupos de objectos a definir pelo utilizador.

No dicionário, estão os objectos de comunicação, os tipos de dados que os nós podem trocar entre si, os objectos de erro, etc. Sempre que um nó pretende executar uma tarefa, qualquer que ela seja, tem de retirar a informação a tratar do dicionário, chamar o objecto de comunicação adequado do dicionário e, opcionalmente, verificar o sucesso da sua tarefa através de um outro acesso ao seu dicionário ou a um dicionário remoto.
O modelo de um dispositivo CAnopen contém dois tipos básicos de objectos:

- *Communication Objects* - cada um representa uma funcionalidade de comunicação específica do dispositivo. Os *Communications Objects* são específicos no perfil de comunicações CANopen.
- *Application Objects* - representam funcionalidades específicas do dispositivo tais como o estado de um sinal de input digital. Os *Application Objects* são defindos no perfil do dispositivo.

Os *Communications Objects for Network Management* - permite estabelecer comunicações *Master/Slave* e pode ser usado para controlo global de dispositivos e verificação do estado.

- *Communication Objects for Network Management* – permite estabelecer comunicações *Master/Slave* e pode ser usado para controlo global de dispositivos e verificaçãp do estado.
- *Communication Objects for Application Data Transfer* - estes podem ainda ser divididos em duas categorias:

    - Alguns objectos fornecem acesso indexado a todos os objectos do dispositivo através do OD, o que permite uma comunicação *Client/Server* com outro dispositivo. Usando estes *Communication Objects* é possível aceder a todas as características do dispositivo,

normalmente para propósitos de configuração. Contudo, uma vez que este tipo de operação requer trocas de informação com o OD, o seu funcionamento não é adequado a um acesso em tempo real, pois sobrecarregaria o protocolo. Um exemplo deste tipo de *Communication Object* é o *Service Data Object* (SDO).

- Outros objectos fornecem acesso directo aos *Application Objects* tornando possível implementar mecanismos de troca de informação síncrona ou assíncrona em tempo real. Este tipo de COBs segue a relação de comunicação *Producer/Consumer* envolvendo um congestionamento mínimo do protocolo. Os *Process Data Objects* (PDos) são um exemplo deste tipo de mecanismo.

4.1.1 Tipos de Dados e regras de Codificação da rede CANopen

Dependendo da propriedade que representa um determinado objecto, este objecto pode ser constituído apenas por um atributo simples, ou pode ser mais complexo, com vários atributos. Os objectos com apenas um atributo são tratados na rede CANopen como variáveis simples, e são simples do tipo Float, Boolean, String ou Integer. Objectos mais complexos, com mais de um atributo, são tratados como Arrays ou Record. Um array é usado para representar um conjunto de atributos do mesmo tipo de dados e um Record, por outro lado, representa um conjunto de atributos de diferentes tipos de dados.

Um exemplo de um objecto simples que pode ser representado por uma variável simples, é o *Device Type Object*, que é armazenado como uma palavra de 32 bits Unsigned. Por outro lado, um Array pode ser usado para representar as sáidas de um módulo digital I/O. Finalmente, os Record podem ser usados, por exemplo, para representar diferentes parâmetros associados a uma mensagem CAN que transmite um dispositivo: o identificador (palavra 16 bits do tipo unsigned), o tipo de troca de dados que é usado (palavra 8 bits do tipo unsigned), o mínimo tempo entre as transmissões (palavra 16 bits do tipo unsigned), etc. Os tipos de dados pré-definidos estáticos e complexos são mostrados na tabela 4.1.

Tabela 4. 1 DSP301 Data Types. [8]

Static			COMPLEX
BOOLEAN	UNSIGNED8	REAL64	PDO COMMUNICATION PARAMETER
INTEGER8	UNSIGNED16	VISIBLESTRING	
INTEGER16	UNSIGNED32	OCTET STRING	PDO MAPPING PARAMETER
INTEGER24	UNSIGNED24	DATE	
INTEGER32	UNSIGNED40	TIME OF DAY	SDO COMMUNICATION PARAMETER
INTEGER40	UNSIGNED48	TIME DIFFERENCE	
INTEGER48	UNSIGNED56	BIT STRING	INDENTIFY
INTEGER56	UNSIGNED64	DOMAIN	
INTEGER64	FLOAT		

Todos os objectos de um dispositivo podem ser cedidos através do dicionário de objectos. O OD é a parte fundamental do modelo do dispositivo já que mapeia a estrutura interna do dispositivo: se a estrutura do OD de um determinado dispositivo é conhecida, então tudo o que é necessário para construir uma aplicação distribuída usando esse dispositivo é conhecido [8]. Cada objecto dentro do OD é endereçado com um índice de 16 bits e um sub-índice de 8 bits. Cada dispositivo na rede possui o seu dicionário de objectos que contem todos os parâmetros que descrevem o dispositivo. A estrutura geral de um dicionário de objectos é apresentada na tabela 4.2.

Tabela 4. 2 Organização do dicionário de objectos [8].

INDEX	OBJECTS
0000	Não usado
0001-001F	Static Data Types
0020-003F	Complex Data Types
0040-005F	Manufactured Specific Data Types
0060-007F	Device Profile Specific Static Data Types
0080-009F	Device Profile Specific Complex Data Types
00A0-0FFF	Reservado
1000-1FFF	Communication Profile Area
2000-5FFF	Manufactured Specific Profile Area
6000-9FFF	Standardised Device Profile Area
A000-FFFF	Reservado

De modo a não tornar este capítulo excessivamente longo, todas as tabelas dos ODs encontram-se no anexo B.

4.1.2 O Dicionário de Objectos de Funcionamento do Alarme

Na figura 4.1 está representado o mecanismo de transmissão entre os sensores e o respectivo teclado com a central de alarme. A comunicação como já foi dita anteriormente é feita através do uso de PDOs. O conteúdo de cada PDO é definido previamente através do acesso ao dicionário de objectos e pode ser configurado recorrendo a mensagens do tipo SDO.

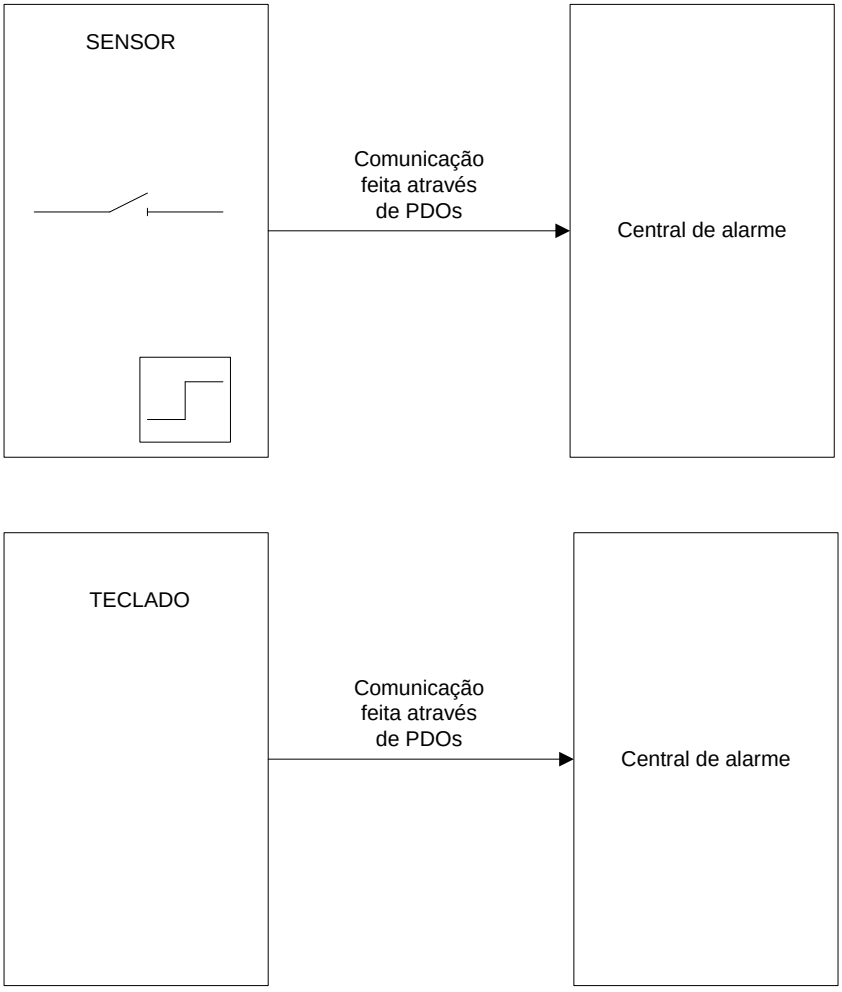

Figura 4-1 Mecanismo de transmissão de um sensor e teclado para com a central de alarme.

4.2 Os Electronic Data Sheets (EDS) e os Dispositivos de Arquivo de configuração (DCF)

OS EDS e DCF são maneiras padrão de transmitir informações sobre o dispositivo para o usuário do dispositvo, especificados pela norma CANopen. A existência destes meios estandardizados devem permitir a implementação de ferramentas que pode executar automaticamente um conjunto de tarefas como a configuração de dispositivos CANopen.

O formato de documentos dos respectivos EDS e DCF está definido para o armazenamento electrónico. Os arquivos resultantes serão codificados no código ASCII, e é recomendável utilizar o conjunto de caracteres ANSI. Um EDS descreve as características a seguir num dispositivo:

- A funcionalidade de comunicação e objectos sao imlementados de acordo com DS301.
- A funcionalidade de dispositivos específicos e objectos são implementados de acordo com os perfis de dispositivos CANopen.

A EDS descreve características que não estão conforme a configuração, é, portanto, um modelo que é sempre uma descrição válida do dispositivo, independentemente da sua configuração. O DCF, por outro lado, descreve não só os objectos que um dispositivo implementa, mas também os valores para uma determinada configuração. O *baudrate* e respectivo nó ID do dispositivo também são incluídos num DCF. Assim, vários DCFs podem ser baseados numa EDS, o que reflecte várias configurações de dispositivos possíveis. Uma EDS deve ser entregue pelo fornecedor de um determinado dispositivo, como parte do seu documento. Em nenhum EDS é fornecido, o padrão EDS para que determinado tipo de dispositivo (com base no perfil de dispositivo aplicável) pode ser utilizado e deve ser válido.

O DCF pode ser muito útil para armazenar uma configuração específica de um dispositivo que será nessário para restaurar mais tarde. É então possível para uma ferramenta de configuração automática para pegar neste arquivo e executar a configuração automaticamente[8].

4.2.1 Estrutura de uma *Electronic Data Sheet* (EDS)

Uma EDS é funcionalmente dividida em três grandes blocos, cada bloco com um ou mais pontos:

- Informações sobre o arquivo – há apenas uma secção nesta parte da EDS e que contém uma descrição do ficheiro, a data e hora da sua criação e outras informações básicas.

- Secção de informações gerais do dispositivo – há também apenas uma secção nesta parte da EDS e que contém informação sobre o produto e informação do dispositivo fabricado.
- Informações gerais do dispositivo – há apenas uma secção nesta parte da EDS e que contém informação do tipo de produto e fabricaçãodo dispositivo. Também é descrito quais os recursos que são implementados no dispositivo e se ele suporta o mapeamento dinâmico de PDOs.
- Secção do Dicionário de Objectos – o número da secção nesta parte da EDS vai depender da informação do dispositivo no dicionário de objectos [8].

5 ENSAIOS

5.1 As Unidades de Ensaio e Respectiva Configuração

Foram utilizadas neste trabalho duas unidades de ensaio [1] para que se avaliassem os princípios base do modelo do sistema. Para isso, teve-se por base a norma CANopen segundo a qual foram específicados um conjunto de dispostivos básicos, definindo as suas funções. O hardware de uma das placas de ensaio, devido a utilizar uma tensão de 3.3V, levou à utilização do microcontrolador MSP430 da *Texas Instruments* [6], também este com um consumo de 3.3V, com uma frequência de funcionamento de 16MHz e disponibilizando até 8Kb de memória RAM. A *interface* do microcontrolador com o controlador CAN MCP2510 [2], foi efectuada a partir de um barramento SPI (*Serial Peripheral Interface*) e também através de um conjunto de linhas de *input* e *output* bastante condicionados. Dos 16 pinos disponíveis, 8 são ocupados pelo driver ULN2803, que é integrado por 8 transístores [10]. Continuando, a ligação do controlador CAN MCP2510 ao barramento deverá ser realizada mediante um *transceiver* CAN, em que o objectivo é o de garantir os níveis lógicos no barramento, recorrendo neste caso ao circuito da *Philips* com a referência PCA82C250 [11].

5.1.1 O Controlador CAN MCP2510

O MCP2510 é um controlador CAN desenvolvido para aplicações simples que necessitem de um interface de comunicação com um barramento CAN. O controlador é constítuido por três bloos principais, o mecanismo de protocolo CAN, os registos *Control Logic e* SRAM que são utilizados para configurar o dispositivo e o seu funcionamento e um bloco do protocolo SPI (*Serial Peripheral Interface*). Na figura 5.1, é apresentado o diagrama de blocos do respectivo controlador. [2]

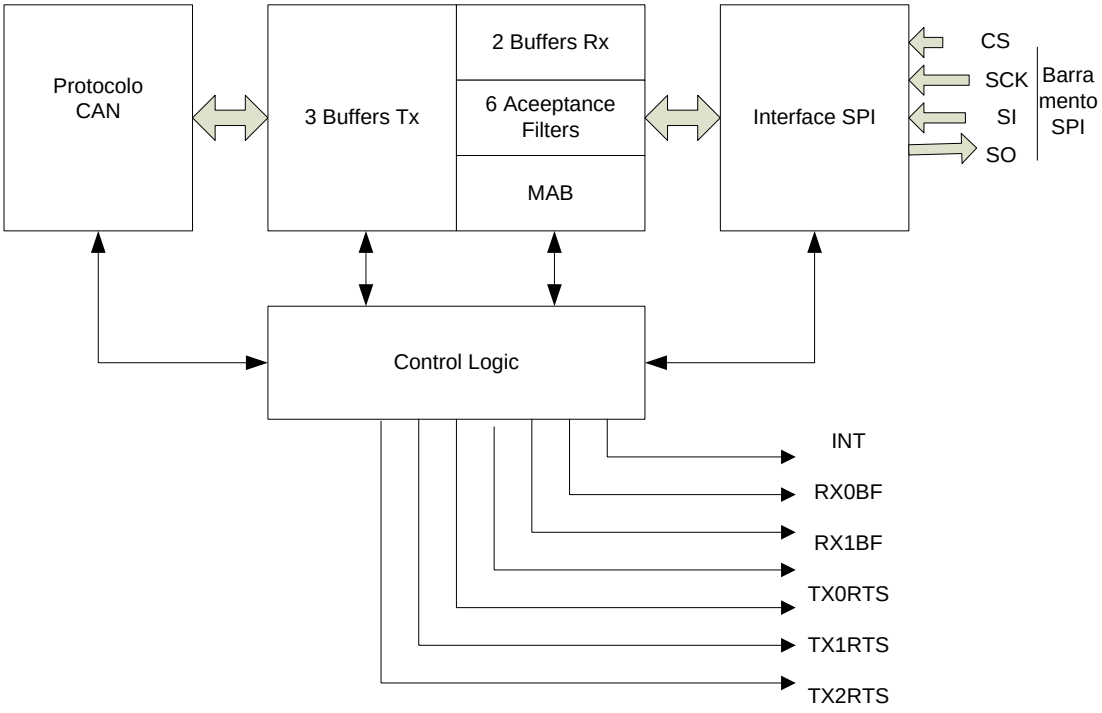

Figura 5- 10 Diagrama de blocos do controlador MCP2510 [2].

O protocolo CAN manipula todas as funções para a recepção e transmissão de mensagens sobre o barramento. As transmissão de mensagens são iniciadas usando bits *control register*, através do interface SPI, ou usando os pins de transmissão permitidos. Qualquer mensagem detectada no barramento é verificada a nível de erros, e é então comparada com o usuário de *Acceptance Filters* que verifica se é necessário mover um dos dois buffers de recepção, rx.

5.1.1.1 *Buffers de Transmissão/Recepção*

O controlador MCP2510 é constituido por 3 buffers de transmissão (tx) e dois de recepção (rx), duas *Acceptance masks*, uma para cada buffer de transmissão, e seis *acceptance filters*.

5.1.1.2 Buffers de Transmissão

Cada um desses buffers ocupa 14 bytes da SRAM, e são mapenados na memória do dispositivo. O primeiro byte, TXBNCTRL, é um registo de controlo associado ao buffer de mensagens. As informações contidas neste registo determinam as condições sob as quais a mensagem será transmitida, indicando o *status* da transmissão, através do registo TXRTSCTRL. Os 5 bytes seguintes são usados para armazenar os indentificadores *standard e* extendidos e também para outras mensagens arbitrárias (através do registo TXBNSIDH usando o registo TXBNDM).

Para o microcontrolador ter acesso de escrita no buffer de mensagens, o bit (TXREQ) do registo TXBNCTRL tem de estar desabilitado (“0”), indicando que o buffer está desabilitado de qualquer mensagem pendente que possa ser transmitida. No mínimo os registos TXBNSIDH, TXBNSIDL e TXBNDLC deverão estar habilitados. Se os bytes de dados estão presentes na mensagem, o registo TXBNEIDm tem também de ser habilitado e o bit 3 (EXIDE) do registo TXBNSIDL terá que estar habilitado em nível alto (“1”). Antes do envio da mensagem o microcontrolador deve inicializar o bit CANINTE.TXINE para habilitar ou desabilitar a geração de uma interrupção quando a mensagem for enviada. Devem ser também inicializados os bits prioritários TXBNCTRL.TXP.

Para inicializar a transmissão de uma mensagem o bit TXBNCTRL.TXREQ deve estar habilitado para cada buffer que é transmitido. Caso a transmissão seja inicializada através do barramento SPI, o bit TXREQ pode ser habilitado ao mesmo tempo que os bits prioritários TXP. Quando o bit TXBNCTRL.TXREQ estiver habilitado, os bits TXBNCTRL.ABTF, TXBNCTRL.MLOA e TXBNCTRL.TXERR serão desabilitados. Ao habilitar o bit TXBNCTRL.TXREQ não implica que a transmissão da mensagem seja efectuada, pois apenas será sinalizada como pronta para a transmissão. A transmissão apenas começará quando o dispositivo detectar que o barramento está apto para tal. Quando a transmissão for completada, o bit TXBNCTRL.TXREQ será desabilitado, ao contrário do bit CANINTF.TXNIF, e será então gerada uma interrupção caso o bit CANINTE.TXNIE esteja habilitado.

O microcontrolador poderá cancelar a mensagem num buffer específico, desabilitando o bit TXBNCTRL.TREQ. Assim, todas as mensagens pendentes podem ser abortadas habilitando o bit CANCTRL.ABAT. Se o bit CANCTRL.ABAT estiver habilitado para cancelar todas as mensagens pendentes, o utilizador deverá desabilitar este bit (depois de verificar que todos os bits TXREQ estejam desabilitados) para continuar a transmitir a mensagem. Na figura 5.2 é apresentado um fluxograma para exemplificar a transmissão de uma mensagem.[2]

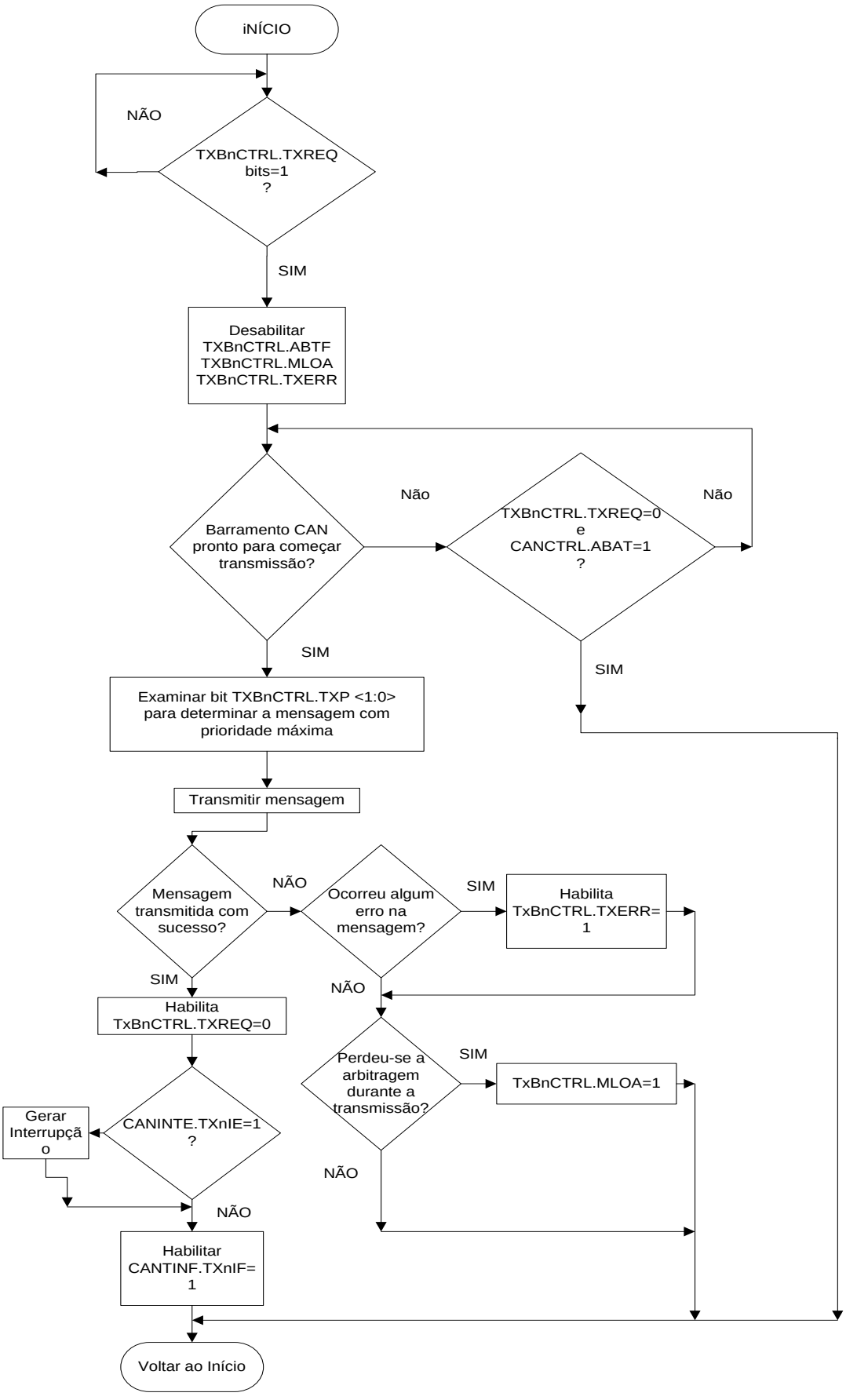

Figura 5- 11 Fluxograma de envio de mensagens [2].

5.1.1.3 Buffers de Recepção

O controlador MCP2510 é constituido por dois buffers de recepção com vários filtros de aceitação para cada um deles, e ainda um *Message Assembly Buffer* (MAB) que actua em separado como um terceiro buffer. Nos 3 buffers de recepção, o MAB está encarregado da recepção da próxima mensagem do barramento. Os restantes buffers são então os registos RXB0 e RXB1 que podem receber uma mensagem completa do protocolo. O microcontrolador pode aceder a um buffer, enquanto que o outro buffer está disponivel para a recepção de mensagens ou estando ainda preso por uma mensagem recebida anteriormente. O MAB é responsável por reunir todas as mensagens recebidas, que serão transmitidas para os buffers RXBN.

Quando a mensagem é movida para receber o buffer, o bit CANINTF.RXNIF é habilitado. Este bit deve ser dasabilitado pelo microcontrolador, quando o processo de recepção da mensagemno buffer for completado. Este bit leva a um bloqueio positivo para garantir que o micriocontrolador finalizou a mensagem antes do MCP2510 tentar carregar uma nova mensagempara o buffer de recepção. Se o bit CANINTE.RXNIE estiver habilitado uma interrupção irá ser gerada no pino INT para indicar que uma mensagem válida foi recebida. O registo RXB0 funciona como o buffer mais prioritário e é constituido por dois filtros de aceptação. O registo RXB1 é constituido por quatro filtros de aceptação o que implica uma menor prioridade emrelação ao registo RXB0.

Além disso, o registo RXB0CTRL pode ser configurado de tal modo que se o RXB0 contém uma mensagem válida, e caso se receba outra mensagem válida,leva a que nao ocorra um *overflow,* levando assim a que a nova mensagem seja movida para o registo RXB1 independentemente dos critérios de aceitação desse mesmo registo. Quando é recebida uma mensagem, os bits FILHIT0, BUKT1, BUKT e RXRTR do registo RXBNCTRL indicarão o número de filtros de aceitação que activaram a recepção, e se a mensagemrecebida é um pedido dce transferência remoto. Os bits RXBNCTRL.RXM habilitam modos especiais de recepção. Normalmente, estes bits são habilitados a “00” para permitir a recepção de todas as mensagens válidas, conforme determinado pelos filtros de aceitação.

Assim, a determinação da acitação do querer ou não receber mensagens *standard* ou extendidas é determinado pelo bit RFXNSIDL.EXIDE no filtro de aceitação. Se os bits RXBNCTRL.RXM estiverem habilitados como “01” ou “10”, o receptor irá aceitar mensagens apenas com identificadores *standard* ou extendidas, respectivamente. Se o filtro tem o bit RFXNSIDL.EXIDE habilitado sem corresponder com o bit RXBNCTRL.RXM, o filtro será inútil. Estes dois modos dos bits RXBNCTRL.RXM podem ser utilizados apenas em sistemas onde é

sabido que somente mensagens *standard* ou extendidas entrarão no barramento. Na figura 5.3 é apresentado um fluxograma para a representação de recepção de mensagens [2].

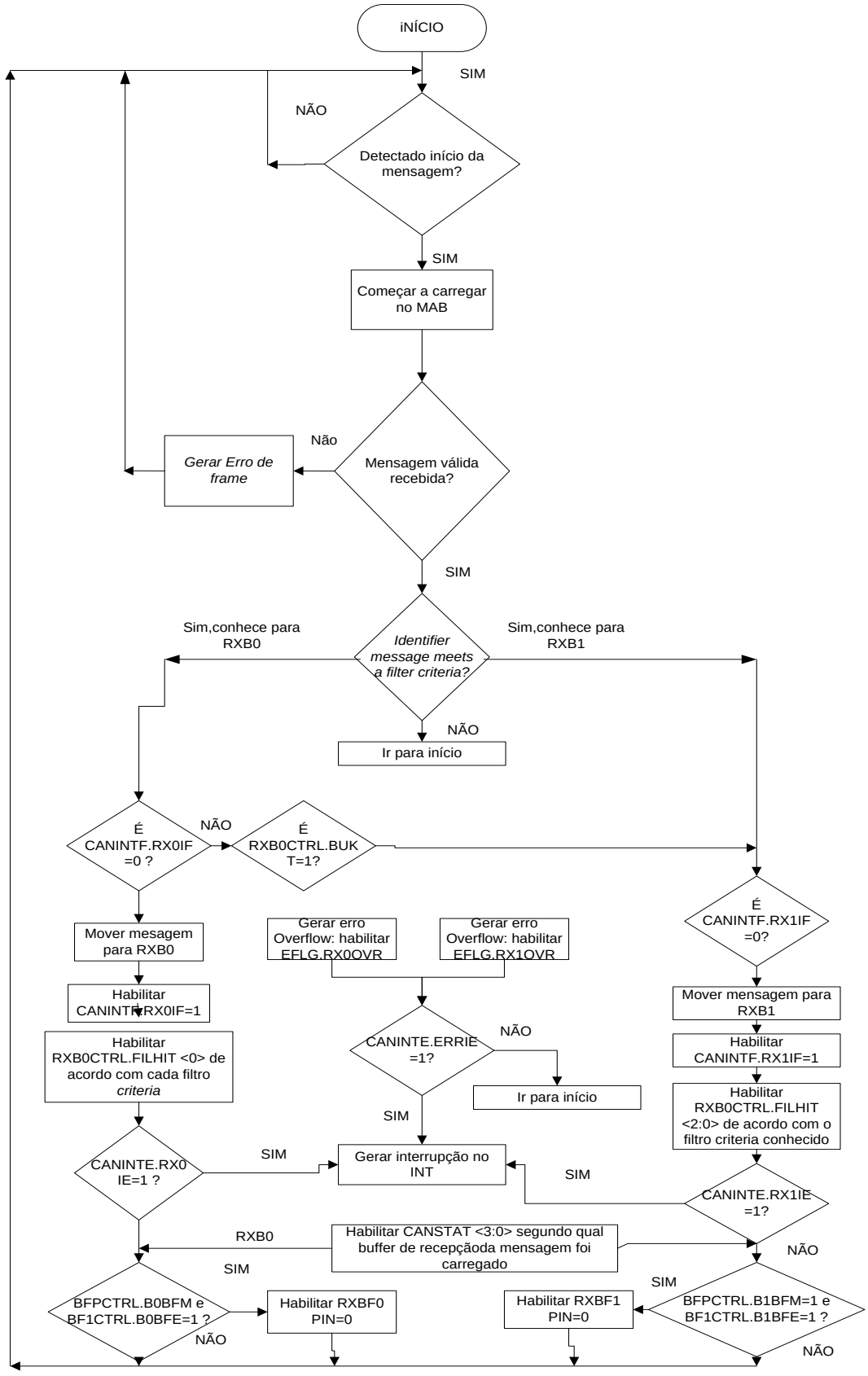

Figura 5- 12 Fluxograma de recepção de mensagens [2].

5.1.1.4 Filtros de aceitação de mensagens e máscaras

As mensagens, filtros e máscaras de aceitação são usadas para determinar se a mensagem no buffer MAB deve ser carregada num qualquer buffer de recepção. Quando uma mensagem válida for recebida no MAB, o *identifier fields* da mensagem é comparado com os valores do filtro. Se existir uma coincidência, a mensagem será carregada para o respectivo buffer de recepção. As máscaras de filtro, que incluem os registos RXFNSIDH até RXMNEID0, são usados para determinar que bits no identificador são examinados com os filtros. Os filtros de aceitação RXF0 e RXF1, e a máscara de flitro estão associados com o registo RXB0, enquanto que os filtros RXF2, RXF3, RXF4, RXF5, e a máscara RXM1 estão associados ao registo RXB1.

5.1.1.5 Bit Timing

É através da configuração do *bit timing*, que se irá definir o *baudrate* com que o sistema irá comunicar. Através disso, todos os nós numa rede CAN devem ter o mesmo *nominal bit rate*, que mais não é que o número de bits transmitidos por segundo assumindo uma transmissão ideal com uma oscilação ideal, numa ausência de ressincronização, sendo definido com um máximo de transmissão de bits equivalente a 1Mb/s. O *nominal bit timing* é assim formado pelos seguintes segmentos:

- Synchronization Segment (Sync_Seg), que é usado para fazer a sincronização dos vários nós, sendo sempre programável com um valor de 1 TQ (*Time Quantum*).
- Propagation Time Segment (Prop_Seg), representa o campo de largura variável utilizado para compensar o atraso de sinais através da rede, sendo programável de 1 a 8 TQ;
- Phase Buffer Segment 1 (Phase_Seg1), que é utilizado para compensar o erro de fase, tendo também uma configuração de TQ programável num intervalo de 1 a 8;
- Phase Buffer Segment 2 (Phase_Seg2), sendo também programado com os mesmo valores de TQ do Phase Buffer Segment 1;

A fórmula para o cálculo do *Nominal Bit Time*, e respectivo *Time Quantum* é representada em seguida:

$$NBT = \frac{1}{NBR} \quad (5.1)$$

$$TQ = 2 \times (BRP + 1) \times T_{OSC} \quad (5.2)$$

$$NBT = (SYNC\_SEG + PROP\_SEG + PHASE\_SEG1 + PHASE\_SEG2) \quad (5.3)$$

5.1.2 Comunicação do controlador MCP2510 com o barramento SPI

O controlador MCP2510 está projectado para fazer interface directamente com o Serial Peripheral Interface (SPI), disponível em muitos microcontroladores e é capaz de suportar os modos “00” e “11”. Os comandos e dados são enviados para o dispositivo através do pino SI. Os dados são conduzidos fora do MCP2510, através da linha SO. O pino $\overline{CS}$ (chip select) tem de estar habilitado obrigatoriamente a nível baixo enquanto a operação é efectuada. A tabela 5.1 exemplifica as instruções com os bytes correspondentes para todas as operações.

Tabela 5- 1 SPI Instruction Set. [2]

Nome da Intrução	Formato da Instrução	Descrição
Reset	1100 0000	Faz um reset aos registos internos, e habilita o modo de configuração
Read	0000 0011	Lê dados do registo começando no endereço seleccionado
Write	0000 0011	Escreve dados para o registo começando no endereço seleccionado
RTS (Pedido de envio)	1000 0nnn	Habilita o bit CTRL.TXREQ para uma ou mais transmissões de buffers 1000 0$n_1 n_2 n_3$ (n_1 para pedido de envio para TXB2, n_2 para TXB1 e n_3 para TXB0)
Read Status	1010 0000	Comando de sondagem que gera os bits para as funções de transmissão/recepção
Bit Modify	0000 0101	Selecção de registos para o Bit modify

5.1.2.1 Instrução de Leitura

A instrução de leitura é iniciada com o pino $\overline{CS}$ em nível baixo, e é então enviada para o MCP2510 seguido pelos endereços de 8 bits (A7 através de A0). Depois das intruções de leitura e endereço enviadas, os dados armazenados no registo no "selected address" serão deslocadas para fora no Pino SO. O ponteiro de endereço interno é automaticamente incrementado para

o próximo endereço depois de cada byte de dados ser deslocado. Assim, é possível ler o próximo registo de endereços continuando a fornecer impulsos de relógio. A operação de leitura é assim encerrada, habilitando o pino chip select ($\overline{CS}$). Toda a operação registada pode ser vista na figura 5-4.

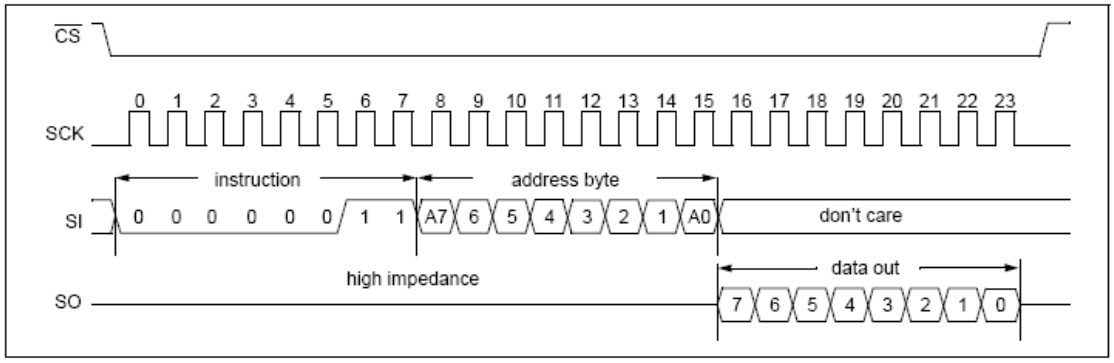

Figura 5- 13 Instrução de Leitura [2].

5.1.2.2 Instrução de Escrita

A instrução de escrita é iniciada com o pino $\overline{CS}$ a nível baixo.Depois disso, a instrução de escrita é então enviada para o controlador CAN, seguido pelo endereço e com pelo menos um byte de dados. É possível gravar registos sequenciais através de geração de impulsos de relógio, isto enquanto o $\overline{CS}$ está em nivel baixo. Os dados serão escritos no registo no início do disparo da linha SCK para o bit D0. Se a linha $\overline{CS}$ é colocada em nível alto antes dos 8 bits serem carregados, a gravação será interrompida. É mostrado na figura 5-5 uma ilustração mais detalhada da sequência de escrita do byte.

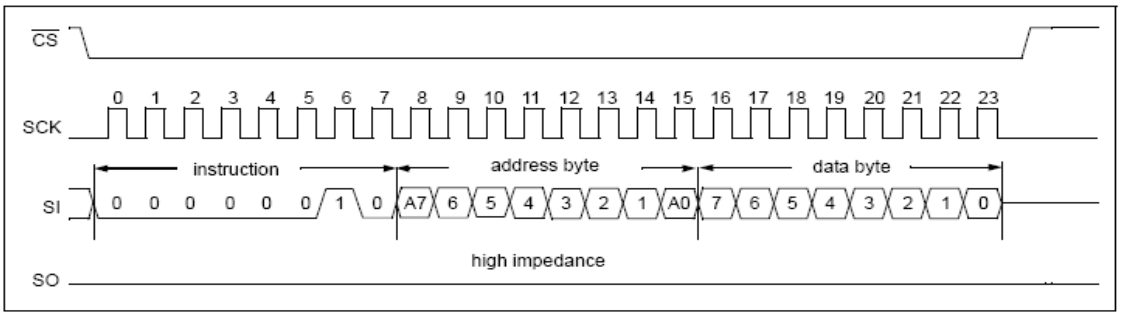

Figura 5- 14 Instrução de Escrita [2].

5.1.2.3 Instrução de *Request to Send* (RTS)

O comando RTS pode ser usado para iniciar a mensagem de transmissão de ou ou mais buffers de transmissão. Começa-se pelo $\overline{CS}$ a nível baixo e o byte de comando RTS é enviado para o MCP2510.Os últimos 3 bits deste comando indica qual dos buffers de transferência está

habilitado a enviar, como é mostrado na figura 5-6. Este comando irá definir o bit de TxBnCTRL.TXREQ para o respectivo buffer. Qualquer um ou todos os três últimos bits pode ser definido num único comando. se o comando RTS é enviado com nnn=000, o comando será ignorado.

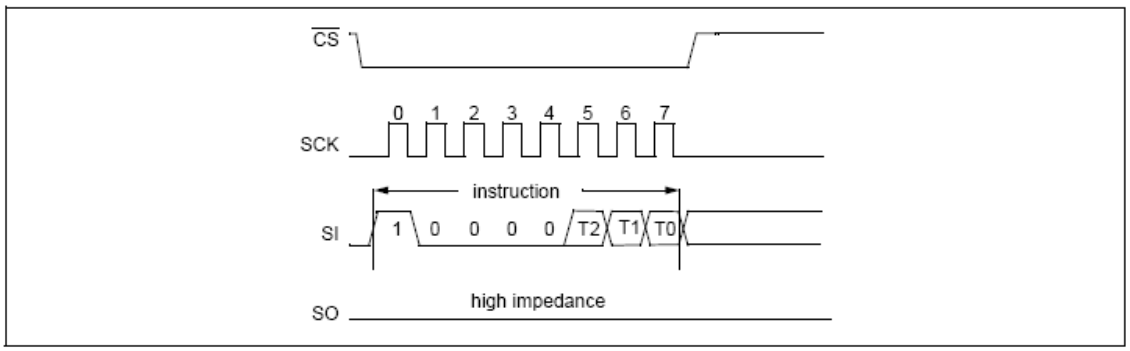

Figura 5- 15 Instrução de *Request to send* [2].

5.1.2.4 Instrução de leitura de *status*

A instrução de leitura de status permite através de uma única instrução o acesso a alguns dos bits usados para mensagens de transmissão e recepção. Esta parte é seleccionada pondo o $\overline{CS}$ em nível baixo, e o comando do status de leitura é enviado para o MCP2510. Após o envio do byte de comando, o controlador retoma 8 bits de dados que contêm o estado. Se os clocks adicionais sao enviados após o primeiro dos 8 bits transmitido, o MCP2510 continuará a fazer o output dos bits de status enquanto o pino $\overline{CS}$ está em nível baixo e os clocks estão providos no SCK. Cada bit de estados retornado neste comando também deve ser lido usando o comando padrao de leitura com o endereço de registo apropriado. A representação esquemática encontra-se presente na figura 5-7.

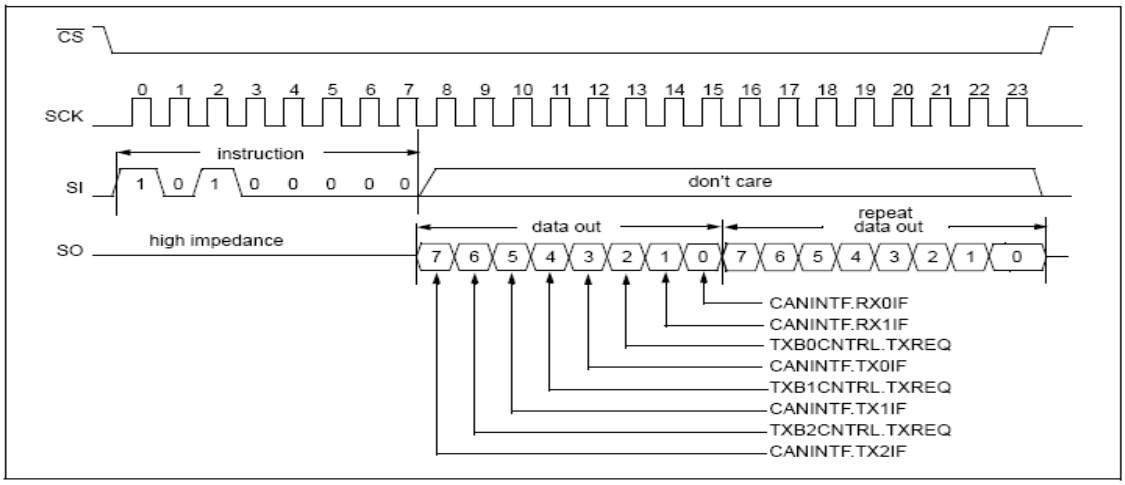

Figura 5- 16 Instrução de leitura de Status [2].

5.1.2.5 A Instrução *Bit Modify*

Esta parte é seleccionada pondo o $\overline{CS}$ a nivel baixo e o byte modify command bit é enviado para o MCP2510. Depois do byte de comando enviado, o endereço para o registo é enviado seguido pelo byte mascara e pelo byte de dados. O byte máscara determina que bits no registo serão permitidos mudar. Um "1" no byte mascara permitirá um bit no registo para mudar, enquanto que o "0" não. O byte de dados determina para que valor deveram mudar os bits modificados. Um "1" no byte de dados habilitara o bit e o "0" irá desabilitá-lo, desde que a máscara desse bit seja activa a "1".

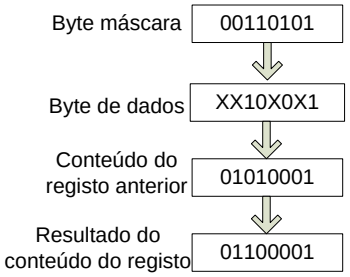

Figura 5- 17 Instrução *Bit Modify* [2].

5.1.2.6 A instrução *Reset*

A instrução reset pode ser usada para reinicializar os registos internos do MCP2510 e habilitar o modo de configuração. Este comando fornece a mesma funcionalidade, através do interface SPI, como o pino $\overline{RESET}$. A instrução de RESET é um byte único de instrução que requer a selecção do dispositivo pondo o $\overline{CS}$ a nível baixo, enviando o byte de instrução, e habilitando logo de seguida o $\overline{CS}$ (figura 5-9).

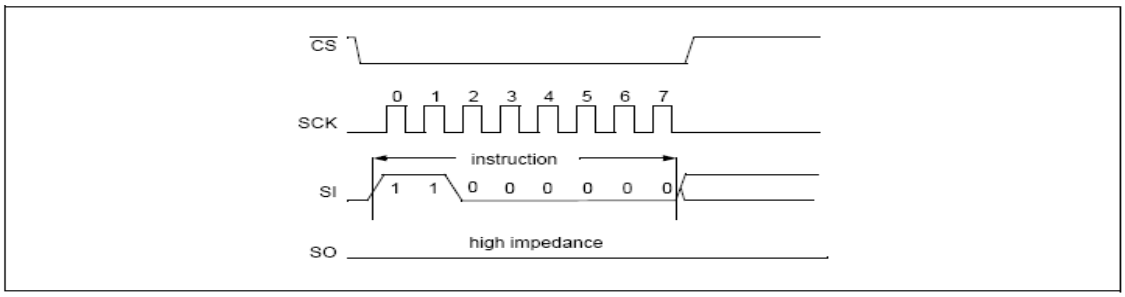

Figura 5- 18 Instrução Reset [2].

6 CONCLUSÕES

6.1 Conclusões

Com o desenvolvimento deste trabalho, foi criado um modelo de sistema de intrusão para espaços residenciais com uma grande flexibilidade operacional e modularidade. Através disso, foram definidos os módulos nessecários para o correcto funcionamento do sistema.
Os modelos de dispositivos concebidos têm funções bem definidas: (i) central de alarme; (ii) sensores associados à central e (iii) respectivo teclado para interacção exterior. Estes cooperam entre si fazendo com que o sistema funcione de forma completamente autónoma, com uma reduzida ou completa ausência de intervenção por parte do utilizador. Para o funcionamento de cada dispositivo e do sistema global ser o mais correcto possível, foram criados os respectivos diagramas de blocos e fluxogramas, com as respectivas entradas e saídas e comunicação bem definidas pela norma CANopen.
Com a evolução do trabalho proposto, foram criados os respectivos dicionários de objectos e EDS (*Electronic Data Sheet*), necessários para a comunicação entre os dispositivos.

6.2 Trabalhos Futuros

A conclusão desta dissertação, pode constituir um princípio para novos trabalhos desenvolvidos. A partir deste, poderá ser desenvolvido todo o harware e software, mais concretamente em relação às interfaces de accionamento e monitorização.
Visto que neste trabalho apenas se tratou de desenvolver uma base ligada à área da segurança e intrusão, seria interessante e útil desenvolver um outro sistema de controlo que interagisse com o sistema proposto. Com a troca de informações entre estes dois sistemas, seria possível ter ainda uma maior interacção com todo o sistema em si, permitindo um leque de opções mais alargado.

BIBLIOGRAFIA

BIBLIOGRAFIA

[1] B.Abreu; "*Concepção de um Modelo de Sistema Para a Gestão da Iluminação Interior em Espaços Residenciais* "; Tese de Mestrado para a Obtenção do Grau de Mestre em Engenharia Electromecânica, Universidade da Beira Interior, Covilhã, 2009.

[2] "*Stand-Alone CAN Controller with SPI Interface*"; Microchip Technology Inc. http://www.microship.com

[3] http://www.microsegur.com

[4] http://www.siemens.pt

[5] B.Ribeiro; "*Um Sistema Distribuído para a Automação de Espaços Residenciais e de Serviços*"; Tese de Doutoramento para a Obtenção do Grau de Doutor em Engenharia Electrotécnica, Universidade da Beira Interior, Covilhã, 2008.

[6] "MSP430x2xxx *Family, Mixed Signal Products*"; Texas Instruments, 2008.

[7] W.Lawrenz; "*CAN Systems Engineering: From Theory to Pratical Applications*"; Spring, New York, 1997.

[8] M.Farsi, M.Barbosa; "*CANopen Implementations: applications to industrial networks*"; RSP-Research Studies Press, Baldock, 2000.

[9] http://www.unibratec.com.br/jornadacientifica

[10] "Octal High Voltage, High Current Darlington Transistor Arrays, ULN2803/ULN2804,"; Motorola, Inc.1996.

[11] Samuel Avelar; "*Protocolo EIB KNX*", Instituto Superior de Engenharia do Porto, Dezembro de 2007.

Anexo A. OUTRAS TECNOLOGIAS UTILIZADAS

A.1. European Installation Bus – EIB

O sistema EIB foi desenvolvido com o objectivo de se construir um sistema de gestão na área de instalações eléctricas. Actualmente é suportado pela EIBA8, e é baseado na técnica de acesso ao meio CSMA/CA. Esta técnica permite que qualquer nó da rede seja mestre, ou seja, a qualquer momento qualquer um dos nó pode tentar aceder ao barramento quando este estiver livre. O tratamento de colisão surge da mesma forma que o protocolo BDLC (*Burroughs Data Link Control*).

O sistema EIB suporta redes de estrutura hierárquica, constiuida por nós, sub-redes, áreas e sistema geral. Cada sub-rede pode ter até 256 nós, cada área pode ter até 15 sub-redes e o sistema pode ter até 15 áreas. Isso totaliza (255 x 16) x 15 + 255 = 61455 nós por sistema. A topologia lógica EIB pode ser visualizada na Figura A.1.

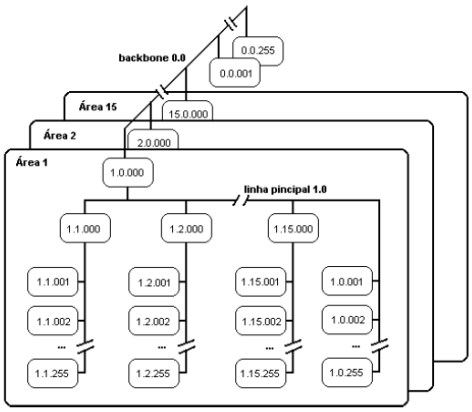

Figura A- 1 Topologia Lógica de um Sistema EIB.

O pacote de transmissão de dados EIB é mostrado na Figura A.2. Pode ainda acrescentar-se o facto em que o campo de endereço possui tanto o endereço do(s) destinatário(s) como do remetente do pacote.

Figura A- 2 Pacote EIB.

Os meios físicos de comunicação disponíveis são:

• EIB.TP – Par Entrelaçado. A taxa de transferência é de 9600 Bit/s sendo o comprimento máximo por sub-rede de 1000m.

• EIB.PL – Linha de Energia. Utiliza uma modulação SFSK (*Spread Frequency Shift Keying*) sendo a distância máxima entre dois dispositivos, sem necessidade do uso de repetidores, de de 600m devido ao alto nível de ruído que este meio possui.

• EIB.RF – Rádio Frequência. Sem retransmissores pode-se obter um alcance de 300m. O ponto forte desse protocolo e o método de acesso ao meio, que possibilita uso em aplicações críticas em tempo real. É conhecido por ser um protocolo mais restrito à Europa e à taxa de transferência de 9600bps. Equipamentos que utilizam este protocolo são em torno de 10 a 100 vezes mais caros que outros equipamentos similares. Na figura A-3, é representado o modo como os dispostivos interagem com o barramento [11].

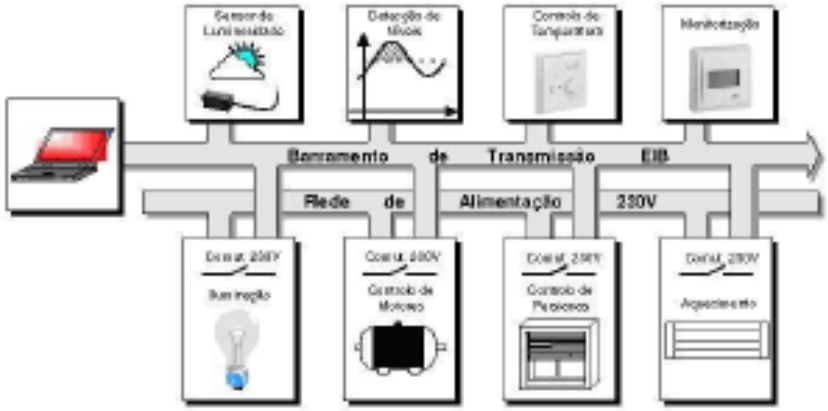

Figura A- 3 Barramento de transmissão EIB [11].

A.2. EIB/KNX

O sistema Instabus EIB/KNX é um sistema de gestão técnica (produzido pela Siemens) para edifícios e habitação baseado na tecnologia KNX, o primeiro *standard* de comunicação aberto, aprovado de acordo com a norma europeia, EN 50090. O standard KNX é baseado na comunicação inicial do EIB com a integração posterior de novos modos de configuração, nomeadamente dos sistemas BatiBUS e EHS. A norma EN 50090, para sistemas electrónicos de residências e edifícios, permitem o controlo de diferentes funções e aplicações em residências e edifícios novos ou existentes entre equipamentos de diferentes fabricantes, que desenvolvem suluções competitivas, flexíveis e adaptadas às necessidades do mercado.

Os diferentes fabricantes de produtos KNX, podem, deste modo, diferenciar os seus equipamentos com mais modos de configuração e, simultaneamente, alargar a gama de produtos. Outro aspecto interessante da aplicação deste sistema em residências é a possibilidade de se aceder facilmente à instalação a partir de qualquer ponto do mundo,

através da internet. Nos edifícios modernos de uso residencial e escritórios, é cada vez mais frequente o uso de sistemas de comando e vigilância, para satisfazer a exigência de maior segurança e conforto. Através da linha de *bus* de expansão radial, pode-se transmitir todas as informações sem restrições, conseguindo-se com esta solução reduzir o número de cabos e condutores, os custos de instalação e os riscos de incêndio.

A.2.1 Características técnicas do sistema Instabus EIB/KNX

A.2.1.1. Aparelhagem base

É sempre necessária para cada linha de *bus* uma fonte de alimentação, que alimente a linha de *bus* através de um filtro indutivo incorporado, sendo a aparelhagem modular e preparada para ser instalada em calha simétrica DIN. A linha de *bus* é um cabo simétrico de dois pares, e é distribuída e ampliada de forma radial, não necessitando de qualquer tipo de resistência final. Permite, deste modo, a comunicação simultânea entre todos os participantes do *bus*. Cada linha de *bus* poderá ter um comprimento máximo de 1000 metros e tem a capacidade máxima de 64 participantes de *bus*. As linhas podem ser interligadas entre si por acopladores de linha e de grupo, permitindo assim a expansão do sistema até 180 linhas e mais de 12.000 participantes [4].

O acoplador de linha actua como filtro de dados, interceptando tramas de dados dos diversos "participantes" da sua linha de *bus* e dando continuidade às tramas destinadas a outras linhas através da redução de carga de informação simultânea nas linhas de *bus* do sistema. Estes acopladores são instalados em quadro eléctrico.

A.2.1.2. Saídas binárias

As saídas binárias são contactos livres de tensão, com um acoplador de *bus* integrado (memória parametrizável), que operam consoante as informações enviadas através da linha de *bus*, actuando directamente sobre os circuitos de potência. Os contactos das saídas binárias são parametrizáveis individualmente e podem actuar com temporizações de cada acção até 150 horas ou através de combinações lógicas *AND* ou *OR*.

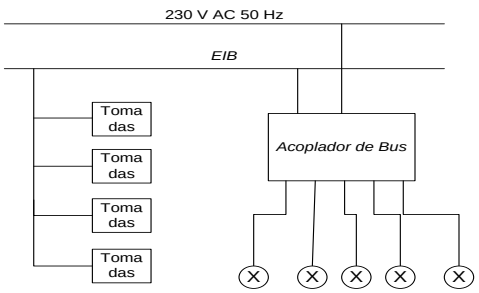

Figura A- 4 Exemplo de um acoplador de *bus* integrado [4].

A.3. European Home Systems - EHS

O protocolo de comunicações EHS nasceu através da globalização de empresas europeias ligadas à area de aparelhos electrodomésticos. A partir deste ponto, criou-se um protocolo aberto com um vasto leque de aplicações, permitindo assim que equipamentos de diferentes fabricantes se comuniquem para que possam compartilhar recursos. O modelo de comunicação EHS especifica a camada física, de enlace, de rede e de aplicação, conforme se mostra na figura A.5.

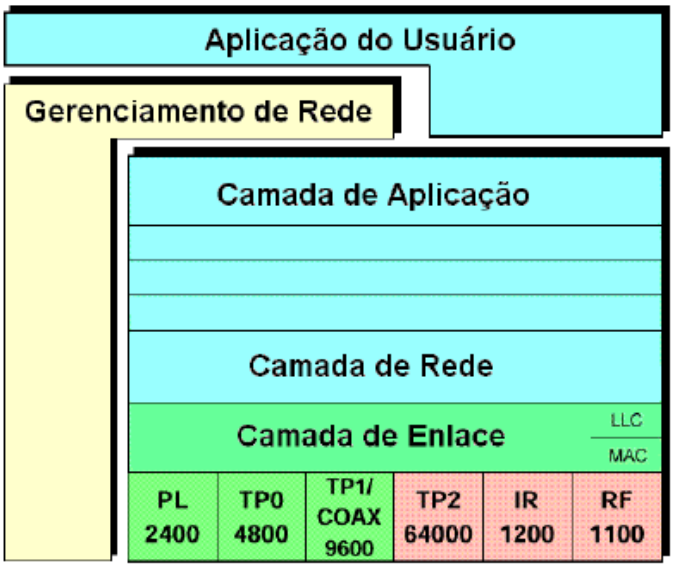

Figura A.5 Arquitectura do protocolo EHS.

A camada de aplicação traduz a "linguagem" da aplicação em pacotes de dados capazes de circular na rede. A camada de rede está relacionada com o rotulamento e endereçamento de pacotes. A camada de enlace, dividida em MAC e LLC, gere a conversão de bits, regras de acesso a rede, recepção e envio de pacotes e mecanismos de repetição. Várias camadas físicas estão definidas devido ao grande número de aplicações que o protocolo abrange. A

rede eléctrica, os infra-vermelhos e rádio podem ser usados como canal de comunicação de baixa velocidade sem a necessidade de cabeamento extra.
Um exemplo seria o controlo de aquecedores, ar condicionado e accionamentos remotos em geral. O Par traçado e cabo coaxial podem ser utilizados sempre que se requer alta velocidade, por exemplo, para aplicações de vídeo, áudio e segurança. As características de cada meio físico suportado pelo protocolo são mostradas na Tabela A.1.

Características do Protocolo:

- Plug and Play.
- Interoperabilidade.
- Expansão e configuração automática.

Meios Físicos:

Uma parte importante de um sistema de automação doméstico é o meio de comunicação. A especificação EHS, versão 1.2, cobre seis meios para transportar informações sendo que outros meios ainda poderão vir a ser acrescentados.

Tabela A. 1 Tipos de meios físicos EHS [9].

Meio Físico	**Par Traçado tipo1 TP1**	**Par Traçado tipo1 TP2**	**Cabo Coaxial CX**	**Linha de Energia PL**	**Rádio RF**	**Infra-Vermelhos**
Aplicação	Propósito geral, controlo	Telefonia, ISDN, dados, controlo	Áudio, Vídeo, TV, dados, controlo	Controlo	Telefone sem fios, controlo	Controlo remoto
Taxa de Transmissão	9.6 Kbps	64 Kbps	9.6 Kbps	2.4Kbps	1.2 Kbps	1.1 Kbps
Acesso	CSMA/CA	CSMA/CD	CSMA/CA	CSMA/ack	CT2	-
Alimentação	35V	35V	15V	230Vac	-	-
Codificação	-	TDM	FDM	-	FDM	-
Topologia	Livre	Barramento	Barramento	Livre	Livre	Livre
Unidades	128	40	128	256	256	256
Alcance	500m	300m	150/50m	Casa	50/200m	Sala

Os meios mais importantes para o protocolo são a linha de energia e o par traçado (TP1).
Num sistema onde o custo é a principal prioridade, o uso de linha de energia tem uma grande vantagem em relação a outros meios.
Não é necessário cabeamento extra, pois todas as casas possuem um cabeamento de rede eléctrica.

A.4. X-10

Os componentes básicos deste sistema são controladores e módulos receptores. Os controladores e módulos receptores podem ser escolhidos à medida de cada aplicação, implicando as ampliações ou modificações apenas o custo dos novos módulos.Na figura seguinte representa-se um exemplo de funcionamento prático do protocolo X-10.

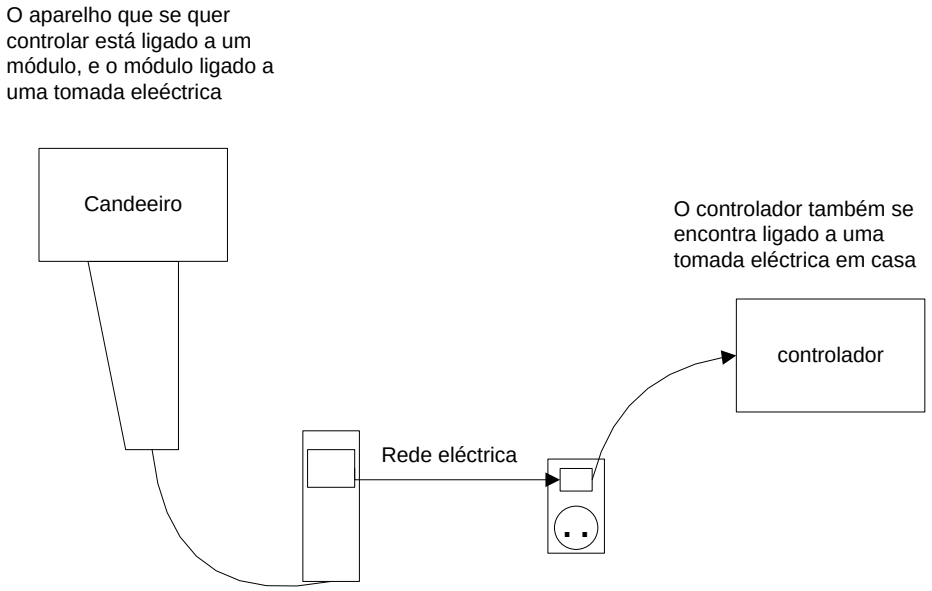

Figura A- 5 Exemplo de componentes utilizados no protocolo X-10.

Os módulos ligam-se às tomadas já exisitentes e enviam sinais pela rede eléctrica, permitindo ligar e desligar luzes, aparelhos ou motores espalhados pela mesma fase de corrente a partir de qualquer controlador, bastando apenas para entrar em funcionamento, ligar os módulos às tomadas. O simples facto de não necessitar de fios e de usar a rede eléctrica já existente permite instalar o sistema sem ser necessário fazer obras para passar fios. Esta facilidade permite a sua rápida intalação em casas já construídas e alterações, em qualquer altura, sem custos adicionais.

A.4.1 Descrição Técnica da Tecnologia X-10

Tecnicamente, o X-10 comunica pela rede eléctrica enviando sinais binários modulados em amplitude *amplitude shift keyed* – ASK numa portadora de 120 KHz. O envio da portadora é feito de uma forma sincronizada com 1 milisegundo (ms) após a passagem do sinal de corrente alternada (50 Hz) da corrente eléctrica [9].

O envio das mensagens está sincronizado com a passagem do sinal de corrente alterna (50 Hz) por zero e tem várias vantagens técnicas e económicas, indicado de seguida:

- O ponto de passagem pelo zero tem menos ruído e menos interferências de outros aparelhos ligados à rede elétrica. Reóstatos, motores e outras fontes de ruído tendem a injectar pequenos sinais de altas frequências nos pontos de tensão mais altos da curva sinusoidal. Além disso, a impedância instantânea da rede eléctrica de corrente alterna tem geralmente o seu ponto mais alto precisamente na passagem pelo ponto zero.
- Permite que o sinal da mensagem seja transmitido num local preciso da curva de correne alterna. Desta forma, o circuito electrónico do receptor pode estar preparado para receber apenas dados durante um determinado período de tempo em que as interferências estão reduzidas ao mínimo, dminuindo assim as probabilidades de se receberem mensagens falsas.
- Permite, de forma mais económica, sincronizar os tempos de aquisição entre receptor e transmissor. Usando como referência pontos na curva da correnter alterna, os tempo de transmissão e recepção estão sempre balizados. Não é, portanto, necessário fazer alterações de *hardware* dos produtos para os utilizar em países co linhas de frequência da corrente eléctrica diferentes (caso dos Estados Unidos que usa 60 Hz, enquanto que na Europa se usa 50 Hz).

Existem, no entanto, algumas desvantagens em usar a passagem do sinal de corrente alterna pelo zero para sincronizar o envio de mensagens. Indicam-se de seguida alguns exemplos:

- A velocidade de transmissão não pode ser superior à da linha de frequência, uma vez que é enviado apenas um bit por cada ciclo.
- Em aplicações trifásicas, a curva sinusoidal da corrente alternada de cada ramo das três fases está 120º em avanço ou atraso relativamente às outras duas. Uma vez que o controlador vai estar numa das fases, o módulo receptor poderá estar numa das outras, a passagem pelo zero da corrente alternada pode ocorrer desfasada e a mensagem não ser recebida.

A.4.2 Descrição das mensagens X-10

Cada mensagem básica é constituída por um sinal de 13 bits: 4 bits para o sinal de início de comunicação (*start code*), 4 bits para o código de casa (*house code*) e 5 bits oara o código unidade/função (*function code*). A mensagem precisa de 22 ciclos de 50 Hz da corrente alternada para que se conclua a sua transmissão.

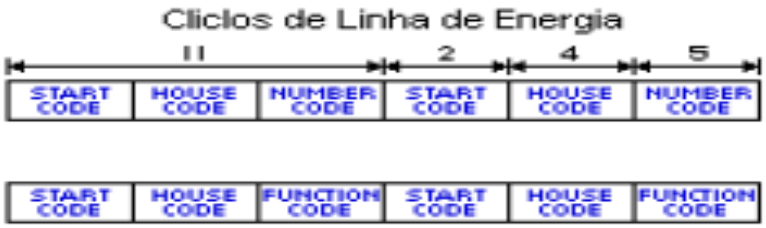

Figura A. 6 Pacote do protocolo X-10.

Os primeiros 4 bits identificam o campo correspondente ao início da comunicação (*start-code*). A transmissão deste pacote é especial, uma vez que utiliza cada passagem do sinal de corrente alternada (50 Hz) pelo zero (positiva e negativa) para enviar 1 bit, enquanto que nos pacotes de informação se transmite apenas um bit por cada ciclo de 50 Hz. Desta forma, este pacote usa o mesmo número de ciclos que 2 bits "regulares".
O campo do código de casa (*house code*) é transmitido durante cada mensagem e os 4 bits contêm o código do controlador que está a enviar a ordem de comando.
O campo do código de unidade/ função (*number code*), contém ou o código da unidade (receptor) ou a função a executar (ordem de comando). O campo *function code* determina o propósito do campo precedente, ou seja, se for um "0" lógico, os 4 bits unidade/função contêm o endereço da unidade; se for um "1" lógico, contêm a função.

Os bits que compõem os campos dos códigos de casa e unidades são transmitidos do bit menos significativo (LSB) para o bit mais significativo (MSB). Os 4 bits refenrentes ao código de casa são sempre enviados juntamente com o pacote de dados do código de unidade/função.

Na figura A.28, estão descritas as várias ordens do código de função. Os primeiros sete códigos são basicamente os que estão definidos nas teclas dos controladores. Os restantes nove são códigos extra, usados selectivamente pelos diversos produtos X-10. Resumidamente, apresenta-se de seguida a descrição de cada um deles:

- O *extended code* (código ampliado) e os *extended data* (dados ampliados) são usados para adicionar mais bytes de informação ao pacote de dados básico.

- O código *hail request* (pedido de confirmação) permite que um módulo ou controlador autoconfigure o seu código de casa, ou seja, que adquira um código de casa não utilizado. Um pacote de código casa/função é transmitido com o código de casa desejado, usando o código *hail request*. Se existir um módulo ou controlador a "acusar" a recepção deste código, este responderá usando o código de casa e o códgo *hail request*, permitindo ao utilizador, por exemplo, ter a certeza de que está a escolher um código de casa que não está a ser utilizado por um vizinho.
- O código *pre-set dim* define um determinado nível de intensidade luminosa para um módulo de lâmpada, sem ter de enviar vários códigos de sinal mais brilho (*bright*) ou sinal menos brilho (*dim*).
- O código *status request* pede uma resposta simples de ligado/desligado a um módulocapaz de exercer esta função.

HOUSE CODES					KEY CODES					
	H1	H2	H4	H8		D1	D2	D4	D8	D16
A	0	1	1	0	1	0	1	1	0	0
B	1	1	1	0	2	1	1	1	0	0
C	0	0	1	0	3	0	0	1	0	0
D	1	0	1	0	4	1	0	1	0	0
E	0	0	0	1	5	0	0	0	1	0
F	1	0	0	1	6	1	0	0	1	0
G	0	1	0	1	7	0	1	0	1	0
H	1	1	0	1	8	1	1	0	1	0
I	0	1	1	1	9	0	1	1	1	0
J	1	1	1	1	10	1	1	1	1	0
K	0	0	1	1	11	0	0	1	1	0
L	1	0	1	1	12	1	0	1	1	0
M	0	0	0	0	13	0	0	0	0	0
N	1	0	0	0	14	1	0	0	0	0
O	0	1	0	0	15	0	1	0	0	0
P	1	1	0	0	16	1	1	0	0	0
					All Units Off	0	0	0	0	1
					All Lights On	0	0	0	1	1
					On	0	0	1	0	1
					Off	0	0	1	1	1
					Dim	0	1	0	0	1
					Bright	0	1	0	1	1
					All Lights Off	0	1	1	0	1
					Extended Code	0	1	1	1	1
					Hail Request	1	0	0	0	1
					Hail Acknowledge	1	0	0	1	1
					Pre-Set Dim	1	0	1	X	1
					Extended Data (analog)	1	1	0	0	1
					Status=on	1	1	0	1	1
					Status=off	1	1	1	0	1
					Status Request	1	1	1	1	1

Figura A-7 Tabela de comando do protocolo X-10.

Anexo B. DICIONÁRIO DE OBJECTOS

B.1 Dicionário de Objectos da Central de Alarme

Tabela B-1 Dicionário de objectos para o perfil de comunicações standard da Central de Alarme.

Índice (hex)	Sub-índice	Nome	Dados/Objectos	Atr.	Default	PDO Map	Comentários
ÁREA DO PERFIL DE COMUNICAÇÕES STANDARD DSP301							
1000	.	Device type	U32	Ro	0x00030000	N	DSP 0x000 Digital outputs and inputs=0x0003
1001	.	Error register	U8	ro	0	S	Erros indicados em DSP301
1002	.	Manufactured status register	U32	ro	0	N	
1004	.	#PDOs supported	ARRAY				
	0	#PDOs supported	U32	ro	0x00050001	N	5RPDO e 1 TPDO
	1	#PDOs sync	U32	ro	0x00000000	N	0RPDO e 0TPDO
	2	#PDO async	U32	ro	0x00050001	N	5RPDO e 1 TPDO
1005	.	COB-ID SYNC message	U32	rw	0x00000080	N	SYNC não usada
1006	.	Communication cycle period	U32	rw	0	N	Não utilizado
1007	.	Synchronous window lenght	U32	rw	0	N	Não utilizado
1008	.	Manufactured device name	VisStr	c	CA	N	CA - Central de Alarme
1009	.	Man. harware version	VisStr	c	0	N	
100A	.	Man. Software version	VisStr	c	0	N	
100F	.	#SDOs supported	U32	ro	0x00000001	N	0 SDO client e 1 SDO server
1010	.	Store parameters	ARRAY				Guarda os parâmetros em memória não volátil
	0	Largest supported sub index	U8	ro	4	N	
	1	Save all parameter	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	2	Save comm. Parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	3	Save app. Parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	4	Save man. Parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
1011	.	Restore default parameters	ARRAY				
	0	Largest supported sub index	U32	ro	4	N	
	1	Restore all parameters	U32	rw	0x00000001	N	Restaura parâmetros
	2	Restore comm. Parameters	U32	rw	0x00000001	N	Restaura parâmetros
	3	Restore app. Parameters	U32	rw	0x00000001	N	Restaura parâmetros
	4	Restore man. Parameters	U32	rw	0x00000001	N	Restaura parâmetros
1400	.	1 RPDO parameter	RECORD				RPDO1 comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	200h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
1401	.	2 RPDO parameter	RECORD				RPDO2 comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	200h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
1402	.	3 RPDO parameter	RECORD				RPDO3 comm. Parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	200h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
1403	.	4 RPDO parameter	RECORD				RPDO4 comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	200h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
1404	.	5 RPDO parameter	RECORD				RPDO5 comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	200h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
1800	.	1 TPDO parameter	RECORD				TPDO comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	180h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono

Tabela B-2 Dicionário de Objectos para o perfil específico do fabricante da Central de Alarme.

Índice (hex)	Sub-índice	Nome	Dados/Objectos	Atr.	Default	PDO Map	Comentários
ÁREA DO PERFIL DE DISPOSITIVO STANDARD							
ÁREA DO PERFIL ESPECÍFICO DO FABRICANTE							
200D	.	Estado do modo de funcionamento	U8	rw	0x02	S	Off=0x00 On=0x01 Automático=0x02
200E	.	Estado da zona de funcionamento	U8	rw	0x00	S	Off=0x00 On=0x01 Automático=0x02
200F		Função da zona seleccionada	U8	rw	0x00	S	Off=0x00 On=0x01
2010		Descriptor da zona seleccionada	U8	rw	0x00	S	Off=0x00 On=0x01
2011		Permissões da zona seleccionada	U8	rw	0x00	S	Off=0x00 On=0x01
2012	.	Estado dos Sensores	U8	rw	0x00	S	Min=0%=0x00; Max=100%=0x64
2013	.	Estado definido no Teclado	U8	rw	0x00	S	Off=0x00 On=0x01
2014	.	Estado Utilizador	U8	rw	0x00	S	Off=0x00 On=0x01

B.2 Dicionário de Objectos para os Sensores da Central de Alarme.

Tabela B-3 Dicionário de Objectos para os Sensores da Central de Alarme.

Índice (hex)	Sub-índice	Nome	Dados/Objectos	Atr.	Default	PDO Map	Comentários
ÁREA DO PERFIL DE COMUNICAÇÕES STANDARD DSP301							
1000	.	Device type	U32	ro	0x00030000	N	DSP 0x000 Digital outputs and inputs=0x0003
1001	.	Error register	U8	ro	0	S	Erros indicados em DSP301
1002	.	Manufactured status register	U32	ro	0	N	
1004	.	#PDOs supported	ARRAY				
	0	#PDOs supported	U32	ro	0x00050001	N	0RPDO e 1 TPDO
	1	#PDOs sync	U32	ro	0x00000000	N	0RPDO e 0TPDO
	2	#PDO async	U32	ro	0x00050001	N	0RPDO e 1 TPDO
1005	.	COB-ID SYNC message	U32	rw	0x00000080	N	SYNC não usada
1006	.	Communication cycle period	U32	rw	0	N	Não utilizado
1007	.	Synchronous window lenght	U32	rw	0	N	Não utilizado
1008	.	Manufactured device name	VisStr	c	SCA	N	SCA - Sensores da Central de Alarme
1009	.	Man. harware version	VisStr	c	0	N	
100ª	.	Man. Software version	VisStr	c	0	N	
100F	.	#SDOs supported	U32	ro	0x00000001	N	0 SDO client e 1 SDO server
1010	.	Store parameters	ARRAY				Guarda os parâmetros em memória não volátil
	0	Largest supported sub index	U8	ro	4	N	
	1	Save all parameter	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	2	Save comm. parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	3	Save app. parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	4	Save man. parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
1011	.	Restore default parameters	ARRAY				
	0	Largest supported sub index	U32	ro	4	N	
	1	Restore all parameters	U32	rw	0x00000001	N	Restaura parâmetros
	2	Restore comm. parameters	U32	rw	0x00000001	N	Restaura parâmetros
	3	Restore app. parameters	U32	rw	0x00000001	N	Restaura parâmetros
	4	Restore man. parameters	U32	rw	0x00000001	N	Restaura parâmetros
1800	.	1 TPDO parameter	RECORD				TPDO comm. Parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	180h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
ÁREA DO PERFIL DE DISPOSITIVO STANDARD							
ÁREA DO PERFIL ESPECÍFICO DO FABRICANTE							
2002	.	Estado da Central de Alarme	U8	rw	0x0A	S	Off=0x00 On=0x01

B.3 Dicionário de Objectos do Teclado.

Tabela B-4 Dicionário de Objectos do Teclado.

Índice (hex)	Sub-índice	Nome	Dados/Objectos	Atr.	Default	PDO Map	Comentários
ÁREA DO PERFIL DE COMUNICAÇÕES STANDARD DSP301							
1000	.	Device type	U32	ro	0x00030000	N	DSP 0x000 Digital outputs and inputs=0x0003
1001	.	Error register	U8	ro	0	S	Erros indicados em DSP301
1002	.	Manufactured status register	U32	ro	0	N	
1004	.	#PDOs supported	ARRAY				
	0	#PDOs supported	U32	ro	0x00050001	N	1RPDO e 1 TPDO
	1	#PDOs sync	U32	ro	0x00000000	N	0RPDO e 0TPDO
	2	#PDO async	U32	ro	0x00050001	N	1RPDO e 1 TPDO
1005	.	COB-ID SYNC message	U32	rw	0x00000080	N	SYNC não usada
1006	.	Communication cycle period	U32	rw	0	N	Não utilizado
1007	.	Synchronous window lenght	U32	rw	0	N	Não utilizado
1008	.	Manufactured device name	VisStr	c	TCA	N	TCA - Teclado da Central de Alarme
1009	.	Man. harware version	VisStr	c	0	N	
100ª	.	Man. Software version	VisStr	c	0	N	
100F	.	#SDOs supported	U32	ro	0x00000001	N	0 SDO client e 1 SDO server
1010	.	Store parameters	ARRAY				Guarda os parâmetros em memória não volátil
	0	Largest supported sub index	U8	ro	4	N	
	1	Save all parameter	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	2	Save comm. parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	3	Save app. parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
	4	Save man. parameters	U32	rw	0x00000002	N	Guarda os parâmetros de forma autónoma
1011	.	Restore default parameters	ARRAY				
	0	Largest supported sub index	U32	ro	4	N	
	1	Restore all parameters	U32	rw	0x00000001	N	Restaura parâmetros
	2	Restore comm. parameters	U32	rw	0x00000001	N	Restaura parâmetros
	3	Restore app. parameters	U32	rw	0x00000001	N	Restaura parâmetros
	4	Restore man. parameters	U32	rw	0x00000001	N	Restaura parâmetros
1400	.	1 RPDO parameter	RECORD				RPDO1 comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	200h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
1800	.	1 TPDO parameter	RECORD				TPDO comm. parameter
	0	Largest sub-index supported	U8	ro	2	N	
	1	COB-ID used by PDO	U32	rw	180h+NODE ID	N	
	2	Transmission type	U8	rw	FE	N	Assíncrono
ÁREA DO PERFIL DE DISPOSITIVO STANDARD							
ÁREA DO PERFIL ESPECÍFICO DO FABRICANTE							
201E	.	Feedback do modo de funcionamento	U8	rw	0x02	S	Off=0x00 On=0x01 Automático=0x02
201F	.	Feedback da zona de funcionamento	U8	rw	0x02	S	Off=0x00 On=0x01 Automático=0x02

Anexo C. AS ELECTRONIC DATA SHEETS

C.1 A EDS da Central de Alarme

```
;-------------------------------------

;---Rui Fonseca Vaz
;---Departamento de Eng. Electromecânica
;---Universidade da Beira Interior
;---Covilhã
;-------------------------------------

[FileInfo]
FileName=CA.EDS
FileVersion=0
Filerevision=0
Description=EDS-File Central de Alarme
CreationTime=10:30AM
CreationDate=29-09-10
CreatedBy=RFV

[DeviceInfo]
VendorName=UBI
VendorNumber=2214
ProductName=Central de Alarme
ProductNumber=1234
RevisionNumber=1
OrderCode=Security - 1234
BaudRate_10=0                   */ Número  de Bits Transmissíveis /*
BaudRate_20=0
BaudRate_50=1
BaudRate_125=0
BaudRate_250=0
```

BaudRate_500=0

BaudRate_800=0

BaudRate_1000=0

SimpleBootUpMaster=0

SimpleBootUpSlave=1

ExtendedBootUpMaster=0

ExtendedBootUpSlave=0

Granularity=8 */ A maior

parte suporta apenas 8 /*

[StandardDataTypes]

0x0001=0

0x0002=0

0x0003=0

0x0004=0

0x0005=1

0x0006=1

0x0007=1

0x0008=0

0x0009=1

0x000A=1

0x000B=0

0x000C=1

0x000D=1

0x000E=1

0x000F=0

0x0020=1

0x0021=1

0x0022=1

0x0023=1

[DummyUsage]

Dummy0001=0

Dummy0002=0

Dummy0003=0

Dummy0004=0

Dummy0005=0

Dummy0006=0

Dummy0007=0

[MandatoryObjects]

SupportedObjects=2

1=0x1000

2=0x1001

[1000]

Subnumber=0

ParameterName=device type

ObjectType=0x07

DataType=0x0007

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=0x00030000

PDOMapping=0

[1001]

SubNumber=0

Parametername=error register

ObjectType=0x07

DataType=0x0005

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=1

[OptionalObjects]

SupportedObjects=17

1=0x1002

2=0x1004

3=0x1005

4=0x1006

5=0x1007

6=0x1008

7=0x1009

8=0x100A

9=0x100F

10=0x1010

11=1x1011

12=0x1400

13=0x1401

14=0x1402

15=0x1403

16=0x1404

17=0x1800

[1000]

Subnumber=0

ParameterName=device type

ObjectType=7

dataType=0x0007

DefaultValue=0x30000

AcessType=ro

PDOMapping=0

[1001]

SubNumber=0

ParameterNumber=error register

ObjectType=0x07

DataType=0x0005

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=

PDOMapping=1

[1002]

SubNumber=0

ParameterNumber=manufactured status register

ObjectType=0x07

DataType=0x0007

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=

PDOMapping=0

[1004]

SubNumber=2

ParameterName=number of PDOs supported

[1004sub0]

ParameterName=number of PDOs supported

ObjectType=0x0008

DataType=0x07

AcessType=ro

DefaultValue=0x50001

PDOMapping=0

[1004sub1]

ParameterName=number of synchronous PDOs

ObjectType=0x0008

DataType=0x07

AcessType=ro

DefaultValue=0x00000

PDOMapping=0

[1004sub2]

ParameterName=number of assynchronous PDOs

ObjectType=0x0008

DataType=0x07

AcessType=ro

DefaultValue=0x50001

PDOMapping=0

[1005]

SubNumber=0

ParameterNumber=communication cycle period

ObjectType=0x07

DataType=0x0007

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1006]

SubNumber=0

ParameterName=communication cycle period

ObjectType=0x07

DataType=0x0007

AcessType=rw

LowLimit=

HighLimit=

DefaultValue00x00

PDOMapping=0

[1007]

SubNumber=0

ParameterNumber=synchronous window length

ObjectType=0x07

DataType=0x0007

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1008]

SubNumber=0

ParameterNumber=manufactured device name

ObjectType=0x07

DataType=0x0009

AcessType=const

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1009]

SubNumber=0

ParameterNumber=manufactured hardware version

ObjectType=0x07

DataType=0x0009

AcessType=const

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[100A]

SubNumber=0

ParameterNumber=manufactured software version

ObjectType=0x07

DataType=0x0009

AcessType=const

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[100F]

SubNumber=0

ParameterNumber=number of SDO´s supported

ObjectType=0x07

DataType=0x07

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=0x00000001

PDOMapping=0

[1010]

SubNumber=4

ParameterName=store parameters

[1010sub0]

ParameterName=largest supported Subindex

ObjectType=0x0008

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=0x04

PDOMapping=0

[1010sub1]

ParameterName=save all parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1010sub2]

ParameterName=save communication parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1010sub3]

ParameterName=save aplication parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1010sub4]

ParameterName=save manufactured defined

parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1011]

SubNumber=4

Parametername=restore default parameters

[1011sub0]

ParameterName=largest supported Subindex

ObjectType=0x0008

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=0x04

PDOMapping=0

[1011sub1]

ParameterName=restore all default parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1011sub2]

ParameterName=restore communication default

parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1011sub3]

ParameterName=restore application default

parameters

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1011sub4]

ParameterName=restore manufactured define default

ObjectType=0x0008

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=0

[1400]

SubNumber=3

ParameterName=RPDO 1 Communication Parameter

[1400sub0]

ParameterName=number of supported entries

ObjectType=0x0009

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=0

[1400sub1]

ParameterName=COB-ID used by PDO

ObjectType=0x0009

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[1400sub2]

ParameterName=transmission type

ObjectType=0x0009

DataType=0x05

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0xfe

PDOMapping=0

[1401]

Subnumber=3

ParameterName=RPDO 2 Communication Parameter

[1401sub0]

ParameterName=number of supported entries

ObjectType=0x0009

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=0

[1401sub1]

ParameterName=COB-ID used by PDO

ObjectType=0x0009

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[1401sub2]

ParameterName=transmission type

ObjectType=0x0009

DataType=0x05

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0xfe

PDOMapping=0

[1402]

Subnumber=3

ParameterName=RPDO 3 Communication Parameter

[1402sub0]

ParameterName=number of supported entries

ObjectType=0x0009

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=0

[1402sub1]

ParameterName=COB-ID used by PDO

ObjectType=0x0009

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[1402sub2]

ParameterName=transmission type

ObjectType=0x0009

DataType=0x05

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0xfe

PDOMapping=0

[1403]

Subnumber=3

ParameterName=RPDO 4 Communication Parameter

[1403sub0]

ParameterName=number of supported entries

ObjectType=0x0009

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=0

[1403sub1]

ParameterName=COB-ID used by PDO

ObjectType=0x0009

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[1403sub2]

ParameterName=transmission type

ObjectType=0x0009

DataType=0x05

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0xfe

PDOMapping=0

[1404]

Subnumber=3

ParameterName=RPDO 5 Communication Parameter

[1404sub0]

ParameterName=number of supported entries

ObjectType=0x0009

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=0

[1404sub1]

ParameterName=COB-ID used by PDO

ObjectType=0x0009

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[1404sub2]

ParameterName=transmission type

ObjectType=0x0009

DataType=0x05

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0xfe

PDOMapping=0

[1800]

Subnumber=2

ParameterName=TPDO 1 Communication Parameter

[1800sub0]

ParameterName=number of supported entries

ObjectType=0x0009

DataType=0x05

AcessType=ro

LowLimit=

HighLimit=

DefaultValue=2

PDOMapping=0

[1800sub1]

ParameterName=COB-ID used by PDO

ObjectType=0x0009

DataType=0x07

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0

PDOMapping=0

[1800sub2]

ParameterName=transmission type

ObjectType=0x0009

DataType=0x05

AcessType=rw

LowLimit=

HighLimit=

DefaultValue=0xfe

PDOMapping=0

[ManufacturerObjects]

SupportedObjects=8

1=0x200D

2=0x200E

3=0x200F

4=0x2010

5=0x2011

6=0x2012

7=0x2013

8=0x2014

[200D]

ParameterName=Estado do modo de funcionamento

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x02

PDOMapping=1

[200E]

ParameterName=Estado da zona de funcionamento

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1

[200F]

ParameterName=Função da zona seleccionada

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1

[2010]

ParameterName=Descriptor da zona seleccionada

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1

[2011]

ParameterName=Permissões da zona seleccionada

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1

[2012]

ParameterName=Estado dos sensores

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1

[2013]

ParameterName=Estado definido no teclado

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1

[2014]

ParameterName=Estado utilizador

ObjectType=0x0007

DataType=0x05

acessType=rw

LowLimit=

HighLimit=

DefaultValue=0x00

PDOMapping=1