



INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Aplicação de Interface Com Sistema Domótico EIB

João Pedro Santos Silva

Dissertação para obtenção do grau de Mestre em

Engenharia Informática e de Computadores

Júri

Presidente: Prof. Luís Rodrigues

Orientador: Prof. Renato Nunes

Vogal: Prof. Alberto Cunha

Setembro 2007



Aplicação de Interface com Sistema Domótico EIB

João Pedro Santos Silva

AGRADECIMENTOS

A todos os Professores do Instituto Superior Técnico que contribuíram para a minha formação, em especial ao Professor Renato Nunes por todo o seu apoio no desenvolvimento desta dissertação.

À minha família, amigos e colegas pelo apoio que me proporcionam continuamente.

RESUMO

Esta dissertação, enquadrada na área das tecnologias domóticas, procura solucionar algumas lacunas existentes nesta área. A domótica é uma área em expansão mas que tem apresentado um ritmo de crescimento inferior ao inicialmente esperado. Este facto deve-se em parte à dispersão do mercado pelas várias tecnologias incompatíveis existentes, que vão desde tecnologias proprietárias até a alguns movimentos de uniformização independentes que levaram à criação de diferentes normas, tendo um resultado oposto ao desejado.

Outra limitação actualmente existente nos sistemas domóticos é a inexistência de aplicações genéricas de monitorização e controlo, que permitam ao utilizador modificar o comportamento do sistema. As soluções actualmente existentes são desenhadas para utilização com um determinado sistema e com um comportamento predefinido, implicando tipicamente a aquisição de uma nova versão da aplicação ou a sua alteração por pessoal técnico, sempre que o utilizador pretender qualquer tipo de modificação ao sistema.

A solução apresentada nesta dissertação permite o desenvolvimento de aplicações de monitorização e comando genéricas, para a utilização com diferentes tecnologias. Esta solução tem como base a tecnologia DomoBus, que através do seu modelo de abstracção de dispositivos, e do seu serviço de comunicações, permite o desenvolvimento de aplicações facilmente configuráveis a partir de ficheiros XML, que permitirão monitorizar e controlar redes de dispositivos de outras tecnologias. Esta facilidade de configuração permite a qualquer utilizador modificar o comportamento do seu sistema, recorrendo à utilização de uma ferramenta adequada para o efeito.

Esta dissertação apresenta um protótipo de um gateway entre a tecnologia DomoBus e a tecnologia EIB/KNX, que servirá como ponto de partida para uma solução multi-tecnológica. Com este gateway será possível aplicar os benefícios do DomoBus ao EIB/KNX. A tecnologia EIB/KNX foi escolhida para esta solução devido ao facto de ser uma tecnologia normalizada com grande implantação no mercado e oferecer um elevado grau de funcionalidade que permite a realização de sistemas complexos e abrangentes.

Palavras-chave: Domótica, Interoperação, XML, Interface utilizador, Aplicações monitorização e controlo, Integração tecnológica.

ABSTRACT

This dissertation, in the area of home automation technologies, attempts to solve some of its existing gaps. Home automation is an expanding area but has presented a growing rhythm below the original expectations. This low rhythm is partially due to the market dispersion throughout several incompatible technologies, some of them proprietary and others originated from several uniformization movements which led to the creation of different standards, having an opposite result to what was intended.

Another existing limitation in the home automation systems is the inexistence of generic monitoring and control applications that allow the user to modify the behavior of the system. The existing solutions are designed to be used with specific systems and with a predefined behavior, typically leading to the acquisition of a new version of the application or its modification by technical personnel, every time the user intends to perform any kind of modification to the system.

The solution presented in this dissertation allows developing generic applications for monitoring and control which can be used with different technologies. This solution is based in the DomoBus approach and uses its device abstract model and its communication service, allowing the development of easily configurable applications based on the use of XML files. With this approach, common users can easily modify the behavior of their systems, using an appropriate tool.

This dissertation presents a prototype of a gateway between the DomoBus technology and the EIB/KNX technology, serving as a starting point for a multi-technological solution. With this gateway the benefits of the DomoBus approach can be applied to the EIB/KNX technology. EIB/KNX was chosen because it is an international standard, it offers a high degree of functionality allowing the development of complex systems and has a large market share.

Keywords: Domotics, Home Automation, XML, User Interface, Monitoring and Control Applications, Technological Integration.

ÍNDICE

AGRADECIMENTOS.....	1
RESUMO.....	2
ABSTRACT	3
ÍNDICE	4
LISTA DE ABREVIATURAS	6
LISTA DE FIGURAS.....	7
LISTA DE TABELAS.....	9
1. INTRODUÇÃO	10
1.1. Enquadramento e Objectivos da Tese	10
1.2. Estrutura da Tese.....	12
2. TECNOLOGIAS DOMÓTICAS.....	13
2.1. Introdução às Tecnologias Domóticas	13
2.2. EIB (European Installation Bus) / KNX (Konnex).....	15
2.2.1. Introdução	15
2.2.2. Topologia da Rede EIB/KNX	16
2.2.3. Componentes EIB/KNX.....	18
2.2.4. Troca de informação e Interoperação.....	22
2.2.5. Protocolo EIB	29
2.2.6. Physical External Interface (PEI).....	32
2.3. DomoBus	37
2.3.1. Introdução	37
2.3.2. Visão Geral da Arquitectura	38
2.3.3. Modelo de Abstracção de Dispositivos.....	39
2.3.4. Serviço de Comunicações.....	42
2.3.5. Endereçamento	43
2.3.6. Mensagens entre Aplicações e rede DomoBus.....	44
2.4. Outras tecnologias	46
2.4.1. X10.....	46
2.4.2. LonWorks	47
2.4.3. CEBus.....	48

2.5.	Conclusão	49
3.	DESCRIÇÃO DA SOLUÇÃO.....	50
3.1.	Arquitectura da Solução.....	51
3.2.	Funcionamento da Solução.....	52
3.3.	Detalhes de Implementação.....	53
3.3.1.	Mapeamento de endereços.....	53
3.3.2.	Ficheiros de Configuração.....	54
3.3.3.	Conversão de Mensagens.....	56
3.3.4.	Comunicação com sistema DomoBus.....	58
3.3.5.	Comunicação EIB/KNX	60
4.	TESTES E RESULTADOS.....	63
4.1.	Rede de Dispositivos EIB/KNX	63
4.2.	Aplicação de Monitorização e Controlo DomoBus	66
4.3.	Procedimentos de teste	68
4.3.1.	Configuração do Adaptador EIB.....	68
4.3.2.	Configuração do ComServ	70
4.3.3.	Resultados	71
5.	CONCLUSÃO	77
6.	BIBLIOGRAFIA	79

LISTA DE ABREVIATURAS

- EIB – “*European Instalation Bus*”: Tecnologia domótica normalizada internacionalmente com grande implantação no mercado. Porá sua evolução é designada por KNX.
- KNX – “*Konnex*”: Evolução da tecnologia EIB.
- BCU – “*Bus Coupling Unit*”: Componente dos dispositivos EIB/KNX responsável pela ligação à rede.
- PEI – “*Physical External Interface*”: Interface responsável pela comunicação entre o BCU e o módulo de aplicação de um dispositivo EIB/KNX.
- EMI – “*External Message Interface*”: Conjunto de normas para a criação de mensagens utilizadas na interface PEI.
- EIS – “*EIB Interworking Standards*”: Normas utilizadas para a interoperação de dispositivos EIB/KNX.
- ETS – “*EIB Tool Software*”: Aplicação utilizada para a instalação e configuração de redes de dispositivos EIB/KNX.
- CM – “*Control Modules*”: Módulos de Controlo DomoBus que controlam o funcionamento dos dispositivos.
- SM – “*Supervision Modules*”: Módulos de Supervisão DomoBus responsáveis pela gestão e supervisão do sistema.

LISTA DE FIGURAS

Figura 1 – Rede EIB	15
Figura 2 – Topologia de uma rede EIB/KNX	17
Figura 3 – Instalação de uma rede EIB/KNX	17
Figura 4 – Topologia Lógica de uma rede EIB/KNX.....	18
Figura 5 – Dispositivo EIB/KNX (BCU e módulo de aplicação)	20
Figura 6 – Rail-Mounted Device	20
Figura 7 – Flush Mounted device.....	21
Figura 8 – Surface Mounted Device.....	21
Figura 9 – Pirâmide de Interoperação.....	23
Figura 10 – Transmissão orientada à ligação (endereçamento físico)	30
Figura 11 – Campos de endereço em endereçamento físico.....	30
Figura 12 – Endereçamento de grupo	31
Figura 13 – Formato de uma trama normal EIB/KNX.....	32
Figura 14 – Formato de uma mensagem do protocolo PEI Type 16.....	34
Figura 15 – Octeto relativo ao tamanho de uma mensagem.....	35
Figura 16 – Protocolo PEI Type 16	36
Figura 17 – Formato mensagem LM_Reset.ind.....	37
Figura 18 – Arquitectura do sistema DomoBus.....	39
Figura 19 – Modelo de um dispositivo domótico	40
Figura 20 – Encaminhamento de mensagens pelo ComServ	42
Figura 21 – Objecto de Comunicação numa aplicação DomoBus.	43
Figura 22 – Endereço DomoBus.....	44
Figura 23 – Formato de mensagens entre aplicações e rede DomoBus.....	45
Figura 24 – Arquitectura da solução de integração DomoBus – EIB/KNX	51
Figura 25 – Formato mensagem DomoBus	57
Figura 26 – Formato mensagem EMI	57
Figura 27 – Objecto comunicação DomoBus.....	59
Figura 28 – Objecto Comunicação EIB/KNX.....	61
Figura 29 – Protótipo de Dispositivos EIB/KNX	63
Figura 30 – DomoDevSimObserver: ecrã envio de mensagens	67

Figura 31 – DomoDevSimObserver: ecrã recepção de mensagens	68
Figura 32 – DomoDevSimObserver: mensagens de <i>report</i>	72
Figura 33 – DomoDevSimObserver: Ligar lâmpada 1.....	73
Figura 34 – DomoDevSimObserver: Respostas aos comandos enviados	74
Figura 35 – DomoDevSimObserver: Mensagem GET	75
Figura 36 – DomoDevSimObserver: Respostas mensagens GET.....	76

LISTA DE TABELAS

Tabela 1 – Endereços Físicos	64
Tabela 2 – Endereços de Grupo	65

1. INTRODUÇÃO

1.1. Enquadramento e Objectivos da Tese

Esta dissertação aborda o conceito de domótica, também referido como casas inteligentes ou edifícios inteligentes. A domótica é uma área que tem vindo a expandir-se ao longo dos últimos anos, embora a um ritmo inferior ao que seria de esperar. Este ritmo lento deveu-se essencialmente a alguma dispersão do mercado pelas diferentes tecnologias existentes, passando também por outros factores que serão detalhados no próximo capítulo desta dissertação.

A domótica oferece um elevado conjunto de benefícios aos seus utilizadores, que vão desde o conforto proveniente da automação de determinadas tarefas e actividades, passando pelo aumento da segurança dos edifícios através de sistemas de vigilância, actualmente largamente utilizados, até a uma melhor gestão e consequente poupança energética.

Actualmente existe no mercado um conjunto diversificado de tecnologias domóticas. Algumas dessas tecnologias são proprietárias, desenvolvidas por alguns fabricantes, com o intuito de criarem tecnologias de custo mais reduzido, conquistando assim o mercado. Por outro lado, existem tecnologias provenientes de movimentos de uniformização que constituem actualmente as tecnologias comerciais mais maduras, com maior implantação no mercado, estando normalizadas internacionalmente.

Existem ainda tecnologias em desenvolvimento, como é o caso do DomoBus, a tecnologia utilizada na solução proposta nesta tese. O DomoBus é proveniente de estudos académicos desenvolvidos ao longo dos últimos anos no Instituto Superior Técnico, pelo Prof. Renato Nunes, com o intuito de providenciar mecanismos de abstracção que suportem a integração e interoperação entre diferentes tecnologias, permitindo ainda uma elevada flexibilidade na configuração de sistemas domóticos, definição do seu comportamento, personalização e adequação às necessidades dos seus utilizadores.

As diferentes tecnologias comerciais apresentam diversas limitações, nomeadamente na capacidade de interoperação entre si, bem como na versatilidade das suas aplicações de monitorização e controlo, e interface dos sistemas com o utilizador. Esta tese foca-se essencialmente nestas limitações, apresentando uma possível solução para a sua minimização.

Como referido anteriormente, a evolução dos sistemas domóticos tem sido algo dispersa dada a existência de diferentes tecnologias, quer provenientes de fabricantes que se especializaram em produtos proprietários, quer por diferentes movimentos de uniformização independentes

que levaram à criação de diferentes normas. Como tal, estas tecnologias existentes são incompatíveis, não existindo actualmente qualquer solução que permita a interoperação entre as mesmas. Na prática, este problema obriga o utilizador à escolha de uma tecnologia para a criação de um sistema domótico, à qual ficará dependente. Para um utilizador desconhecedor destas tecnologias que pretenda equipar um edifício com uma rede de dispositivos domóticos, esta é uma escolha difícil e tipicamente irreversível, dado que para posterior expansão do sistema normalmente não será possível a utilização de uma tecnologia diferente.

Ao mesmo tempo, a monitorização e controlo de redes de dispositivos da generalidade das tecnologias existentes no mercado, é efectuada a partir de aplicações proprietárias que são desenhadas para a tecnologia em causa, destinadas a um determinado edifício e com um comportamento predefinido. Estas soluções não estão aptas para lidar com alterações ao sistema que controlam, não permitindo ao utilizador modificar o comportamento do sistema, adaptando-o a novas necessidades ou preferências. Na prática, um utilizador adquire uma aplicação de monitorização e controlo para uma determinada rede de dispositivos que acabou de instalar na sua habitação, e uma posterior modificação nesta rede ou uma alteração de comportamento irá implicar a necessidade de aquisição de uma nova versão desta aplicação. Desta forma, as soluções existentes não satisfazem geralmente os seus utilizadores a longo prazo.

A solução a ser descrita nesta dissertação tem por objectivo a resolução dos problemas apresentados anteriormente, recorrendo à tecnologia DomoBus e a uma aplicação de adaptação a diferentes sistemas disponíveis comercialmente. A escolha da abordagem DomoBus deveu-se à sua flexibilidade e facilidade de configuração através de ficheiros XML, sendo esta uma característica inexistente em qualquer tecnologia actualmente existente no mercado. Dada esta elevada configurabilidade, através de uma ferramenta adequada, qualquer utilizador poderá modificar o comportamento do seu sistema, de acordo com as suas necessidades e preferências.

O sistema comercial de dispositivos domóticos escolhido usa a tecnologia EIB/KNX que é actualmente dominante no mercado da domótica, sendo também a mais completa. Esta tecnologia surgiu com um dos referidos movimentos de uniformização, estando actualmente largamente implantada por todo o mundo e foi recentemente objecto de normalização internacional.

A solução proposta no presente trabalho permite solucionar a ausência de aplicações de monitorização e controlo flexíveis que satisfaçam os seus utilizadores, dado que as aplicações DomoBus permitem uma rápida e simples configuração para a interacção com qualquer tipo de dispositivos domóticos. Estas aplicações permitirão a qualquer utilizador de um sistema domótico EIB/KNX, configurá-lo de forma a ir de encontro às suas necessidades e preferências. Por outras palavras, esta solução permitirá aplicar as vantagens da tecnologia DomoBus à tecnologia EIB/KNX.

Uma vez que a tecnologia DomoBus permite a comunicação com diferentes redes de dispositivos, a utilização de gateways semelhantes ao que irá ser apresentado nesta dissertação, irá criar a possibilidade de integração de redes de diferentes tecnologias, utilizando as aplicações DomoBus como ponto de interligação das mesmas.

1.2. Estrutura da Tese

No capítulo dois desta dissertação é apresentado o estado da arte onde serão descritas duas das principais tecnologias existentes na área da domótica. Estas duas tecnologias servem de base para a implementação da solução posteriormente apresentada.

No capítulo três será descrita a solução proposta nesta dissertação. Será descrita a arquitectura usada e o seu funcionamento, sendo depois apresentados alguns detalhes de implementação.

Em seguida será apresentado um exemplo concreto para demonstração da solução proposta e serão ainda descritos alguns dos testes efectuados ao sistema e os respectivos resultados.

Por fim são apresentadas conclusões, com realce para os benefícios trazidos pela solução desenvolvida e sendo avaliadas as escolhas tecnológicas efectuadas. São também apresentadas sugestões de trabalho futuro para complementar a solução desenvolvida.

2. TECNOLOGIAS DOMÓTICAS

2.1. Introdução às Tecnologias Domóticas

A domótica tem vindo a expandir-se ao longo dos últimos anos devido aos benefícios e elevado grau de conforto que traz aos seus utilizadores. Apesar da domótica ainda não possuir uma cota de mercado elevada, podemos observar um aumento do número de empresas especializadas neste tipo de tecnologia, oferecendo uma elevada e diversa gama de produtos. De referir que esta expansão se deve em grande parte ao crescimento das novas tecnologias, nomeadamente o acesso doméstico à Internet, bem como a evolução da computação móvel [4].

Para além do elevado grau de conforto, outros benefícios da domótica são também o aumento da segurança e gestão energética. Como sistema direccionado para aumento da segurança, temos por exemplo, os sistemas de vigilância, actualmente bastante utilizados. Entre os sistemas direccionados para a poupança de energia, podemos considerar por exemplo, um sistema de iluminação que se desliga automaticamente quando não é detectada qualquer presença humana, ou que se reduz caso a intensidade luminosa proveniente do exterior, seja suficiente. A domótica pode ainda constituir um papel importante na utilização mais eficiente de fontes de energia renovável como a energia solar [4].

A evolução dos sistemas domóticos não tem decorrido à velocidade que seria de esperar. O período inicial foi dominado por produtos proprietários, não existindo normas para a criação de sistemas compostos por produtos de vários fabricantes. Após este período foram iniciados alguns movimentos de uniformização com o intuito de permitir compatibilidade entre produtos de diferentes fabricantes, promover reduções de custos, bem como permitir uma expansão mais rápida destes sistemas. No entanto, estes movimentos de uniformização independentes levaram à criação de diferentes normas, não tendo o efeito esperado na divulgação deste tipo de sistemas, levando a alguma confusão e dispersão do mercado. Mais recentemente, com o intuito de liderar o mercado, alguns fabricantes voltaram-se do novo para tecnologias proprietárias, utilizando produtos mais simples, com menos funcionalidades, e consequentemente, mais baratos [4].

Esta existência de várias tecnologias incompatíveis poderá constituir um grande factor na dificuldade de divulgação dos sistemas domóticos. Por outro lado, pode também considerar-se uma limitação a ausência de normas para a abordagem ao problema da especificação de um dado sistema e a sua adaptação a uma habitação em concreto e aos requisitos específicos dos seus utilizadores. Nalguns casos, como sucede com o sistema EIB/KNX, abordado nesta tese, existem disponíveis aplicações (e.g., ETS3) mas estas são específicas para uma dada

tecnologia e destinam-se a ser usadas por pessoal técnico especializado e não pelo utilizador comum [3].

Para além das referidas limitações, a inexistência de aplicações genéricas de monitorização e controlo aptas para lidar com alterações e que permitam ao utilizador modificar o comportamento do sistema adaptando-o a novas necessidades ou preferências, constitui talvez a maior barreira à expansão dos sistemas domóticos. As soluções existentes são aplicações específicas para utilização com uma determinada tecnologia, num determinado edifício e com um comportamento predefinido. A aquisição de uma solução deste tipo, geralmente não satisfaz os seus utilizadores a longo prazo, pois a necessidade de modificação do sistema domótico, implica a necessidade de aquisição de uma nova versão da aplicação [3].

Dentro das principais tecnologias domóticas existentes no mercado podemos referir o CEBus, o LonWorks, o X10 e o KNX. Este último surgiu como um dos anteriormente referidos movimentos de uniformização, denominado inicialmente por EIB (European Installation Bus). Mais recentemente, outras empresas optaram por se unir ao EIB, tendo então mudado o seu nome para KNX (Konnex), no entanto este sistema ainda é vulgarmente conhecido pelo seu anterior nome.

Nesta dissertação optou-se por investir essencialmente em duas tecnologias: EIB/KNX e DomoBus, sendo estas as que se consideraram mais marcantes na área da domótica por diferentes motivos. Actualmente o EIB/KNX é uma tecnologia comercial madura, bastante divulgada e com grande implantação no mercado em virtude de ser uma solução normalizada internacionalmente, bastante completa, oferecendo um elevado nível funcional. Pelos mesmos motivos, esta foi a tecnologia escolhida para o desenvolvimento do protótipo apresentado nesta dissertação.

A escolha do DomoBus deveu-se ao modelo de abstracção que este propõe tornando-o independente da tecnologia dando assim suporte à interoperação com diferentes sistemas. Esta tecnologia serve de base para desenvolvimento da solução apresentada nesta dissertação. Esta tecnologia é fruto de um trabalho de investigação académico desenvolvido no Instituto Superior Técnico, desenhada com o intuito de ser o mais simples possível de forma a minimizar o seu custo, mas ao mesmo tempo, bastante flexível e poderosa.

Desta forma, neste capítulo será descrita a tecnologia EIB/KNX, para o qual irá ser detalhada a arquitectura e topologia, referindo também os principais componentes que a compõem. No sentido de mostrar como os dispositivos interoperam entre si, irão ser detalhados os seus modos de endereçamento e normas de comunicação. Com o intuito de perceber como é que uma aplicação de monitorização, instalada num computador ligado à rede, controla um sistema EIB, será detalhado o protocolo que permite esta comunicação.

Numa segunda parte, para que possam ser identificadas as diferenças entre um sistema DomoBus e os restantes sistemas, irá ser dada uma visão geral sobre a arquitectura deste. Seguidamente, de forma a perceber como uma aplicação DomoBus poderá controlar dispositivos de diferentes tecnologias, este documento irá explicar o modelo de abstracção de

dispositivos utilizado por este sistema. Por fim, será detalhado todo o sistema de comunicação utilizado pelas aplicações DomoBus, sendo este uma peça fundamental na solução que irá ser apresentada nesta dissertação pois é este sistema que permite a interligação entre múltiplas redes de dispositivos.

2.2. EIB (European Installation Bus) / KNX (Konnex)

2.2.1. Introdução

O sistema EIB/KNX foi desenhado de forma a poder ser instalado tanto em edifícios de grandes dimensões, como edifícios de escritórios, escolas, hospitais e fábricas, bem como em residências ou edifícios de habitação (condomínios). A sua finalidade é monitorizar e controlar sistemas de iluminação, aquecimento, ventilação, ar condicionado, balanceamento de carga ou sistemas de vigilância.

O sistema EIB/KNX permite aos seus dispositivos retirarem a energia necessária para o seu funcionamento, directamente das redes de comunicação. Outros dispositivos podem, adicionalmente, necessitar de fornecimento de energia por outros meios, como os dispositivos que comunicam com o sistema por rádio frequência ou infravermelhos [1].

A figura seguinte mostra alguns exemplos de dispositivos comuns num sistema EIB/KNX [1].

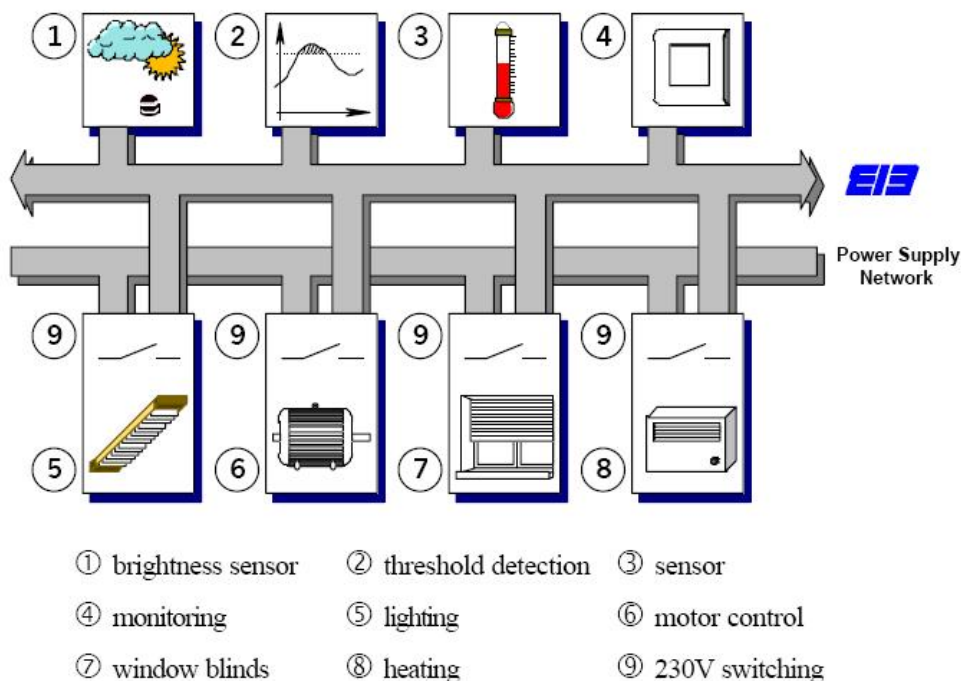


Figura 1 – Rede EIB

Como já foi referido, os sistemas EIB/KNX são desenhados para permitir monitorizar e controlar dispositivos num edifício. Desta forma permite transferência de informação entre dispositivos conectados à rede. Esta solução foi criada também com o intuito de suportar estas aplicações de uma forma flexível e com custo reduzido, no entanto, não podemos afirmar que esta será actualmente a solução domótica mais flexível e barata.

O sistema EIB/KNX é usualmente implementado como um sistema descentralizado, no entanto, é possível, sempre que necessário, implementar aplicações centralizadas para gestão do sistema. As soluções descentralizadas são implementadas com base unicamente em dispositivos que podem ser emissores ou receptores, que comunicam directamente entre si, sem necessidade de hierarquia ou supervisão da rede. Por outro lado, pode ser ligado à rede, uma aplicação de controlo, para gestão do sistema. Esta aplicação corre usualmente num computador que pode ser ligado à rede a partir de qualquer localização, permitindo assim uma gestão centralizada do sistema [1].

2.2.2. Topologia da Rede EIB/KNX

O sistema EIB/KNX suporta diferentes meios físicos de comunicação, como “Twisted Pair” (par entrançado), que é o tipo de rede mais comum, mas também rede eléctrica, rádio frequência e infravermelhos. É ainda possível instalar gateways para outras redes como Ethernet ou WLAN.

A topologia da rede pode ser entendida como a descrição dos caminhos através dos quais os sinais de comunicação podem ser transportados. Em alguns casos, a comunicação não é efectuada com base em sinais eléctricos, como é o caso das redes de rádio frequência e infravermelhos, contrariamente às habituais redes de “par entrançado” (*Twisted Pair*), rede eléctrica ou fibra óptica.

As redes de “par entrançado” são o meio de transmissão de dados mais comum num sistema EIB/KNX. Os segmentos da rede podem possuir uma topologia arbitrária (linear, em estrela, em árvore, em anel, ou combinações destas, como é visível na figura 2) constituída por secções individuais, desde que os requisitos eléctricos (comprimento dos segmentos) não sejam excedidos [1].

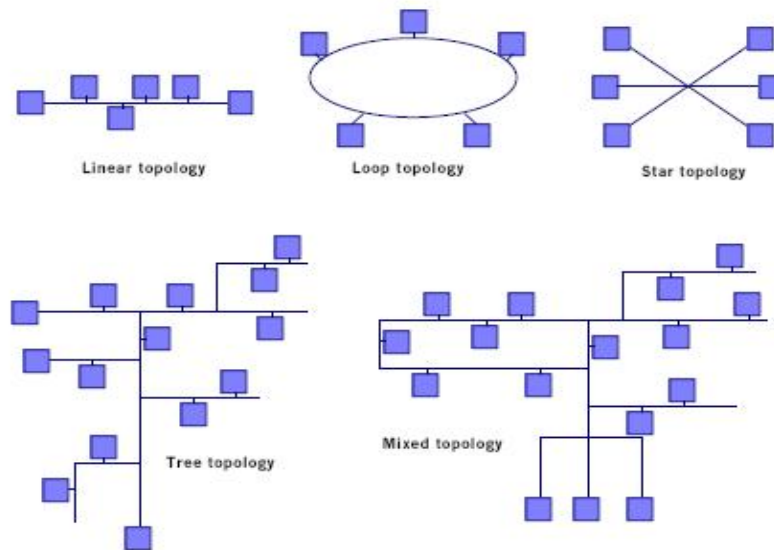


Figura 2 – Topologia de uma rede EIB/KNX

A rede EIB/KNX necessita de pelo menos um par para funcionar. No entanto podem ser usados cabos com dois pares, nos quais um dos pares será usado para transmissão de informação, e o segundo para fornecer energia complementar aos dispositivos. A instalação de uma rede EIB/KNX é semelhante à instalação de uma rede eléctrica, sendo assim utilizada tipicamente a topologia em árvore (ver figura 3).

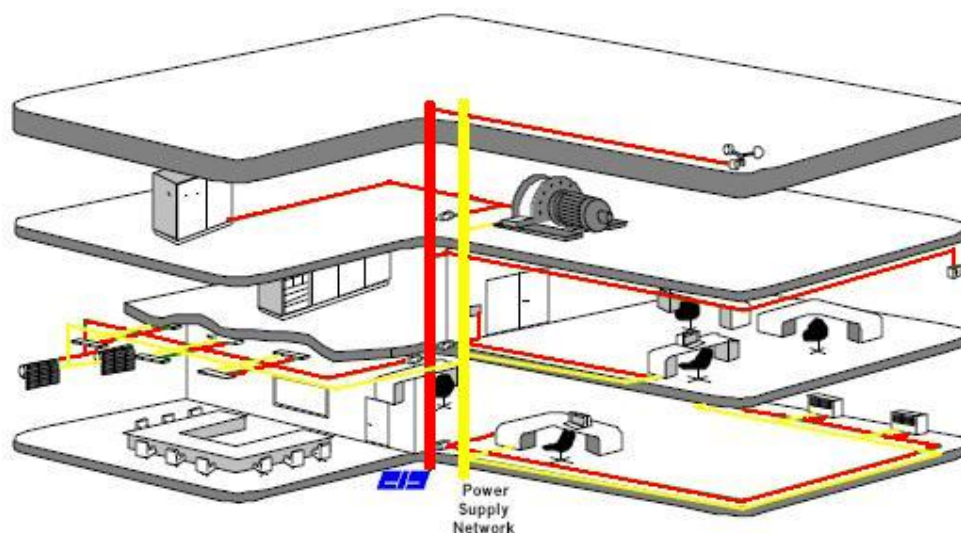


Figura 3 – Instalação de uma rede EIB/KNX

A tecnologia EIB/KNX pode acomodar um máximo teórico de 65.536 dispositivos correspondendo a um espaço de endereçamento de 16 bit. A topologia de uma rede permite interligar 256 dispositivos num segmento, podendo até 15 segmentos ser ligados a uma linha principal formando uma área. Uma linha de *backbone* pode ser usada para interligar até 15 áreas.

A figura seguinte representa a topologia de uma rede EIB/KNX. Podemos observar uma linha de *backbone* à qual são ligadas várias áreas através de *Area Couplers*. À linha principal dentro de cada área são ligados vários segmentos através de *Line Couplers*. Podemos também observar na imagem o espaço de endereçamento das várias áreas, segmentos e dispositivos.

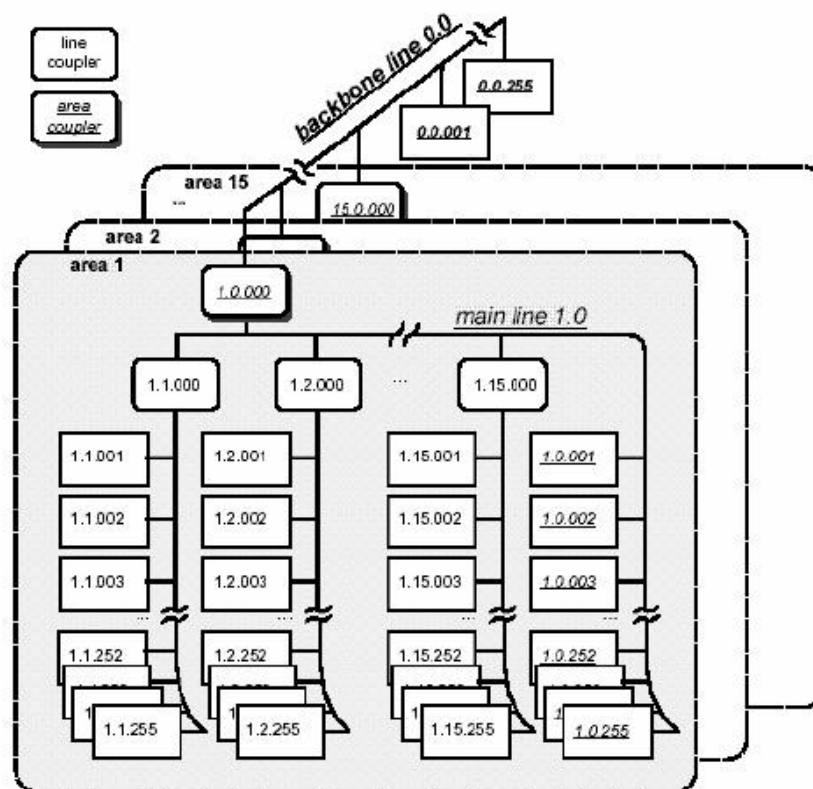


Figura 4 – Topologia Lógica de uma rede EIB/KNX

2.2.3. Componentes EIB/KNX

Os componentes EIB/KNX estão divididos em três tipos, de acordo com as suas funcionalidades:

- Componentes básicos, como fontes de alimentação, filtros de sinal, etc.

- Componentes de sistema, que suportam as operações básicas do sistema como os Bus Coupling Units (BCU), Line Couplers (LC), Área Couplers, Repeaters, etc.
- Dispositivos EIB/KNX que são dedicados a aplicações como sensores, actuadores, decodificadores de sinal, painéis indicativos, etc. Este dispositivos são conectados à rede EIB/KNX através de BCU's ou interfaces semelhantes.

Os BCU podem ser dispositivos independentes através dos quais os módulos de aplicação se conectam à rede ou, em alternativa, podem estar integrados nestes dispositivos. Um BCU é composto por um módulo de transmissão e um módulo controlador (Communication Controller). O módulo de transmissão é responsável pelo envio e recepção de dados da rede. O módulo controlador possui um microprocessador que suporta uma interface externa (PEI), um sistema operativo e pode eventualmente executar um programa de controlo do dispositivo a que está associado [1].

O “repeater” é um repetidor de sinal para as redes par entrançado. A sua função é regenerar os sinais eléctricos permitindo unir segmentos de rede criando redes extensas, necessárias para grandes edifícios.

O “Line Coupler” é um componente de sistema para as redes “Twisted Pair”. Este componente partilha as mesmas funções básicas com o “Repeater”, mas conecta um segmento de rede, à linha principal da área a que pertence. O line coupler garante a filtragem e o encaminhamento dos pacotes de dados, da rede principal para o seu segmento e vice-versa, o que implica uma melhor gestão do tráfego global da rede. Por outras palavras, este componente une ramificações da rede EIB/KNX à rede principal [1].

O “Area Coupler” tem um funcionamento semelhante ao “Line Coupler”, no entanto serve para ligar uma área à linha *Backbone* da rede a que pertence.

2.2.3.1. Estrutura dos dispositivos EIB

Os dispositivos EIB/KNX são geralmente constituídos por duas partes: BCU e módulo de aplicação (ver fig. 5). Os BCU, como referido anteriormente, permitem a ligação do dispositivo à rede. Esta divisão de módulos permite separar a comunicação com a rede, do hardware da aplicação. Existe ainda a possibilidade de ter o BCU e o módulo de aplicação em dispositivos separados.

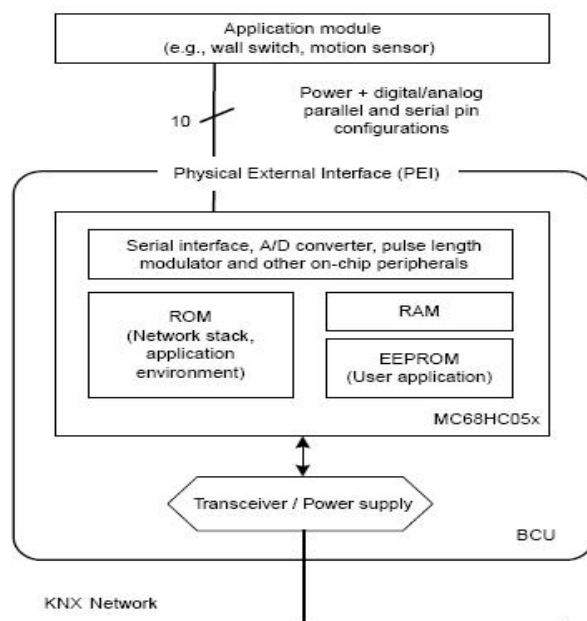


Figura 5 – Dispositivo EIB/KNX (BCU e módulo de aplicação)

Do ponto de vista da instalação, os dispositivos EIB/KNX podem ser divididos em quatro grupos [1]:

1. *Rail-mounted*: dispositivos EIB/KNX para controlo de aplicações montados numa calha, facilitando a sua arrumação.



Figura 6 – Rail-Mounted Device

2. *Flush mounted*: dispositivos para os quais o BCU é montado dentro da parede e o módulo da aplicação anexado a este, montado à superfície da parede, como interruptores, decodificadores de sinal, painéis indicativos, etc.



Figura 7 – Flush Mounted device.

3. *Surface mounted*: dispositivos nos quais tanto o BCU como o módulo de aplicação estão montados à superfície da parede.



Figura 8 – Surface Mounted Device

4. *Device mounted*: dispositivos incorporados nos equipamentos de um edifício, como aquecedores, lâmpadas, ar condicionado, etc.

2.2.3.2. Gateways para redes externas

Um sistema EIB/KNX pode ser conectado, via gateways, a redes externas. Esta conexão pode ser efectuada a partir de qualquer ponto da rede, tanto da rede principal como qualquer segmento.

Estes gateways conectam a rede EIB/KNX a outras redes, tipicamente redes Intranet ou Extranet, onde as camadas de comunicação são diferentes, especialmente nos casos em que é necessário efectuar tradução de endereços de uma rede para outra.

Na prática, estes gateways permitem o acesso remoto a redes EIB/KNX através de aplicações de monitorização e controlo, via Internet, bem como a interoperação entre diferentes redes EIB/KNX, fisicamente distantes, mas ligadas entre si através de uma rede local ou Internet.

Embora estes gateways permitam a comunicação remota com redes EIB/KNX através de comunicações TCP-IP, não existe qualquer tipo de modelação de dispositivos e meios de comunicação tecnologicamente independentes que suportem a interoperação entre redes de dispositivos de outras tecnologias. É esta limitação da tecnologia EIB/KNX e das restantes

tecnologias domóticas existentes no mercado, que a solução apresentada nesta dissertação, pretende resolver.

2.2.4. Troca de informação e Interoperação

2.2.4.1 Troca de informação

A comunicação entre um sensor (por exemplo, interruptor) e um actuador (por exemplo, lâmpada) é composta por uma sequência de operações. Utilizando o protocolo EIB/KNX, um interruptor que possui um endereço físico único, pode comunicar com várias lâmpadas através de um endereço de grupo.

O endereçamento em grupo é baseado na troca de mensagens entre um emissor e um conjunto de receptores que partilham o mesmo endereço de grupo. Um emissor apenas pode enviar uma mensagem para um endereço de grupo, ou seja, não pode enviar uma mensagem destinada a vários endereços de grupo simultaneamente. Por outro lado um receptor pode subscrever diversos endereços de grupo, permitindo-o receber mensagens de vários emissores. Desta forma, voltando ao exemplo anterior, todos os dispositivos que subscrevem o endereço de grupo da lâmpada, vão receber o comando enviado pelo interruptor.

Os modos de endereçamento serão detalhados no capítulo 2.2.5.1 desta dissertação.

2.2.4.2. Interoperação

O objectivo da comunicação num sistema EIB/KNB consiste na interoperação entre sensores e actuadores. Esta interoperação é sem dúvida a base do funcionamento de um sistema EIB/KNX, bem como de qualquer outro sistema domótico. No entanto esta interoperação apenas é suportada entre dispositivos EIB/KNX, não existindo qualquer sistema que possibilite a interoperação com dispositivos de outras tecnologias.

A figura seguinte mostra os diferentes níveis de interoperação existentes neste sistema, começando no formato dos dados, até funcionalidade das aplicações.

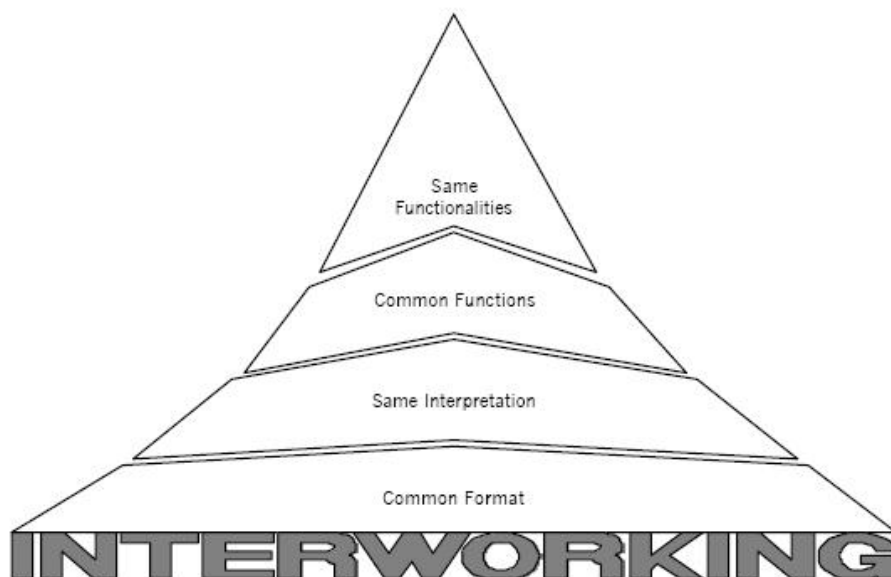


Figura 9 – Pirâmide de Interoperação

Analisando a pirâmide, na base da interoperação está a troca de dados com o mesmo formato. Todos os intervenientes têm que utilizar o mesmo formato de mensagens.

O nível seguinte consiste em garantir que a informação trocada é entendida por todos os intervenientes da mesma forma. Se um sensor envia, por exemplo, o valor da temperatura de uma sala, o actuador (por exemplo, um aquecedor) tem que conhecer a que se refere o valor.

O terceiro nível consiste em garantir que os intervenientes conhecem as funções que ambos executam. Se um interruptor envia um comando para desligar um actuador, logicamente esse actuador tem que conhecer a função desligar.

No topo da pirâmide está a partilha das mesmas funcionalidades de forma permitir o mesmo comportamento em diferentes produtos. Se um termóstato envia uma mensagem a um sistema de aquecimento, para aumentar a temperatura para um determinado valor, o sistema de aquecimento tem que suportar essa funcionalidade

Para garantir estes níveis de interoperação, foram definidas normas: EIS (EIB Interworking Standards), através das quais devem ser criadas e programadas as aplicações [1]. Estas normas serão referidas na próxima secção desta dissertação.

Para que os níveis de interoperação anteriores sejam atingidos, são necessários alguns pré-requisitos. Para que os dispositivos consigam comunicar correctamente têm que ser configurados entre si. Os modos de configuração de dispositivos existentes serão referidos posteriormente nesta dissertação.

2.2.4.3. EIB Interworking Standards (EIS)

Como foi referido anteriormente, os dispositivos EIB/KNX têm que conseguir comunicar entre si, mesmo que sejam de diferentes fabricantes. Desta forma, para além da utilização do mesmo protocolo, necessitam de conseguir interpretar correctamente as mensagens enviadas por outros dispositivos.

As normas EIS definem regras para a interoperação de funções utilizadas pelos dispositivos. O seu objectivo consiste em definir os valores e interpretação dos dados incluídos nos objectos de comunicação utilizados nas funções. Estas funções estandardizadas são chamadas de *EIB-functions*.

EIB-Functions

O nome das *EIB-functions* está relacionado com a primeira aplicação a que foram dirigidas, mas podem ser usadas para outras aplicações. Dando um exemplo, a função “Dimming” foi inicialmente dirigida para variar a intensidade de sistemas de iluminação, no entanto também pode ser utilizada em sistemas de aquecimento. Ao mesmo tempo, uma função pode ser composta por várias sub-funções, dirigidas às mesmas aplicações mas com funcionalidades ligeiramente diferentes.

De seguida serão apresentadas as diferentes funções disponibilizadas pelas normas EIS.

EIS 1: Switching – Função que utiliza 1 bit, usualmente utilizada para activar ou desactivar uma aplicação.

EIS 2: Dimming – Esta função suporta três sub-funções: “position” que permite activar ou desactivar a aplicação através de 1 bit (na realidade não é mais do que a função EIS 1); “control” que permite aumentar ou diminuir a intensidade, através de uma variável de 4 bits (*dimming* relativo); “value” que permite a atribuição directa de um determinado valor (*dimming* absoluto) através de uma variável de 8 bits idêntica à utilizada na função EIS 6.

EIS 3: Time – Função que permite enviar o tempo (“time of day”: dia da semana, horas, minutos, segundos) actual a todos os dispositivos que processam esta informação. O tempo é representado por uma variável de 3 bytes.

EIS 4: Date – Função que permite enviar a data (ano, mês, dia) actual a todos os dispositivos que processam esta informação. A data é representada por uma variável de 3 bytes.

EIS 5: Value – Função que permite o envio de valores físicos entre dispositivos, numa variável de 2 bytes, com o seguinte formato binário: um bit (S) referente ao sinal da mantissa, quatro bits (EEEE) referentes ao valor do expoente em base dois (de 0 a 15), e onze bits (M...M) referentes ao valor da mantissa em complemento para dois (de -2048 a 2047). O cálculo do valor é efectuado através da seguinte fórmula: $[(-1)^S * (0.01M) * 2^E]$ podendo então variar entre -671.088.64 a +670 760.96.

EIS 6: Scaling – Função que permite a transmissão de valores relativos com uma resolução de 8 bits. Geralmente utilizada para valores percentuais.

EIS 7: Drive Control – Função composta por duas sub-funções: “move” que permite colocar um dispositivo em movimento ou mudar a sua direcção; “step” que permite para um dispositivo ou indicar movimento gradual.

EIS 8: Priority – Função que permite forçar o estado de um actuador. Se esta propriedade for activada é possível forçar o estado do actuador independentemente do funcionamento previamente programado.

EIS 9: Float Value – Função que permite transmissão de dados em vírgula flutuante usando 4 bytes.

EIS 10: 16-bit Counter Value – Função que permite transmitir valores que representam contadores de 16 bits.

EIS 11: 32-bit Counter Value – Função idêntica à anterior mas para contadores de 32 bits.

EIS 12: Access – Controla permissões de acesso a aplicações através de uma variável de 4 bytes.

EIS 13: EIB-ASCII-Char – Permite o envio de um único carácter.

EIS 14: 8-bit Counter – Função idêntica à função 10 mas usando 8 bits

EIS 15: Character String – Permite o envio de mensagens textuais com tamanho até 14 bytes.

2.2.4.4. Modos de Configuração

De forma a satisfazer as diferentes necessidades de instalação dos vários dispositivos, o sistema EIB/KNX possui um conjunto de funcionalidades para permitir a escolha entre diversos modos de configuração de dispositivos.

Estes diferentes modos de configuração permitem aos fabricantes, alguma liberdade na construção dos seus produtos, garantindo ao mesmo tempo a consistência necessária à interoperação entre dispositivos.

Através do uso da ferramenta ETS (EIB Tool Software), ou outra ferramenta de gestão compatível com o sistema EIB/KNX, a interoperação entre dispositivos com diferentes modos de configuração, é possível. Estas ferramentas acedem ao perfil de cada dispositivo para conhecer o modo de configuração dos mesmos, utilizando depois procedimentos de gestão de acordo com a informação recolhida, para configurar a interoperação entre dispositivos [5].

De seguida serão descritos resumidamente os vários modos de configuração suportados pelo sistema EIB/KNX.

1. “System Mode”

Os dispositivos que implementam “System Mode” oferecem o processo de configuração mais versátil, transferindo a responsabilidade de interligação com outros dispositivos para um *configuration master*. Este papel é geralmente executado por um conjunto de ferramentas de gestão (tipicamente a aplicação ETS), executadas a partir de um computador conectado à rede EIB/KNX.

Para aceder a informação mais detalhada sobre os dispositivos, o ETS utiliza uma base de dados onde estão armazenadas todas as funcionalidades que estes suportam. Esta informação é geralmente criada e mantida pelos fabricantes dos dispositivos, e importada para o ETS pelo utilizador.

A ferramenta ETS suporta a configuração das seguintes opções:

- Binding*: Estabelecer os endereços de grupo de forma a permitir comunicação simultânea entre vários dispositivos.

- Parameterisation*: Estabelecer os parâmetros de configuração dos dispositivos de acordo com a documentação dos fabricantes.

- *Downloading*: Carregar programas para dispositivos que suportam diversas funcionalidades. Como já foi referido neste texto, os dispositivos são geralmente compostos por um BCU e um módulo de aplicação. Os BCU suportam o carregamento de aplicações de controlo do dispositivo.

A responsabilidade de configuração destas funcionalidades, para os vários dispositivos, cabe naturalmente ao instalador do sistema [5].

Actualmente, a grande maioria das instalações de redes de dispositivos EIB/KNX é configurada usando este modo.

2. “Controller Mode”

O *Controller Mode* é definido para suportar a instalação de um número limitado de dispositivos num segmento da rede. Uma instalação utilizando este modo irá consistir num dispositivo especial apelidado de controlador que está encarregue de suportar o processo de configuração. O controlador suporta uma ou mais aplicações (por exemplo, iluminação). Salvo algumas excepções, não é necessário, mas recomendado que o controlador permaneça conectado à rede após o processo de configuração.

Os dispositivos que implementam este modo de configuração apresentam uma descrição limitada das suas funcionalidades, pois como já foi referido, para armazenar informação mais detalhada sobre dispositivos é necessário uma ferramenta externa como o ETS. Estes dispositivos têm a possibilidade de serem dinamicamente configurados, estabelecendo o necessário endereço físico, endereços de grupo e parâmetros de funcionamento.

Durante o processo de configuração, o papel do controlador é criar associações entre um grupo de objectos, para uma dada funcionalidade. O controlador não necessita de conhecer, à partida, nenhuma das funcionalidades suportadas pelos dispositivos, estas serão lidas, no momento da instalação, a partir do perfil de cada dispositivo.

Segundo as instruções do instalador, o controlador atribui endereços individuais a cada dispositivo, estabelece as ligações entre os dispositivos, e estabelece os parâmetros necessários para cada dispositivo [5].

3. “Push-Button Mode”

O modo *Push-button*, tal como o modo *controller*, é direccionado para suportar instalação de um número limitado de dispositivos num segmento de rede. Não existe, no entanto, necessidade de nenhum dispositivo especializado na configuração. Cada dispositivo que implementa este modo de configuração, necessita de meios próprios para configurar a aplicação que vai executar.

Cada dispositivo possui a capacidade de ser dinamicamente configurado, especificando o seu endereço físico, endereço de grupo e parâmetros necessário ao seu funcionamento. A troca de parâmetros com outros dispositivos também é possível, no entanto a configuração é essencialmente local.

Aquando da configuração, o instalador designa sucessivamente os dispositivos cujas funções vão ser interligadas, de acordo com as especificações de cada fabricante. A troca de dados de configuração entre os dispositivos, tipicamente sensores e actuadores, ocorre através de um

serviço da camada de aplicação. Os dispositivos emissores, autonomamente adquirem o seu endereço de grupo único, enviando-o posteriormente aos dispositivos receptores [5].

4. “Logical Tag Reflex”

No modo *Logical Tag Reflex* não é necessária nenhuma ferramenta de configuração. Os dispositivos têm a possibilidade de atribuir a si próprios e às suas funcionalidades, uma etiqueta contendo um determinado valor a ser tido em conta durante a configuração.

Funções correspondentes em diferentes dispositivos, às quais foram atribuídas etiquetas com os mesmos valores são automaticamente interligadas e irão interoperar. É ainda possível a atribuição de vários valores a cada funcionalidade, permitindo a interoperação independente com diferentes grupos de dispositivos ou funcionalidades.

O funcionamento destes dispositivos é baseado em endereços de grupo, sendo que, do ponto de vista da gestão da rede, estes dispositivos aparecem como um conjunto, por não possuírem endereços individuais [5].

5. “Logical Tag Supervised”

No modo *Logical Tag Supervised* não é necessária nenhuma ferramenta de configuração, no entanto é necessário que cada aplicação possua um supervisor. Um supervisor pode supervisionar mais do que uma aplicação. Tal como no modo anterior, os dispositivos e as suas funcionalidades têm a capacidade de adquirir etiquetas com valores que serão usados durante o processo de configuração.

O supervisor acede ao perfil de cada dispositivo, e através de mecanismos normalizados, estabelece para cada um os endereços de grupo que necessitam para o seu correcto funcionamento. Como no modo anterior, funcionalidades de dispositivos diferentes que possuam etiquetas com os mesmos valores, são interligadas e irão interoperar. Tal como no modo anterior, o funcionamento destes dispositivos baseia-se em endereçamento de grupo [5].

6. “A-Mode”

Este modo, contrariamente aos outros é destinado a instalações muito simples, podendo ser efectuadas por um utilizador sem conhecimentos extensos sobre o sistema EIB/KNX. Desta forma, os dispositivos incluem mecanismos que autonomamente permitem localizar outros dispositivos com os quais irão interoperar. Estes têm também a capacidade de adquirir os seus próprios endereços físicos.

Os dispositivos que utilizam este modo de configuração estão geralmente direccionados a uma única aplicação que contém o seu próprio *Application Controller* (dispositivo encarregue de controlar uma dada aplicação), sendo este também o *Configuration Master* para esta aplicação. A configuração dos endereços de grupo está também a cargo do *Configuration Master* [5].

2.2.5. Protocolo EIB

2.2.5.1. Modos de endereçamento

Os dispositivos numa rede EIB/KNX podem ser endereçados de duas formas: endereçamento físico (operações de sistema) e endereçamento de grupo (operações normais).

Todos os dispositivos da rede são identificados por um endereço físico único. Este endereço de um dispositivo consiste num número que identifica a área da rede onde este está ligado, na linha (segmento da rede), e no número do dispositivo. Num pacote enviado para a rede, o campo de endereço de origem contém sempre o endereço físico do emissor. No endereço de destino geralmente só é usado o endereço físico para inicialização, programação e operações de diagnóstico (transmissão orientada à ligação). Daí, este tipo de endereços serem considerados endereços de sistema.

A figura seguinte ilustra uma transmissão orientada à ligação (através de endereços físicos) [1].

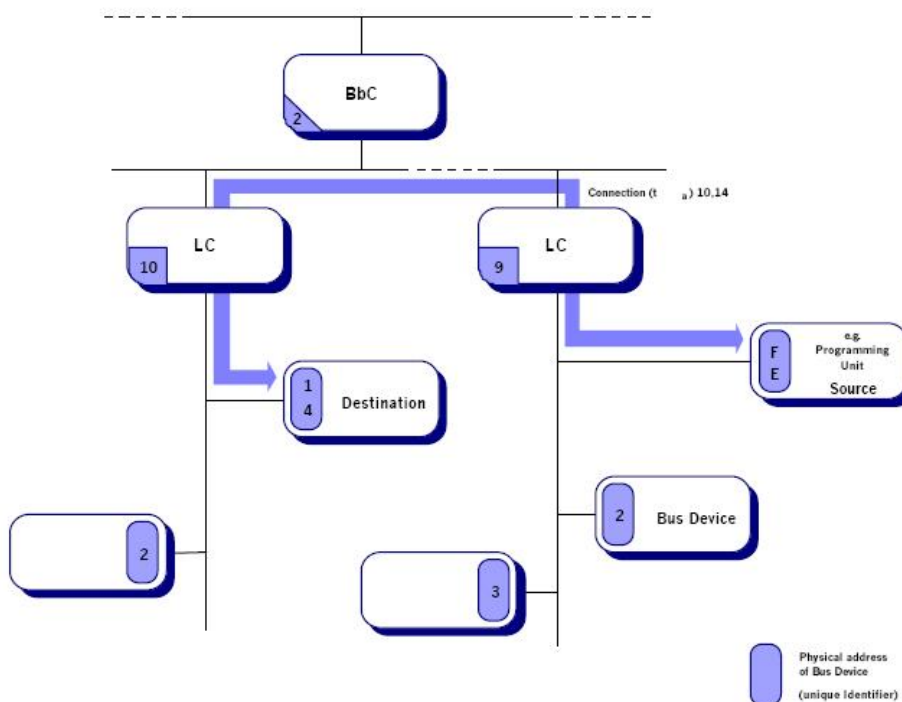


Figura 10 – Transmissão orientada à ligação (endereçamento físico)

Como é visível na figura seguinte, um pacote relativo a esta ligação, possui como endereço de origem “2.9.FE” em que o valor “2” corresponde à área, o valor “9” corresponde à linha (identificada no *Line Coupler*), e o valor “FE” relativo ao número do dispositivo. Por sua vez, o endereço de destino é “2.10.14”, onde, novamente, o valor “2” corresponde à área, o valor “10” à linha, e o valor “14” ao número do dispositivo de destino.

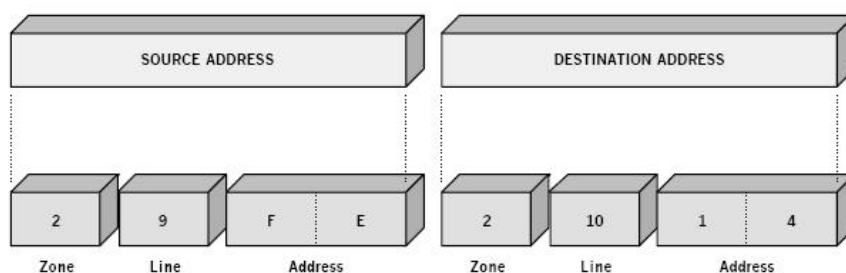


Figura 11 – Campos de endereço em endereçamento físico

O endereçamento de grupo corresponde ao modo de operação normal. Funcionalidades de dispositivos EIB/KNX contendo o mesmo valor de endereço de grupo, podem ser controlados em simultâneo através de uma única mensagem com destino a este mesmo endereço. Por outro lado, várias funcionalidades de um dispositivo podem pertencer a vários grupos diferentes, e serem activadas por um elemento de um desses grupos, sem interferir com os dispositivos dos outros grupos a que pertence.

No caso ilustrado na figura 12 podemos observar a vermelho, os endereços de grupo (1, 2 e 100) subscritos pelos dispositivos. O dispositivo “2.9.91” envia uma mensagem para o endereço de grupo “100”. Esta mensagem irá influenciar todas as aplicações que subscrevem este endereço de grupo, neste caso, as aplicações “out1” e “out2” do dispositivo “2.10.12” e as aplicações “out1” e “out2” do dispositivo “2.10.13”.

Caso a mensagem transmitida tivesse como endereço de destino “1”, esta iria influenciar as aplicações “out1” dos dispositivos “2.10.12” e “2.10.13”, não influenciando nenhuma das outras aplicações.

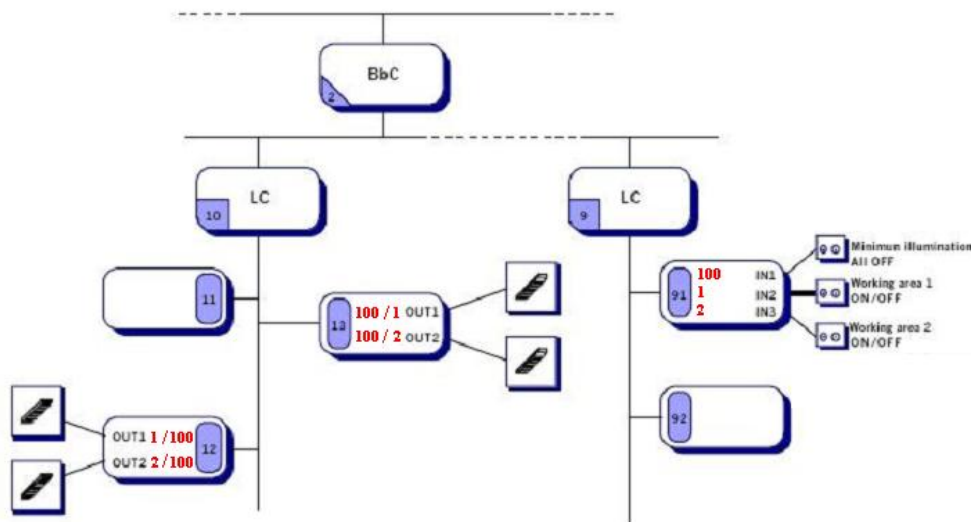


Figura 12 – Endereçamento de grupo

O endereçamento de grupo é uma ligação lógica entre dispositivos. Um sensor pode apenas transmitir para um dado endereço de grupo, enquanto um actuador pode receber a partir de vários endereços de grupo. O endereçamento de grupo aumenta a flexibilidade na instalação de novos dispositivos, na medida em que apenas é necessário subscrever os correctos endereços de grupo.

2.2.5.2. Tramas EIB/KNX

A figura seguinte ilustra a composição de uma mensagem EIB/KNX, que é enviada para a rede a partir de um BCU.

octet 0	1	2	3	4	5	6	7	8	..	N - 1	N ≤ 22
Control Field	Source Address		Destination Address		Address Type; NPCI; length	TP CI	AP CI	data /AP CI	data		Frame Check

Figura 13 – Formato de uma trama normal EIB/KNX.

A informação contida no campo *Control Field* permite determinar a prioridade de uma trama e distingue uma trama normal, de uma trama estendida. As tramas estendidas são fruto de uma extensão ao protocolo EIB/KNX (actualmente suportadas por poucos dispositivos), sendo semelhantes à ilustrada na figura, permitem um tamanho máximo de 254 octetos, contrariamente a uma trama normal que apenas permite 22.

Os campos *Source Address* e *Destination Address* contêm os endereços de origem e destino da trama. Como foi referido no capítulo anterior desta dissertação, o endereço de origem é sempre um endereço físico, no entanto, o endereço de destino pode ser um endereço físico ou um endereço de grupo. O tipo de endereço de destino utilizado em cada trama é referido no campo *Address Type* da mesma. O campo *NPCI Length* possui o número máximo de saltos (*hops*) da trama, sendo decrementado em cada router até chegar a 0, sendo então a trama descartada.

O campo *TPCI* (*Transport Layer Protocol Control Information*) controla as relações de comunicação do ponto de vista da camada de transporte, como por exemplo, para estabelecer e manter uma ligação ponto-a-ponto. Por sua vez, o campo *APCI* (*Application Layer Protocol Control Information*) controla os serviços da camada de aplicação (leitura, escrita, resposta...) que estão disponíveis entre os intervenientes da comunicação.

Os octetos seguintes são relativos aos dados a enviar, dependendo obviamente da função pretendida, podendo ocupar até 14 octetos numa trama normal. Para os casos em que é necessário transferir dados superiores a 14 octetos (caso o utilizador pretenda carregar uma aplicação para um BCU), é da responsabilidade da ferramenta de gestão separar os dados em várias tramas.

Por fim, o campo *Frame Check* permite confirmar a consistência e fiabilidade da trama.

2.2.6. Physical External Interface (PEI)

A *Physical External Interface* (PEI) é uma interface situada num dispositivo EIB/KNX entre o *Bus Coupling Unit* (BCU) e o módulo de aplicação. Como já referido anteriormente, o BCU contém todas as camadas relativas ao protocolo de comunicação EIB/KNX e, opcionalmente, a aplicação interna do utilizador. Este protocolo, como será descrito mais à frente, é de grande

importância para o desenvolvimento do trabalho em causa, pois permite a comunicação entre um computador e o BCU que o conecta à rede.

O módulo de aplicação contém valores de *input* e *output*, que podem ser lidos e/ou escritos. Este módulo possui duas interfaces diferentes de comunicação, que consequentemente, permitem aceder a estes valores de forma diferente. Por um lado, através de uma interface paralela PEI I/O, estes valores de *input* e *output* podem ser acedidos pela aplicação interna do BAU. Por outro lado, pode existir comunicação entre o BAU e o módulo de aplicação através de uma interface série PEI, sendo este segundo módulo responsável pela leitura e escrita dos referidos valores.

Para comunicação série PEI, podem ser dados três exemplos típicos:

- O módulo de aplicação contém registos de 8 bits que podem ser lidos ou escritos.
- O módulo de aplicação contém o seu próprio micro controlador que executa uma aplicação externa do utilizador. Isto implica que o dispositivo EIB/KNX será um sistema de dois processadores (um no BCU e outro no módulo de aplicação) com comunicação série.
- O módulo de aplicação é um PC que comunica com a rede assincronamente via interface série. Neste caso, a aplicação externa será uma ferramenta de gestão, do género do referido anteriormente, ETS.

O protocolo PEI possui 21 tipos diferentes, entre os quais os 0, 1 e 20 são utilizados para fins especiais, os 3, 5, 7, 9, 11, 13, 15 e 18 estão reservados para extensões futuras, os tipos 2, 4, 6, 8, 17 e 19 são utilizados na comunicação paralela PEI I/O. Por fim, os tipos, 10, 12, 14 e 16 estão reservados para comunicação série PEI [1].

Cada um dos tipos PEI 10, 12, 14 e 16 suportam diferentes versões do protocolo. O tipo 16 consiste em comunicação assíncrona. O tipo 12 suporta comunicação síncrona com interface de mensagens. O tipo 14 suporta também comunicação síncrona mas com interface de blocos. O tipo 10 é assíncrono tal como o tipo 16, mas adicionalmente suporta a definição de um protocolo próprio entre o módulo de aplicação e o BCU [1].

Os sistemas EIB/KNX não têm necessariamente de possuir uma *Physical External Interface*, sendo permitidas outras formas de comunicação, entre o BCU e o módulo de aplicação, como por exemplo, memória RAM partilhada, ou um outro protocolo não baseado na interface PEI. Existe ainda o caso em que um dispositivo EIB/KNX é totalmente integrado, isto é, não existe qualquer separação entre o BCU e o módulo de aplicação [1].

Como já foi referido anteriormente, a comunicação entre um PC que corre uma aplicação de gestão e um sistema EIB, é feita tipicamente através de comunicação série síncrona. Sendo assim, para a criação do referido adaptador entre uma rede EIB/KNX e uma aplicação baseada

no sistema DomoBus, será utilizada a interface PEI *type* 16. Desta forma, seguidamente iremos abordar a comunicação série por interface PEI.

2.2.6.1. Serial PEI Communication

Recapitulando, um Bus Coupling Unit (BCU) é um caso particular de um Bus Access Unit (BAU), que adicionalmente fornece uma interface PEI (Physical External Interface), para comunicação com o módulo de aplicação. Vinte e um tipos diferentes de interfaces PEI, divididos em quatro categorias, permitem diferentes formas de comunicação, desde IO digital/analógico a comunicação série/paralela [7].

Ao conectar um PC, através da sua interface de dados RS-232, a uma rede EIB, tipicamente é utilizado um BCU (para este efeito designado BCU1, e um adaptador entre a sua interface e o PC), e um protocolo PEI Type 16, permitindo assim uma comunicação série assíncrona [7].

Formato das Mensagens

Uma mensagem (datagrama) trocada entre o módulo de comunicação externo (PC) e o BCU tem que seguir um formato predefinido. Um datagrama de tamanho n é composto por um octeto contendo o tamanho da mensagem e $n-1$ octetos contendo o conteúdo da mensagem, como ilustrado na figura seguinte [1].



Figura 14 – Formato de uma mensagem do protocolo PEI Type 16

O octeto respectivo ao tamanho contém o número de octetos seguintes a serem enviados. O primeiro bit (mais significativo) contém a paridade de todo o octeto. O segundo e o terceiro bit são sempre definidos com o valor "01", com a finalidade de identificarem o octeto como sendo relativo ao tamanho da mensagem. Os cinco bits seguintes representam então o número de octetos contidos na mensagem, podendo este valor variar entre 0 e 31, sendo que, numa mensagem não vazia, o primeiro deste octetos está reservado para o código da mensagem, podendo então os dados do utilizador ocupar até 30 octetos.

P	0	1	X	X	X	X	X
---	---	---	---	---	---	---	---

Figura 15 – Octeto relativo ao tamanho de uma mensagem

O conteúdo das mensagens relativo ao código da mensagem e dados do utilizador são definidos pelas normas External Message Interface (EMI). As primitivas EMI permitem o acesso, por parte da aplicação externa executada no PC, às camadas de comunicação do BCU. No modo normal de funcionamento, o PC pode compor e receber indicações ou confirmações de pacotes EIB/KNX direccionados ou originados nas camadas de comunicação do sistema. No modo denominado *busmonitor*, o PC apenas pode monitorizar a rede, sendo o BCU responsável por recolher pacotes da rede redireccionando-os para si [7].

A selecção entre uma das camadas de comunicação do BCU ou o modo *busmonitor* é efectuada através da primitiva `PC_SetValue.req`, alterando o byte relativo ao estado do sistema [7].

Desta forma, o octeto da mensagem relativo ao *message code* define a primitiva EMI, relativa ao comando que se pretende enviar. Consequentemente, os octetos da mensagem, relativos ao *user data*, dependem do comando a executar, isto é, do *message code*. O *message code* mais relevante é o utilizado nas mensagens de grupo cujo valor é `0x11`.

Descrição do Protocolo da Interface PEI

Para que as mensagens acima descritas sejam transmitidas entre a aplicação externa e o BCU através do protocolo PEI, são necessárias quatro fases:

1. Pedido de comunicação (*hardware handshake*)
2. Transferência do byte de tamanho (*software handshake*)
3. Transferência de dados
4. Pausa

Através dos sinais RTS (*Ready To Send*) e CTS (*Clear To Send*), ocorre um *hardware handshake* por cada byte a transferir. Antes de transmitir um byte, o iniciador da comunicação (quer seja o PC ou o BCU) tem que activar a sua linha de RTS. Seguidamente este necessita de monitorizar a sua linha de CTS. Se o sinal de CTS for activado significa que o receptor aceitou o pedido de ligação (isto é, o *hardware handshake* teve sucesso) e é permitida a transferência de um byte. Assim que o último bit foi enviado, o emissor desactiva o seu sinal de RTS e verifica se o receptor também desactivou o sinal de CTS. Neste caso significa que o byte foi transmitido com sucesso e que pode passar à transmissão do próximo byte [7].

O passo seguinte consiste na transmissão do byte relativo ao tamanho da mensagem (*software handshake*). Se um dos intervenientes não tem nada para enviar, transmite a sequência de bits 0xFF, sinalizando que irá ocupar o papel de receptor, enquanto o emissor envia o tamanho da mensagem a enviar. No caso de pedidos de envio simultâneos por parte do PC e do BCU, este último terá prioridade sobre o PC, sendo-lhe garantido acesso à linha. Desta forma, o PC terá que esperar até que o BCU tenha enviado completamente a sua mensagem, para que possa então iniciar um novo pedido de transmissão de dados [7].

O *software handshake* termina assim que o byte relativo ao tamanho da mensagem tenha sido transmitido. Isto significa que após esta fase, qualquer outro pedido de comunicação, independentemente do emissor, terá que esperar até que a mensagem actual seja totalmente transmitida.

Assim que a ligação é correctamente estabelecida, o emissor envia os bytes relativos ao conteúdo da mensagem, sendo a linha de RTS e CTS reiniciada entre cada envio. Em paralelo, o receptor responde com bytes de valor 0x00 [7].

Depois da transferência de uma mensagem ter terminado, um próximo acesso ao canal de comunicação poderá ser iniciado após um período de espera de 3 ms.

A figura seguinte ilustra as várias fases de transferência de uma mensagem [7].

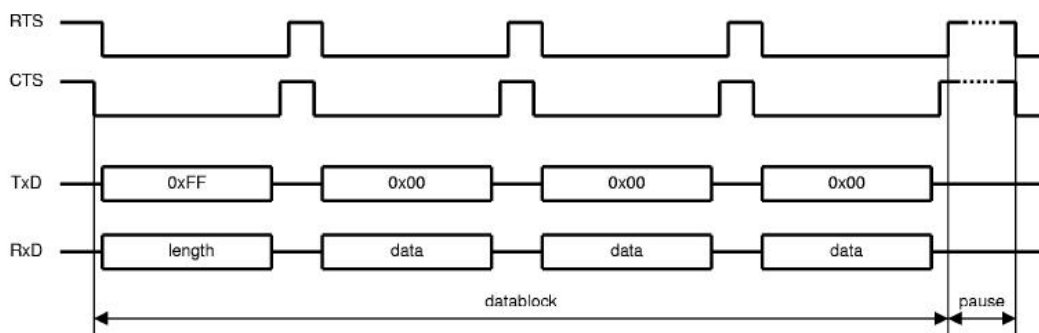


Figura 16 – Protocolo PEI Type 16

Em caso de erro, como erros de protocolo ou time-outs, o BCU reinicia a porta de série. Seguidamente este activa sua linha RTS e sonda a linha de CTS até que esta esteja também activa. Quando a linha de CTS estiver activa, o BCU espera 10 ms antes de enviar um novo pedido de transferência de dados. Caso contrário, considera o pedido por parte da aplicação externa (PC). O período de time-out considerado na transferência de um bloco de dados é de cerca de 130 ms.

Após um hardware ou software reset, o BCU tenta enviar uma mensagem `LM_Reset.ind`, mas apenas uma vez. Se ocorrer algum erro de comunicação durante esta transmissão, não será repetida. A mensagem `LM_Reset.ind` é composta por um único byte `A0h`, ou seja, um byte de tamanho indicando uma mensagem vazia [1].

P	0	1	L	L	L	L	L	- Formato normal do byte de tamanho.
1	0	1	0	0	0	0	0	- Mensagem <code>LM_Reset.ind</code>

Figura 17 – Formato mensagem `LM_Reset.ind`

2.3. DomoBus

2.3.1. Introdução

O sistema DomoBus foi desenvolvido como um projecto académico com o intuito de ultrapassar várias dificuldades ao tentar utilizar diferentes produtos para efeitos de testes e experimentação. O DomoBus tem assim sido usado como ferramenta de aprendizagem e para testar novas ideias e aplicações. Este sistema permite a implementação de soluções complexas minimizando o seu custo, para além da possibilidade de interligação com outras redes, permitindo assim atingir uma solução integrada. A arquitectura do DomoBus permite ainda uma aproximação distribuída onde múltiplos módulos de supervisão podem coexistir dentro da mesma rede, sendo assim possível que diferentes zonas de uma rede possam ser geridas por módulos diferentes, trazendo benefícios relativamente a balanceamento de carga e fiabilidade [4].

Um sistema DomoBus possui duas vertentes: uma vertente de baixo nível que consiste na rede de dispositivos domóticos DomoBus, composta por sensores, actuadores e componentes de sistema, como os módulos controladores e módulos de routing. A segunda é uma vertente de alto nível, composta pelas aplicações de interface com o utilizador, o serviço de comunicações, e as normas de interoperação entre dispositivos. A vertente de maior relevância para o desenvolvimento do referido trabalho final de curso que deu origem a este artigo, é a vertente de alto nível, pois é esta vertente que foi desenhada com o intuito de permitir a interoperação com outros sistemas, tornando-a assim no ponto principal de abordagem desta secção do artigo. Por outro lado, com o intuito de mostrar as principais diferenças entre este sistema e os sistemas restantes, seguidamente será abordada de uma forma geral a arquitectura deste sistema.

2.3.2. Visão Geral da Arquitectura

O sistema DomoBus é composto, essencialmente por Módulos de Controlo (CM – “Control Modules”) e Módulos de Supervisão (SM – “Supervision Modules”), que estão conectados através de uma rede, permitindo-os interagir e cooperar entre si [4].

Os Módulos de Controlo (CM) são placas com microprocessadores que se conectam directamente a interruptores, sensores de temperatura, receptores de infravermelhos e outros dispositivos de “input”. Estes módulos podem controlar dispositivos eléctricos, como por exemplo, ajustar intensidade de luzes ou ligar e desligar pequenos motores, iluminação, aquecedores eléctricos ou sistemas de ar condicionado. Podem ainda executar acções mais específicas como gerar sinais de infravermelhos para comandar televisões ou sistemas Hi-Fi [4].

Os CM actuais são bastante flexíveis, pois usam micro controladores que incluem código, memória e vários periféricos para efeitos de temporizações e comunicação. Cada CM possui um transmissor (EIA-485) e interfaces para leitura de interruptores e outros dispositivos de “input”. Cada CM pode executar diferentes aplicações e consequentemente executar diferentes funcionalidades. Esta capacidade consiste numa vantagem perante os outros sistemas domóticos existentes, nos quais, habitualmente, um módulo de controlo apenas controla um dispositivo. Desta forma, uma solução domótica através deste sistema, pode tornar-se economicamente mais vantajosa, permitindo que um CM controle diversos dispositivos [4].

Num típico sistema EIB por exemplo, um interruptor é composto por um módulo que incorpora um micro-controlador, com periféricos, fonte de alimentação, hardware de comunicação e o próprio interruptor em si. Por outro lado, o sistema DomoBus usa um CM, separado do interruptor, possivelmente “arrumado” juntamente com outros CM’s, e funciona como um normal interruptor eléctrico. Esta aproximação permite o uso de interruptores normais e outros dispositivos mais comuns, tornando o sistema global mais barato e vantajoso em vários aspectos [4].

A rede DomoBus pode ainda ser decomposta em vários segmentos, através de módulos de routing que ligam os subsegmentos ao segmento principal. Esta abordagem permite um elevado tráfego na rede, pois comunicações que circulam dentro de um segmento não afectam as comunicações dos outros segmentos [4].

Os Módulos de Supervisão (SM) são responsáveis pela gestão e supervisão do sistema. Estes recebem informação dos CM e, de acordo com regras e comportamento pré estabelecido, processam a informação recebida, enviando posteriormente aos CM a resposta adequada. Um sistema pode ser composto por vários SM dependendo do número de dispositivos que compõem a rede. Uma pequena rede, pode conter apenas um SM, mas em caso de uma rede de larga escala, pode existir um SM por cada subsegmento. Esta abordagem permite uma supervisão distribuída da rede, sendo cada CM gerido pelo SM do seu segmento da rede, trazendo benefícios no que diz respeito a tempos de resposta e fiabilidade da rede. Os SM de uma rede podem ainda trocar informação entre si para coordenar respostas a um determinado

evento. Eventualmente, esta troca de informação pode até ser executada através de uma rede diferente (Ethernet ou WLAN, por exemplo). Esta funcionalidade dos SM permite ainda interacção com outros sistemas, abrindo a possibilidade de atingir soluções integradas [4].

A figura seguinte esquematiza a arquitectura do sistema DomoBus [4].

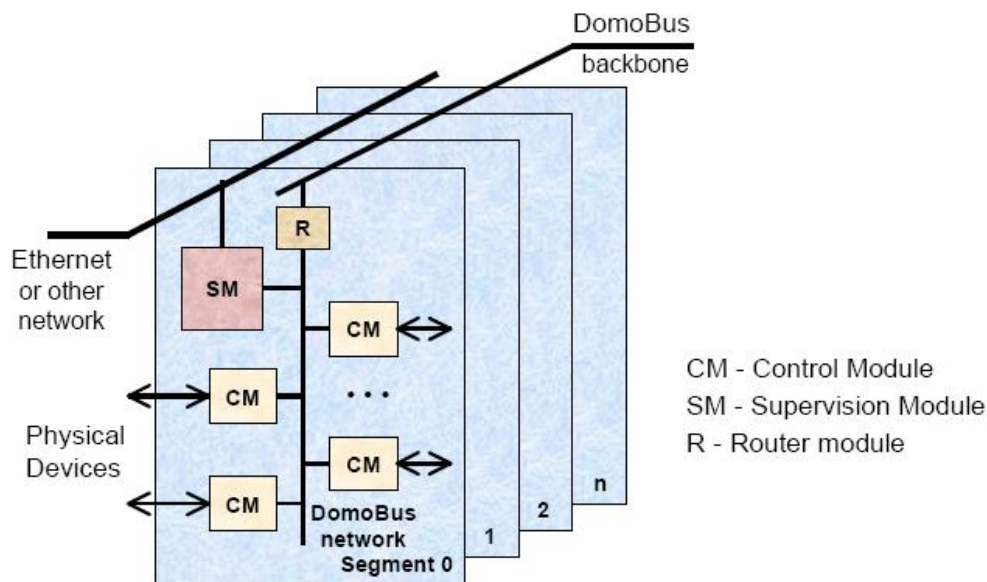


Figura 18 – Arquitectura do sistema DomoBus

2.3.3. Modelo de Abstracção de Dispositivos

Um dos principais objectivos do sistema DomoBus consiste em abordar a domótica de um modo genérico e flexível, permitindo representar qualquer dispositivo. Desta forma, possui uma modelo de abstracção que representa os dispositivos de uma forma tão simples quanto possível mas poderosa o suficiente para descrever todas as capacidades do dispositivo independentemente da sua tecnologia [3].

Com este objectivo, este modelo de abstracção permite representar qualquer dispositivo, caracterizando-o através de um conjunto de propriedades, tendo cada propriedade um valor. No caso de um sensor de temperatura, este pode ter apenas uma propriedade que contem o valor da temperatura medido pelo sensor. No caso de um dispositivo de iluminação com intensidade variável, este pode ser representado através de duas propriedades. Uma das propriedades pode ser um valor booleano que indica se o dispositivo está ligado ou desligado, enquanto a segunda pode ser um valor inteiro que indica, numa escala de 1 a 100, a

intensidade da luz. O uso independente destas duas propriedades permitiria, por exemplo, desligar ou ligar o dispositivo sem alterar a sua intensidade luminosa previamente definida [3].

Este modelo consegue descrever dispositivos tão simples como o referido no exemplo anterior, bem como dispositivos mais complexos, como uma televisão, com propriedades básicas como o estado ligado ou desligado, volume de som, canal, ou propriedades mais complexas como o nível de brilho, cor ou contraste [3].

A figura seguinte expressa em UML o modelo de abstracção de um dispositivo, utilizado no sistema DomoBus, que será descrito de seguida [3].

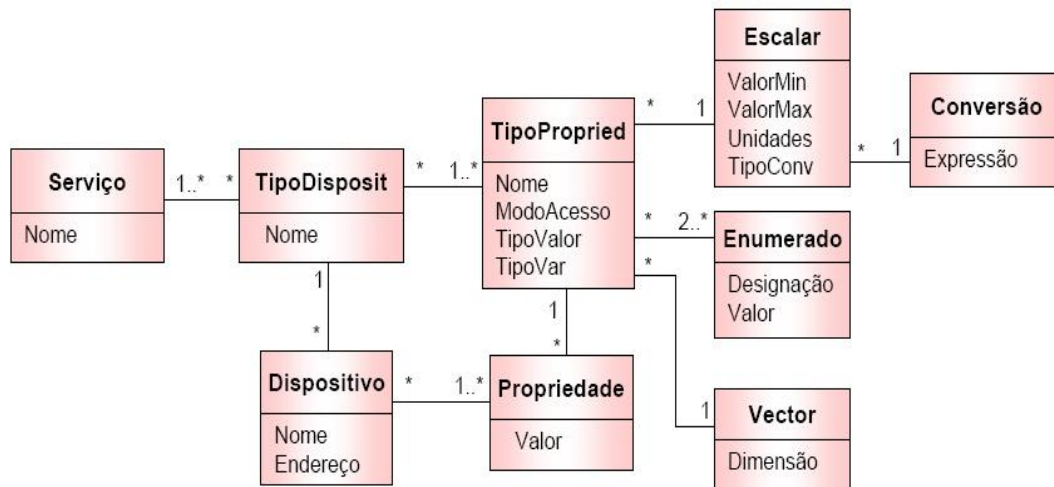


Figura 19 – Modelo de um dispositivo doméstico

A representação de um dispositivo possui uma entidade *TipoDispositivo* que distingue diferentes dispositivos como lâmpadas, sensores de temperatura, interruptores, sistemas de ar condicionado, electrodomésticos, etc. Este modelo permite facilmente adicionar novos tipos de dispositivos, bastando apenas que seja possível descrever as suas propriedades [3].

Por sua vez, a classe *Serviço* permite agrupar dispositivos que fornecem funcionalidades semelhantes. Exemplos de tipos de serviços podem ser: iluminação, climatização, segurança, entretenimento, etc. Um dispositivo pode ser associado a diferentes serviços, o que é útil, por exemplo, para caracterizar sensores de presença, que podem ser utilizados num sistema de vigilância, bem como num sistema de iluminação. Esta classe permite uma indexação de dispositivos por funcionalidade, facilitando, por exemplo, o caso de um utilizador que pretende aceder ao serviço de climatização, podendo aceder a todos os dispositivos responsáveis por este serviço, como sistemas de ar condicionado, aquecedores e sensores de temperatura [3].

A classe *TipoPropriedade* serve para caracterizar todas as propriedades de um tipo de dispositivo. Esta classe é composta pelo nome da propriedade, que pode ser *temperatura*, *intensidade*, etc. O atributo *ModoAcesso* define se a propriedade é apenas de leitura, escrita ou

leitura e escrita, que fará todo o sentido por exemplo, no caso de um sensor de temperatura, que obviamente não deverá poder ser escrito, mas no caso de um sistema de iluminação de intensidade variável, o valor da intensidade da luz poderá ser lido e modificado. Segundo este modelo, a interacção com os dispositivos físicos do sistema DomoBus, são realizadas exclusivamente através de leitura e escrita das propriedades dos dispositivos [3].

O atributo *TipoValor* da classe *TipoPropriedade* permite distinguir três tipos de valores: *Escalar*, *Enumerado* e *Vector*. O tipo *Escalar* indica um valor que varia entre um valor mínimo e um valor máximo. Este valor pode pertencer a um tipo de unidades, como por exemplo graus Celsius, no caso de um sensor de temperatura, ou um valor em percentagem numa escala de 0 a 100, ao até pode ainda não possuir qualquer tipo de unidades. Um *escalar* pode ainda estar sujeito a uma conversão, indicada pela variável *TipoConv*. Para estes casos, existe uma classe *Conversão* que contém a expressão simbólica a utilizar a efectuar a conversão. Dando um exemplo prático em que esta funcionalidade seria útil, consideremos um sensor de temperatura com uma precisão de 0,1°C, variando entre -25,0°C e 65,0°C. A representação para este caso seria com valores entre -250 e 650, sendo depois a expressão de conversão 0,1x [3].

Este valor *escalar* é ainda utilizado para representar valores booleanos, para propriedades que apenas possuem os estados ligado e desligado. Nestes casos o valor 0 corresponde ao falso e o valor 1 ao verdadeiro [3].

Para representar um sistema idêntico a um ar condicionado com quatro posições: desligado (0), ventilar (1), arrefecer (2) e aquecer (3), podemos também utilizar um valor *escalar*. No entanto o tipo *Enumerado* permite uma representação do estado de forma mais amigável permitindo representar o valor da propriedade em vez do seu valor numérico. Quando uma propriedade é deste tipo, está associada a dois ou mais objectos da classe *Enumerado*, contendo esta classe dois atributos que designam o seu nome e respectivo valor [3].

O terceiro tipo de valores é o tipo *Vector* que é usado em propriedades que contêm uma lista de caracteres (representando um texto ou outro tipo de informação) [3].

Passando à classe *Dispositivo*, esta tem como objectivo definir os dispositivos concretos de uma habitação, contrariamente à classe *TipoDispositivo*, que apenas definia os tipos base de dispositivos existentes. Podemos considerar por exemplo a definição de um dispositivo designado por *LâmpadaQuarto*, sendo este associado ao tipo de dispositivo *LampadaRegulável*, previamente definido. Deste modo, tornam-se imediatamente conhecidas as propriedades que possui. Este dispositivo fica assim associado a todas as propriedades que possui o tipo de dispositivo a que pertence, podendo ser então definidos todos os valores para estas mesmas propriedades [3].

Este modelo utilizado no DomoBus é então muito flexível e eficaz. Após a definição dos tipos de dispositivos mais habituais, torna-se muito simples adicionar dispositivos concretos que se pretende instalar numa habitação. Neste caso só são necessários adicionar novos tipos de dispositivos para casos de aparelhos menos comuns, mas até esta definição de novos tipos é bastante simples [3].

Este modelo permite ainda representar o tempo a partir de um dispositivo relógio. Este dispositivo possui as normais propriedades: segundos, minutos, horas, dias, meses e anos. Pode ainda representar outras propriedades mais abstractas como a distinção entre dia e noite, fins-de-semana ou feriados. Este dispositivo é representado da mesma forma que outro qualquer dispositivo, tornando possível programar o comportamento de um sistema com base no tempo, de forma genérica e uniforme [3].

2.3.4. Serviço de Comunicações

O sistema DomoBus oferece um serviço de comunicação, de seu nome ComServ, de forma a simplificar a comunicação entre aplicações DomoBus, independentemente da sua localização [14].

Este serviço ComServ (ver fig. 20) consiste num processo que analisa e encaminha as mensagens entre aplicações DomoBus, quer sejam aplicações locais, remotas ou executadas em controladores domóticos DomoBus de Baixo nível.

A comunicação entre o ComServ e as aplicações ocorre através de uma interface de sockets, utilizando datagramas UDP. Desta forma, qualquer aplicação que pretenda enviar uma mensagem para a rede, apenas tem que a escrever para o porto de escuta do ComServ, que é conhecido à partida.

Cada aplicação numa rede DomoBus possui um socket, a partir do qual recebe mensagens, bem como um endereço DomoBus. Quando esta se inicia, envia uma mensagem de registo ao ComServ, na qual está descrito o seu endereço e porto do seu socket para recepção de mensagens. O ComServ guardará então registo de todas as aplicações activas, incluindo os seus endereços e portos de escuta, podendo em qualquer altura enviar-lhes mensagens [14].

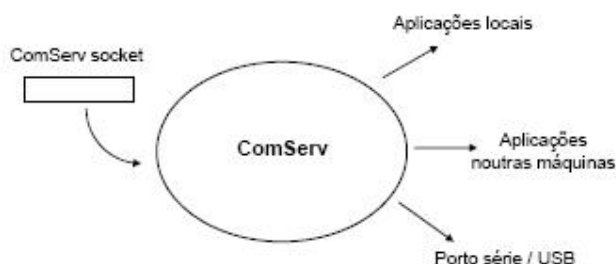


Figura 20 – Encaminhamento de mensagens pelo ComServ

Cada aplicação possui ainda um objecto responsável pelas comunicações, apelidado de objecto de comunicação (ver fig. 21). Este objecto é responsável pela criação do socket, registo no ComServ, recepção e envio de mensagens.

Uma vez que o protocolo UDP não é um protocolo totalmente fiável, as mensagens DomoBus devem ser confirmadas, permitindo assim conhecer se o envio de comandos por parte das aplicações foi recebido com sucesso. Adicionalmente, assume-se o modelo de comunicação “pelo menos uma vez”, em que se garante que qualquer mensagem enviada é recebida pelo destinatário pelo menos uma vez, não garantindo que esta foi única. De forma a resolver o problema causado pela recepção de mensagens repetidas, os comandos do sistema devem ser absolutos, ou seja, comandos repetidos não devem afectar o sistema. Desta forma não deveram existir comandos que alterem o estado de um dispositivo com base no seu estado anterior. Dando um exemplo, caso se pretenda ligar um dispositivo que esteja desligado, o comando a enviar deverá ser um comando de ligar explícito, e não apenas um comando para mudar o dispositivo para o seu estado oposto, pois desta forma mensagens repetidas iriam afectar o sistema.

Os objectos de comunicação são também responsáveis pela retransmissão de mensagens para as quais não ouve confirmação. Esta retransmissão de mensagens ocorrerá durante um período de tempo predefinido, ao fim do qual, caso não seja obtida nenhuma confirmação, será assumida uma falha e comunicada à aplicação.



Figura 21 – Objecto de Comunicação numa aplicação DomoBus.

2.3.5. Endereçamento

Numa rede DomoBus, os controladores domóticos (nós) estão interligados por diversos segmentos da rede. Cada nó possui então um conjunto de aplicações, tendo estas por sua vez, um conjunto de dispositivos (sensores e actuadores).

Cada dispositivo é endereçado a partir de um endereço físico de 32 bits, sendo os 24 bits mais significativos referentes à aplicação em que se encontra, e 8 para identificar o dispositivo. Assim, o encaminhamento das mensagens é baseado nestes 24 primeiros bits, relativos à aplicação, sendo depois esta que analisa os 8 bits menos significativos de forma a saber qual dos seus dispositivos é referido.

Os 32 bits do endereço de um dispositivo correspondem a diversos campos, como pode ser visto na figura seguinte [14].

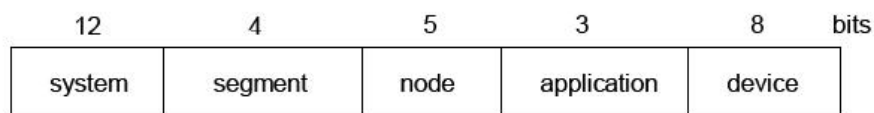


Figura 22 – Endereço DomoBus.

Como já foi referido, o ComServ é responsável pelo encaminhamento e transporte das mensagens entre as aplicações de controlo e a rede DomoBus. Desta forma, estas aplicações de alto nível não necessitam de tratar o diversos campos de endereço de uma mensagem, necessitando apenas de guardar os endereços de cada dispositivo. O envio de mensagens por parte das aplicações de alto nível segue o formato de mensagens DATA do ComServ. Estas mensagens, embora possuam um campo de endereço de 32 bits, uma vez que o endereçamento é só relevante às aplicações, só os primeiros 24 bits de cada endereço devem ser preenchidos, sendo o resto dos bits preenchidos com o valor “0”. No caso das mensagens terem como destino aplicações da rede DomoBus, a informação do dispositivo irá apenas no campo de dados, que será depois analisado pela aplicação de baixo nível correspondente [14].

2.3.6. Mensagens entre Aplicações e rede DomoBus

O protocolo DomoBus suporta vários tipos de mensagens, desde mensagens para as aplicações se registarem no ComServ, mensagens para fazer desregisto e mensagens de dados.

Como já foi referido, as aplicações de gestão comunicam através de mensagens segundo o formato DATA do ComServ, independentemente da aplicação de destino. A manipulação destas mensagens é da responsabilidade do objecto de comunicação de cada aplicação. A figura seguinte ilustra o formato deste tipo de mensagens [14].

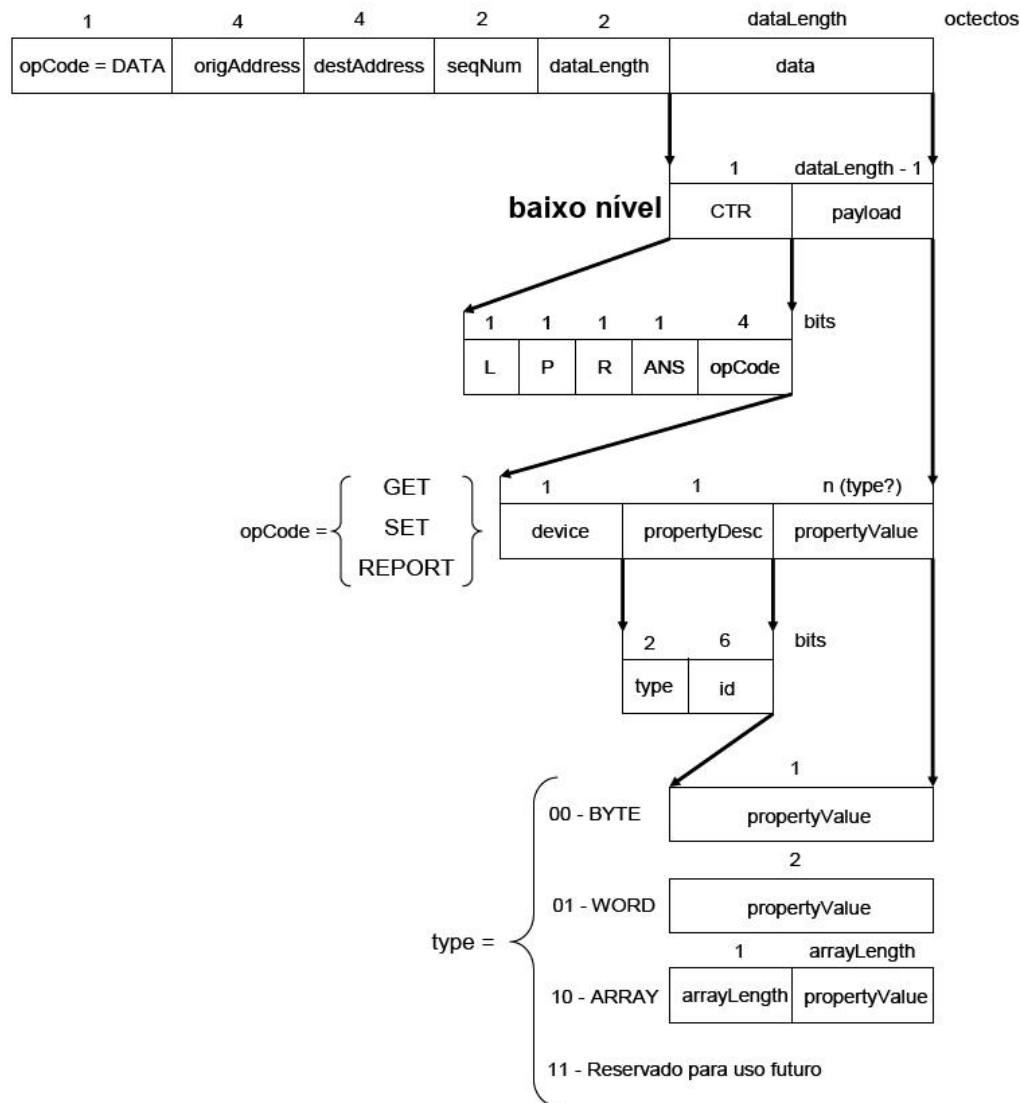


Figura 23 – Formato de mensagens entre aplicações e rede DomoBus

O primeiro octeto identifica o tipo de mensagem, que neste caso é do tipo DATA. Os campos seguintes dizem respeito aos endereços de destino e origem. Como já foi referido, no endereço de destino só devem estar preenchidos os 24 bits mais significativos, relativos à aplicação, sendo o dispositivo identificado mais tarde. Os dois campos seguintes consistem no número de sequência da mensagem e tamanho de dados nela contidos [14].

O campo seguinte contém os dados a enviar para o dispositivo. O primeiro byte contém informação de controlo como a seguir se descreve: destinado a futura extensão existe ainda um bit livre, o segundo bit especifica a prioridade da mensagem (alta ou baixa), o terceiro bit é utilizado como numeração de trama, para detecção de mensagens repetidas (para esta tarefa apenas é necessário um bit que é alternado entre os valores 0 e devido à utilização do protocolo stop-and-wait utilizado na transferência de mensagens), o quarto identifica se a mensagem é de pedido ou resposta, e os últimos quatro bits identificam a operação a executar

(e.g., GET, SET ou REPORT). O resto da mensagem contém um byte que identifica o dispositivo, um byte referente à propriedade a que se pretende aceder, e por fim, o resto da mensagem contém o valor da propriedade (podendo ser de qualquer um dos tipos suportados pelo sistema).

2.4. Outras tecnologias

Como referido anteriormente, existem diversas tecnologias no mercado para além do EIB/KNX. Estas tecnologias não foram escolhidas para uma primeira fase de implementação da solução apresentada nesta dissertação devido a algumas limitações que apresentam em relação ao EIB/KNX.

Esta secção irá então apresentar resumidamente outras tecnologias relevantes existentes no mercado, referindo as suas principais desvantagens em relação à tecnologia escolhida nesta dissertação.

2.4.1. X10

O X-10 é o protocolo mais antigo usado nas aplicações domóticas. Foi desenvolvido entre 1976 e 1978 com o objectivo de transmitir dados por linhas de baixa tensão (110V nos EUA e 230V na Europa) a muito baixa velocidade (60 bps no EUA e 50 bps na Europa) e com custos muito baixos. Ao usar as linhas eléctricas da habitação, não se torna necessário ter novos cabos para ligar os dispositivos [16].

O protocolo X-10 é proprietário mas a sua patente já expirou pelo que, actualmente, qualquer fabricante pode produzir dispositivos X-10 e oferecê-los ao público.

Graças ao seu amadurecimento (mais de 20 anos no mercado) e à tecnologia implementada, os produtos X-10 têm um preço muito competitivo sendo líderes no mercado residencial Norte-Americano com as instalações a serem realizadas por electricistas sem conhecimentos de automação ou informática ou até pelos próprios utilizadores. O preço e a facilidade de instalação são de facto os principais pontos fortes desta tecnologia.

Existem três tipos de dispositivos X-10: os que só podem transmitir ordens conhecidos por controladores, os que só podem receber ordens conhecido por receptores, e os dispositivos que podem receber e enviar ordens, que são na prática receptores com capacidade de responder e confirmar a realização correcta de uma ordem (feed-back) [16].

Os controladores enviam sinais de comando para os receptores que, por sua vez, fazem actuar o dispositivo eléctrico que lhe está ligado.

Os receptores são adaptadores que se instalam entre o dispositivo eléctrico que se pretende controlar e a fonte de corrente eléctrica que o alimenta. Os receptores vêm dotados de dois pequenos comutadores giratórios, um com 16 letras (código da casa) e o outro com 16

números (código do dispositivo), que permitem identificar um dos 256 endereços possíveis. Numa mesma instalação pode haver vários receptores configurados com o mesmo endereço, todos realizam a função pré-designada, desde que um controlador envie um telegrama com o seu endereço de destino.

A pequena gama de endereços existentes nesta tecnologia torna-a inadequada para grandes edifícios, dado que só permite a existência de 256 dispositivos receptores independentes. Esta é uma importante limitação relativamente às outras tecnologias existentes.

Qualquer acção num sistema X10 implica o envio de duas mensagens: mensagem de selecção do dispositivo e mensagem com a ordem a executar. De forma a minimizar possíveis falhas de comunicação, as mensagens são enviadas em duplicado. Este protocolo de comunicação torna o envio de comandos para os dispositivos um processo demasiado moroso, sendo este outro ponto fraco desta tecnologia.

Por outro lado, dado que este protocolo não inclui qualquer controlo sobre a correcta recepção de mensagens por parte dos dispositivos receptores, esta tecnologia apresenta um baixo nível de fiabilidade.

Concluindo, a tecnologia X10 é uma tecnologia com bastante sucesso dado o seu baixo custo e facilidade de instalação, no entanto apresenta bastantes limitações, comparando nomeadamente com o EIB/KNX. O X10 é apenas dirigido para pequenos ambientes residenciais dado a sua incapacidade para suportar muitos dispositivos, e a pouca fiabilidade, robustez e rapidez do seu protocolo de comunicação.

2.4.2. LonWorks

A Echelon Corporation apresentou a tecnologia LonWorks no ano 1992 e desde então múltiplas empresas a têm vindo a usar para implementar redes de controlo distribuídas e automatizadas. Apesar de estar desenhada para cobrir todos os requisitos da maioria das aplicações de controlo, só tem tido êxito a sua implementação em edifícios administrativos, hotéis e indústrias. Devido ao seu custo, os dispositivos LonWorks não têm tido grande implementação nas casas, sobretudo porque existem outras tecnologias com prestações idênticas e mais baratas [15].

Segundo a Echelon, o LonWorks é um sistema aberto a qualquer fabricante que queira usar esta tecnologia sem depender de sistemas proprietários, o que permite reduzir os custos e aumentar a flexibilidade. Apesar de a Echelon usar o conceito de “sistema aberto”, como veremos posteriormente, isto não é realmente verdade pois a tecnologia não se pode implementar se não for usado um circuito integrado registado pela Echelon [15].

Os dispositivos LonWorks usam um micro-controlador especial chamado Neuron Chip. O Neuron Chip tem um identificador único, o Neuron ID, que permite identificar qualquer

dispositivo de forma unívoca dentro de uma rede LonWorks. Este identificador com 48 bits é gravado na memória EEPROM durante o fabrico do circuito.

O sistema LonWorks, tal como o EIB/KNX possui um modelo de comunicação que é independente do meio físico sobre o qual funciona, isto é, os dados podem transmitir-se sobre par entrançado, corrente eléctrica, fibra óptica, radiofrequência, infravermelhos e cabo coaxial, entre outros.

Os dispositivos LonWorks comunicam entre si enviando telegramas que contêm o endereço do destinatário, informação para o routing, dados de controlo assim como os dados da aplicação do utilizador e um checksum para detecção de erros. Todas as comunicações de dados são iniciadas num Neuron Chip [15].

No que diz respeito aos fabricantes, a Echelon só concedeu licenças a três fabricantes que têm de pagar comissões por cada circuito fabricado. Para além disso o desenho do Neuron Chip permanece secreto e nenhum outro fabricante, para além destes três, pode fabricar este produto. Por estes motivos, ao não existir concorrência real e de a produção estar controlada pela Echelon, os preços não se reduziram tanto como deviam para permitir que os dispositivos LonWorks tenham um preço competitivo para as aplicações residenciais. Portanto, apesar de a Echelon se empenhar em dizer que o LonWorks é um sistema aberto, a realidade vem demonstrando que não o é.

Concluindo, embora a tecnologia LonWorks seja de facto bastante fiável e robusta, não apresenta grandes vantagens em relação ao EIB/KNX. Pelo contrário, a dificuldade de acesso ao Neuron Chip constitui uma grande desvantagem e o seu elevado custo torna pouco viável a sua utilização em sistemas residenciais.

2.4.3. CEBus

A tecnologia CEBus é um standard que surgiu nos Estados Unidos da América, estando normalizado pela EIA (Electronic Industries Association). Esta tecnologia apresenta algumas semelhanças com outras tecnologias nomeadamente o EIB/KNX.

Tal como acontece com o EIB/KNX e o LonWorks, o modelo de comunicação do CEBus pode ser implementado sobre fios de par entrançado, corrente eléctrica, cabo coaxial, fibra óptica, rádio frequência e infra vermelhos [17].

O CEBus suporta também uma topologia flexível. Um dispositivo pode ser instalado em qualquer localização em que seja conveniente e conectado a qualquer tipo de meio físico, desde que contenha a interface CEBus apropriada. As mensagens podem ser transferidas de um meio de comunicação para outro através de um circuito electrónico designado *router* [17].

Tal como no EIB/KNX, para além de mensagens individuais, todos os dispositivos ou grupos de dispositivos podem ser endereçados numa única mensagem através de endereços de grupo. Um dispositivo CEBus pode pertencer a um ou mais grupos. Contrariamente ao EIB/KNX, esta

subscrição de endereços de grupo não é da inteira responsabilidade do instalador do sistema, o fabricante do dispositivo pode escolher se este suporta endereçamento de grupo e quantos endereços de grupo pode subscrever.

Esta tecnologia é flexível mas apresenta custos relativamente altos. Apenas um número muito reduzido de empresas implementaram produtos CEBus pelo que possui uma penetração de mercado muito baixo (praticamente nula fora dos EUA). Isso deve-se, em parte, ao efeito concorrencial do X10 que é muito mais barato, mais simples de instalar e tem uma larga divulgação.

2.5. Conclusão

Este capítulo reflecte um trabalho de pesquisa realizado com o intuito de conhecer o estado da arte relativamente ao funcionamento dos sistemas domóticos, e mais especificamente adquirir os conhecimentos necessários para a criação de uma possível solução para os problemas anteriormente identificados nestas tecnologias.

Como foi referido anteriormente, as aplicações de monitorização e controlo existentes para sistemas domóticos, são aplicações específicas, desenvolvidas para uma dada tecnologia, uma dada habitação e com um comportamento predefinido, sendo consequentemente limitadas. A solução procurada para este problema irá permitir a criação de aplicações versáteis, seguindo o modelo DomoBus, tornando-se uma valiosa alternativa às ferramentas actualmente existentes. Por outro lado, esta solução abre portas para a possibilidade de criação de redes domóticas compostas por diferentes tecnologias, até agora inexistentes.

Dentro das tecnologias existentes no mercado, este capítulo focou-se nos sistemas EIB/KNX, dado que é uma tecnologia bastante completa, oferecendo um leque diversificado de possibilidades de utilização e tendo uma boa penetração no mercado, especialmente a nível europeu. Pelo mesmo motivo, esta foi também a tecnologia escolhida para o desenvolvimento da solução que irá ser apresentada neste documento. Desta forma, foram abordadas neste capítulo várias características importantes desta tecnologia, desde a sua topologia, aos níveis de interoperação necessários para interacção entre dispositivos, passando pelo protocolo de comunicação com a rede, peça fundamental para a implementação da solução pretendida.

A tecnologia DomoBus aqui apresentada, embora ainda numa fase inicial de desenvolvimento, possui características ideais para a utilização pretendida nesta solução, nomeadamente pela sua versatilidade, e pelo seu serviço de comunicações aqui apresentado, que irá permitir a interligação entre diversas redes de dispositivos.

3. DESCRIÇÃO DA SOLUÇÃO

Como foi referido no capítulo anterior, o sistema DomoBus foi desenhado com o intuito de ser simples, versátil e ao mesmo tempo bastante poderoso. Esta versatilidade pode ser encontrada a vários níveis, como no seu modelo de abstracção de dispositivos. Esta representação de dispositivos a partir do seu conjunto de propriedades (ou funcionalidades) permite à partida representar qualquer tipo de dispositivos, independentemente da sua tecnologia. Isto implica que todos os dispositivos EIB/KNX podem ser representados desta forma, recorrendo a propriedades que correspondam às suas funcionalidades. Na prática, dando um exemplo concreto, um módulo EIB/KNX de iluminação de intensidade variável, pode ser representado segundo o modelo DomoBus como um dispositivo composto por duas propriedades, sendo uma respeitante ao seu estado (ligado/desligado) e outra respeitante à sua intensidade.

Por outro lado, as aplicações de monitorização e controlo segundo a tecnologia DomoBus, são desenhadas de forma versátil e facilmente configuráveis. Como foi descrito no capítulo anterior, uma aplicação de monitorização é configurada a partir de ficheiros XML, no qual são inseridos todos os aspectos relativos à rede a monitorizar, desde os dispositivos nesta contidos, suas propriedades, serviços a que estão associados, etc. Esta adaptabilidade permite a rápida configuração de uma nova rede ou a reconfiguração uma rede que tenha sido modificada, característica inexistente nas aplicações de monitorização e controlo de qualquer outra tecnologia domótica.

Estes dois aspectos, tornam as aplicações DomoBus capazes de representar uma rede de dispositivos de qualquer outra tecnologia nomeadamente o EIB/KNX. Estas aplicações associadas a outras tecnologias serão extremamente versáteis tornando-as uma escolha ideal para a resolução do problema anteriormente referido.

A comunicação entre as aplicações DomoBus, quer sejam estas de alto ou baixo nível, é controlada, como anteriormente referido, pelo serviço de comunicações ComServ. Este serviço é então responsável pela capacidade que esta tecnologia possui, de interligação de diferentes redes de dispositivos. Através de uma solução que permita efectuar a comunicação entre este serviço e redes de dispositivos de outras tecnologias, será então possível a criação de sistemas domóticos multi-tecnológicos, algo impossível até ao momento.

A solução aqui apresentada consiste então num adaptador ou gateway que irá permitir a interoperação entre o sistema DomoBus e o sistema EIB/KNX. Este adaptador será responsável por efectuar a comunicação entre o serviço de comunicações do DomoBus e uma rede de dispositivos EIB/KNX.

3.1. Arquitectura da Solução

A figura 24 ilustra a arquitectura da solução que irá permitir a interoperação entre o sistema EIB/KNX e o sistema DomoBus. O adaptador ou gateway EIB irá ser executado num computador com sistema Operativo Windows e com porta de série RS232.

O adaptador irá comunicar com o serviço de Comunicações DomoBus (ComServ) através de uma interface de sockets UDP, usando a API disponibilizada para este efeito. É a partir desta ligação que o adaptador irá receber mensagens provenientes das aplicações de controlo DomoBus com destino à rede de dispositivos. Devido às potencialidades do ComServ, estas aplicações de controlo poderão ser executadas localmente ou remotamente através da Internet.

Por outro lado, o adaptador possuirá uma interface (PEI Driver) que será responsável por comunicar com o BCU (Bus Coupling Unit) de ligação à rede EIB/KNX, através da porta série RS232. Esta comunicação será executada segundo o protocolo PEI Type 16, detalhado no segundo capítulo deste documento.

O adaptador EIB será então responsável pela recepção de mensagens provenientes do ComServ, conversão para o formato EIB/KNX e redireccionamento para a rede de dispositivos, efectuando o processo inverso com as respectivas respostas.

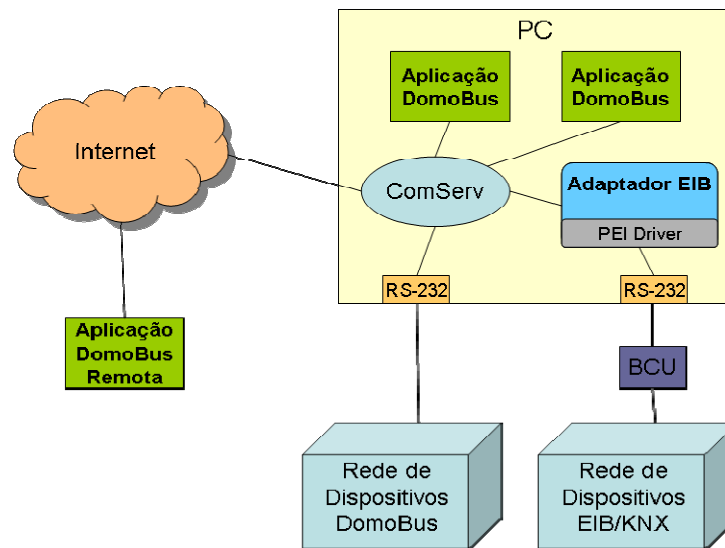


Figura 24 – Arquitectura da solução de integração DomoBus – EIB/KNX

3.2. Funcionamento da Solução

O serviço de comunicações ComServ possui um mecanismo de segurança contra o registo de aplicações DomoBus não fidedignas. Este mecanismo de segurança passa pela prévia configuração, a partir de um ficheiro que contém os endereços das aplicações DomoBus que possuem permissões para serem registadas neste serviço, com os respectivos endereços IP, quer sejam estes locais ou remotos. Desta forma, todas as aplicações a integrar com esta solução necessitam de ser previamente configuradas no DomoBus, nomeadamente todos os endereços de aplicações escolhidos para representar os dispositivos DomoBus.

Como referido no capítulo anterior desta tese, um dispositivo DomoBus possui um endereço de 32 bits, sendo os primeiros 24 referentes à aplicação de baixo nível que o controla. Desta forma, um sistema DomoBus poderá ser composto por diversas aplicações de baixo nível, e cada uma delas poderá ser responsável por 256 dispositivos. Para que uma rede de dispositivos EIB/KNX, a ser integrada nesta solução, possa ter mais do que 256 dispositivos, o adaptador EIB poderá então estar associado a vários endereços de aplicação DomoBus.

O adaptador DomoBus – EIB/KNX permite a conversão de mensagens entre os dois sistemas. Desta forma necessita de conhecer vários aspectos referentes aos dispositivos existentes na rede EIB/KNX. A execução desta aplicação inicia-se com a interpretação de ficheiros de configuração. Estes ficheiros, que serão descritos na próxima secção deste documento, contêm toda a informação necessária para que possa ser efectuada a tradução das mensagens, desde os seus endereços de origem e destino, ao próprio conteúdo da mensagem. Após a interpretação destes ficheiros, o adaptador poderá então efectuar a ligação com o ComServ e o registo dos seus endereços de aplicação DomoBus. A partir deste momento, o serviço de comunicações poderá comunicar com o adaptador EIB, como se de uma aplicação DomoBus se tratasse.

Para o correcto funcionamento desta solução, todas as aplicações de monitorização e controlo DomoBus que irão interagir com a rede de dispositivos EIB/KNX, deverão ser configuradas para tal. Esta configuração, como foi anteriormente referido, passa apenas pela criação de um ficheiro XML que representará todos os dispositivos existentes e suas características. Este ficheiro possuirá o formato previamente definido para a representação de dispositivos DomoBus, não tendo que ter nenhuma das suas características modificadas para o funcionamento com dispositivos de outra tecnologia. Os dispositivos EIB/KNX são representados normalmente, recorrendo ao modelo de abstracção utilizado pelo DomoBus.

Após o registo no ComServ, o adaptador EIB irá iniciar a comunicação com o BCU (Bus Coupling Unit) de ligação com a rede de dispositivos. Esta comunicação é iniciada através de um conjunto de mensagens de configuração trocadas entre o adaptador EIB e o BCU. Terminada esta configuração, o adaptador estará pronto para efectuar a comunicação com os dispositivos.

Durante o seu funcionamento normal, o adaptador escuta mensagens provenientes tanto do ComServ como da rede de dispositivos. Aquando da recepção de uma mensagem proveniente

do ComServ, esta é convertida para posterior envio para a rede de dispositivos. Esta conversão começa pela tradução do endereço de destino da mensagem, recorrendo ao conteúdo do ficheiro de configuração interpretado no arranque da aplicação. Seguidamente é identificada a propriedade a ler ou escrever, e neste segundo caso, o valor a escrever. Esta propriedade é então mapeada numa função EIS suportada pelo dispositivo (estas funções encontram-se descritas na secção 2.2.4.3 deste documento) mais uma vez, com base na informação contida no ficheiro de configuração. Poderá agora ser construída uma mensagem no formato suportado pelo EIB/KNX que será posteriormente enviada para a rede de dispositivos.

No caso de recepção de uma mensagem proveniente da rede de dispositivos, é efectuado o processo oposto. É identificado o endereço de origem e convertido para um endereço DomoBus, é identificada a função EIS e o respectivo valor e mapeados numa propriedade DomoBus. É construída a respectiva mensagem DomoBus e enviada para o ComServ.

Esta solução permite assim a leitura e escrita dos estados de todos os dispositivos EIB/KNX, por parte de uma aplicação de monitorização e controlo DomoBus, bem como o report espontâneo às mesmas de qualquer alteração do estado de um dispositivo.

3.3. Detalhes de Implementação

3.3.1. Mapeamento de endereços

O protocolo de comunicação DomoBus utiliza uma gama de endereços de 32 bits, no entanto o sistema EIB/KNX possui dois modos de endereçamento diferentes: endereçamento físico, utilizado na configuração dos dispositivos e enviado no campo de endereço de origem de uma mensagem, e o endereçamento de grupo, utilizado durante o normal funcionamento dos dispositivos. Como referido anteriormente, este modo de endereçamento permite tanto o envio de mensagens para um único dispositivo ou para um conjunto, consoante o número de dispositivos que subscrevam determinado endereço.

Desta forma, na conversão de uma mensagem proveniente do ComServ destinada a apenas um dispositivo, o respectivo endereço de destino DomoBus terá que ser convertido num endereço de grupo EIB/KNX. No entanto, o DomoBus não possui a noção de grupos de dispositivos, o que poderia implicar que esta funcionalidade do EIB/KNX fosse ignorada. Para resolver esta situação, optou-se pela criação da noção de dispositivo virtual DomoBus, que irá então corresponder a um grupo de dispositivos EIB/KNX. Tal como no EIB/KNX em que um grupo de dispositivos apenas pode subscrever o mesmo endereço de grupo se possuírem as mesmas características, um dispositivo virtual DomoBus apenas pode ser composto por dispositivos do mesmo tipo. Este dispositivo virtual representa-se assim abstractamente como

qualquer outro dispositivo, e possuirá um conjunto de propriedades características do tipo dos dispositivos que o compõem.

3.3.2. Ficheiros de Configuração

Para além dos ficheiros de configuração respectivos às aplicações de monitorização e controlo, onde são representados abstractamente os dispositivos a controlar, e do ficheiro de configuração do ComServ que garante a fiabilidade das aplicações neste registadas, o adaptador EIB possui também dois ficheiros de configuração do seu funcionamento.

O primeiro ficheiro, em formato XML, irá possuir toda a informação necessária ao funcionamento do adaptador, não relativa a dispositivos. De momento, este ficheiro possui apenas dois campos, mas é aqui que futuramente deverá ser adicionada qualquer informação que possa vir a ser necessária.

O primeiro campo deste ficheiro contém a porta de série a utilizar na comunicação dos dispositivos. Tipicamente a porta de série utilizada será a porta "COM1", no entanto diferentes computadores poderão possuir portas RS232 com diferentes designações, portanto será útil que tal possa ser configurado. O segundo campo corresponde ao endereço DomoBus da aplicação de monitorização e controlo para a qual devem ser reportadas espontaneamente todas as alterações de estado dos dispositivos.

Este ficheiro de configuração tem o seguinte formato:

```
<ConfigFile>
  <Port Name="COM1"/>
  <ReportTo Address="0x100100"/>
</ConfigFile>
```

O segundo ficheiro de configuração possuirá a informação necessária para efectuar a tradução das mensagens entre o sistema DomoBus e o sistema EIB/KNX. Na prática, este ficheiro possui informação que permitirá efectuar o mapeamento entre os endereços dos dispositivos, e o mapeamento das suas propriedades DomoBus em funções EIB.

Este ficheiro tem o seguinte formato:

```
<DomoBus-KNX_mapping>
  <DeviceMappingList>
    <DeviceMapping DomoBusAddress="0x00010001" DeviceTypeID="1"
KNXPhysicalAddress="1105">
```

```

        <Property ID="1" KNXGroupAddress="0101"/>
        <Property ID="2" KNXGroupAddress="0102"/>
    </DeviceMapping>
    <DeviceMapping DomoBusAddress="0x00010002" DeviceTypeID="1"
KNXPhysicalAddress="1106">
        <Property ID="1" KNXGroupAddress="0201"/>
        <Property ID="2" KNXGroupAddress="0202"/>
    </DeviceMapping>
    <DeviceMapping DomoBusAddress="0x00020001" DeviceTypeID="2"
KNXPhysicalAddress="1201">
        <Property ID="1" KNXGroupAddress="200"/>
    </DeviceMapping>
</DeviceMappingList>
<PropertyMappingList>
    <PropertyMapping DeviceTypeID="1" PropertyID="1"
KNXFunctionType="EIS1"/>
    <PropertyMapping DeviceTypeID="1" PropertyID="2"
KNXFunctionType="EIS5"/>
    <PropertyMapping DeviceTypeID="2" PropertyID="1"
KNXFunctionType="EIS3"/>
</PropertyMappingList>
</DomoBus-KNX_mapping>

```

Como foi referido no capítulo referente ao DomoBus desta dissertação, segundo o modelo de abstracção desta tecnologia, um dispositivo é representado através de um conjunto de propriedades. Uma mensagem DomoBus é assim enviada com o endereço do dispositivo a aceder, bem como a respectiva propriedade. Desta forma, para que esta mensagem possa ser convertida numa mensagem EIB/KNX é necessário converter o endereço de destino DomoBus num endereço de grupo KNX, e a propriedade DomoBus numa funcionalidade suportada pelo dispositivo EIB/KNX.

Este ficheiro é assim dividido em duas secções diferentes: “DeviceMappingList” e “PropertyMappingList”. A primeira secção diz respeito ao mapeamento de endereços. Por cada dispositivo existente do ponto de vista do DomoBus, existe uma entrada “DeviceMapping” no ficheiro. Cada uma destas entradas possui os atributos referentes ao endereço DomoBus do dispositivo, identificador do tipo de dispositivo e endereço físico KNX. O identificador do tipo de dispositivo será o mesmo utilizado no ficheiro de configuração das aplicações DomoBus, e será apenas necessário neste ficheiro para evitar replicação desnecessária de informação, como

será explicado mais à frente. Quanto ao endereço físico KNX, será um campo inexistente para algumas entradas, caso estas se refiram a um dispositivo virtual DomoBus. Este atributo será apenas necessário para a conversão do endereço de origem de uma mensagem proveniente de um dispositivo EIB/KNX.

Para cada entrada do tipo “DeviceMapping”, existem ainda várias entradas do tipo “Property”, consoante o número de propriedades que compõem o respectivo dispositivo, segundo o modelo de abstracção DomoBus. Estas entradas mapeiam as propriedades de cada dispositivo num endereço de grupo a ser utilizado pelo sistema EIB/KNX. Todas as funcionalidades de um dispositivo EIB/KNX necessitam de subscrever diferentes endereços de grupo, por não existir a noção de propriedades como no caso do DomoBus. Desta forma, não é possível que a tradução de um endereço DomoBus para um endereço de grupo KNX seja uma relação de um para um, é então necessário ter em conta a propriedade DomoBus que se está a aceder, para efectuar este mapeamento. Estes atributos “Property” serão idênticos quer se trate de um dispositivo DomoBus real ou virtual.

A secção de “PropertyMappingList” contém informação que permitirá efectuar o mapeamento de uma propriedade numa função suportada pelo EIB/KNX (“EIB Functions”). Esta secção é indexada pelo identificador do tipo de dispositivo, uma vez que para dispositivos iguais, as mesmas propriedades DomoBus representarão as mesmas funcionalidades. Esta identificação do tipo de dispositivo evitará então a necessidade de replicação desta informação em dispositivos idênticos.

3.3.3. Conversão de Mensagens

Embora a tecnologia EIB/KNX pretenda ser normalizada com um uso aberto, a “Konnex Association” comercializa produtos para o desenvolvimento de aplicações de comunicação, não sendo então do seu interesse a divulgação completa e detalhada de todo o funcionamento desta tecnologia. O processo de interpretação do conteúdo das mensagens EIB/KNX foi então um processo algo complexo, realizado com base na pouca documentação existente e na escuta de mensagens trocadas entre uma aplicação para configuração (ETS3), e a rede de dispositivos.

Por sua vez, na tecnologia DomoBus, o formato das mensagens encontra-se claramente detalhado, o que facilitou a sua interpretação.

Como antes referido, uma mensagem de dados DomoBus possui o seguinte formato (para uma representação mais detalhada, ver secção 2.3.6 deste documento):

8 bits	32 bits	32 bits	16 bits	16 bits	8 bits	8 bits	2 bits	6 bits	N bits
opCode	origAddr	destAddr	seqNum	dataLen	CTRL	device	propType	propId	propValue

Figura 25 – Formato mensagem DomoBus

O que se pretende com esta solução é então converter a informação contida numa mensagem DomoBus com o formato representado, numa mensagem EIB/KNX. Esta mensagem KNX será semelhante à detalhada na secção 2.2.5.2 deste documento, com diferenças mínimas dado que a anterior se refere a uma trama enviada para a rede por um BCU (Bus Coupling Unit), e a pretendida nesta situação será enviada para o BCU que posteriormente efectuará o preenchimento dos restantes campos. A mensagem a ser enviada ao BCU tem, resumidamente, o seguinte formato:

8 bits	8 bits	16 bits	16 bits	1 bits	3 bits	4 bits	10 bits	N bits
m_code	priority	Unused	destAddr	addrType	routing count	length	CTRL	data

Figura 26 – Formato mensagem EMI

A criação de uma mensagem de comunicação com um endereço de grupo KNX implica então a conversão de alguns campos da mensagem proveniente do DomoBus. Através da informação contida nos ficheiros de configuração acima descritos, a partir do endereço de destino DomoBus e do identificador da propriedade a aceder, é possível identificar o endereço de grupo com o qual se pretende comunicar. Através do mapeamento entre uma propriedade de um tipo de dispositivos DomoBus e a respectiva função EIB suportada, é possível determinar o tipo de dados a enviar na mensagem KNX, e consequentemente, preencher o campo “length” da mesma. Quanto ao campo de dados da mensagem KNX, poderá ser o mesmo que se encontra no campo “propValue” da mensagem DomoBus, ou em determinados casos, poderá ser necessário efectuar alguma conversão, tipicamente simples.

Os restantes campos da mensagem KNX são comuns a todas as mensagens, como é o caso do octeto relativo ao “message code” (valor “0x11” para todas as mensagens de grupo), e do octeto relativo à prioridade da mensagem, não sendo assim necessário qualquer tipo de mapeamento.

Quanto às mensagens de report, provenientes da rede de dispositivos EIB/KNX, o processo utilizado na sua conversão é semelhante ao inverso. Nesta situação, o formato das mensagens é idêntico, exceptuando o campo relativo ao endereço de destino da mensagem, que contrariamente ao apresentado, não vem vazio mas sim com o endereço físico do dispositivo em causa. Nestas mensagens de report, o campo relativo ao endereço de destino será

preenchido com o endereço de grupo ao qual a funcionalidade está associada. Toda esta informação permitirá a criação de mensagens de dados DomoBus de forma idêntica á descrita acima para o processo inverso.

3.3.4. Comunicação com sistema DomoBus

Fruto de um trabalho final de curso realizado anteriormente por um grupo de alunos, com o objectivo de simplificar todo o processo de comunicação entre aplicações DomoBus, foi desenvolvida uma API de comunicação a ser incorporada nas aplicações desenvolvidas para esta tecnologia. Esta API implementa o objecto de comunicação que permite a comunicação com qualquer aplicação DomoBus ligada ao sistema, quer seja esta local ou remota, através do anteriormente referido, serviço de comunicações ComServ.

Esta API foi desenvolvida em linguagem C e posteriormente estendida para a linguagem C#, materializando-se nas bibliotecas dinâmicas DomoBus_COMM.dll e DomoBus_COMM_CSharp.dll respectivamente. A comunicação da solução aqui apresentada com o sistema DomoBus foi desenvolvida recorrendo a esta segunda biblioteca.

A figura seguinte representa o objecto de comunicação implementado por esta API:

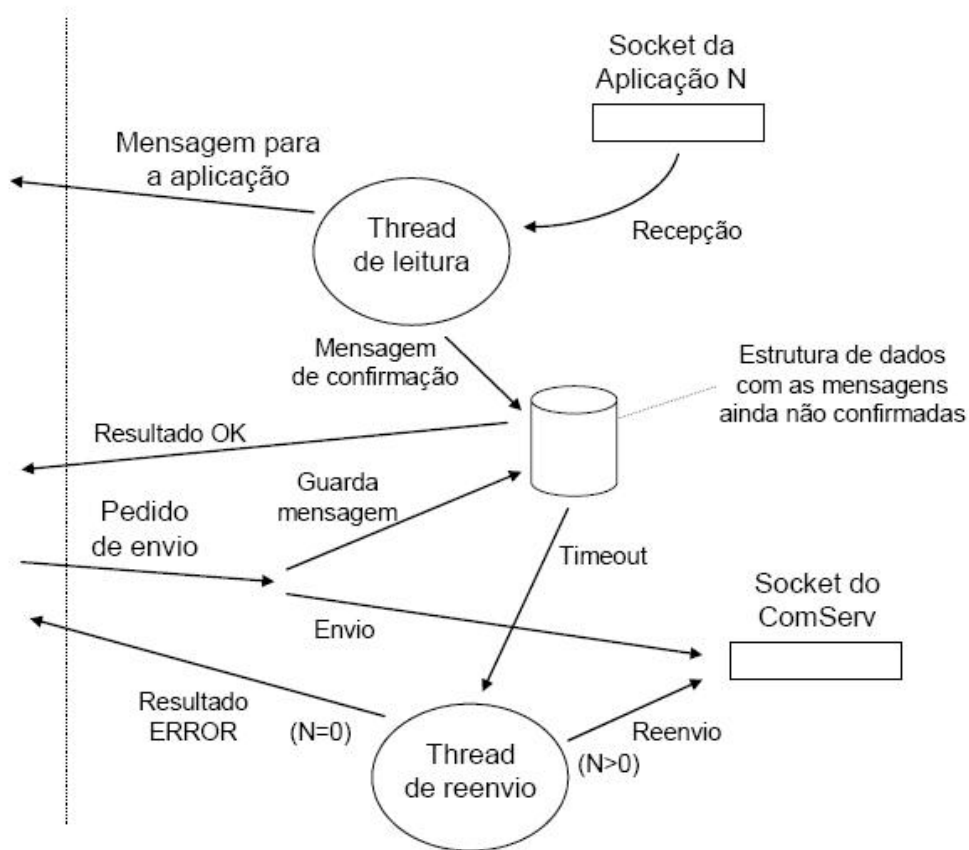


Figura 27 – Objecto comunicação DomoBus

O objecto de comunicação possui duas threads, uma de leitura do socket, a qual está bloqueada enquanto não chegar uma mensagem, e outra que trata do reenvio de mensagens que ainda não tenham obtido resposta. Quando é recebido um pedido de envio, a mensagem é escrita no socket do ComServ e é guardada uma cópia numa estrutura de dados para o caso de ser necessário o seu reenvio.

Quando chega uma nova mensagem ao socket de escuta, ela é analisada para verificar se é uma resposta. Se não for, a mensagem é entregue à aplicação. Se for uma resposta, é analisada a estrutura de dados que contém as mensagens a aguardar resposta para desbloquear as funções que esperavam pela resposta.

Na solução aqui apresentada, o adaptador EIB inicia o processo de comunicação invocando o método "DomoBus_CommOpen()" disponibilizado por esta API, repetidamente, registrando todos os endereços de aplicação que estão sobre sua responsabilidade.

Esta API disponibiliza também o método "DomoBus_GetMsg()" para a recepção de mensagens. Este é um método bloqueante que é libertado sempre que uma mensagem é recebida pelo objecto de comunicação. Desta forma, o adaptador EIB possui uma tarefa cuja

função é invocar este método, e sempre que uma mensagem é recebida, tratá-la e reencaminhá-la para o sistema EIB/KNX.

O envio de respostas para o sistema DomoBus é efectuado através do método "DomoBus_SendAnswer()" disponibilizado por esta API, e o envio de mensagens de report destinadas às aplicações de monitorização e controlo, através do método "DomoBus_SendMsg()".

Esta API disponibiliza ainda alguns métodos para o tratamento de mensagens DomoBus, extremamente útil que para a decomposição e interpretação de mensagens recebidas do sistema DomoBus, quer para a criação de mensagens a partir de informação recebida da rede de dispositivos EIB.

3.3.5. Comunicação EIB/KNX

A comunicação com a rede de dispositivos EIB é efectuada através da ligação entre o computador e um BCU (Bus Coupling Unit) via porta de série RS232.

Na solução aqui apresentada, foi desenvolvida uma classe C# que representa o objecto responsável por esta comunicação. Esta classe disponibiliza métodos públicos para iniciar a comunicação, terminar a comunicação, enviar mensagens para a rede e receber mensagens provenientes da rede. Esta classe torna assim a comunicação com a rede de dispositivos, o mais independente possível da aplicação que a invoca. Esta opção permite a modificação do código da classe de comunicação, evitando a necessidade de rescrever a aplicação que a instancia, e permite ainda a futura utilização da mesma, no desenvolvimento de outras aplicações em que seja necessário efectuar comunicação com redes de dispositivos EIB/KNX.

Esta classe de comunicação utiliza o protocolo PEI Type 16, descrito anteriormente na secção 2.2.6 deste documento.

A figura seguinte representa o normal funcionamento do objecto de comunicação implementado por esta classe:

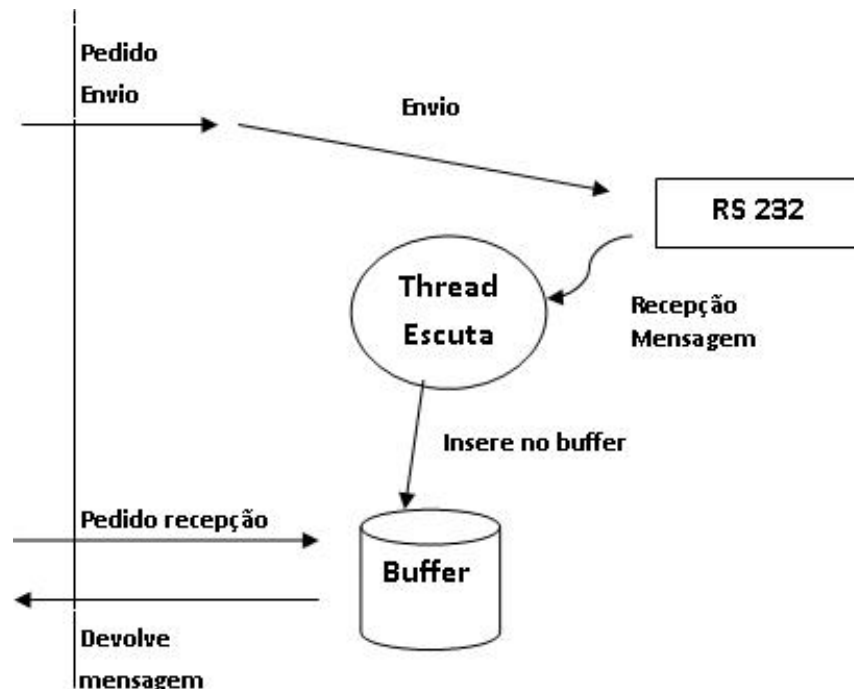


Figura 28 – Objecto Comunicação EIB/KNX

O primeiro método público disponibilizado por esta classe de comunicação é o método “open()”. Este método configura o funcionamento da porta se série, estabelecendo os valores de variáveis de “timeouts”, paridade, “stop-bits”, etc. Depois de inicializada a porta COM, este método troca algumas mensagens de inicialização do BCU (Bus Coupling Unit), configurando-o para a comunicação com os dispositivos existentes na rede. Este método inicializa ainda a tarefa de recepção de mensagens provenientes da rede de dispositivos.

O método público “Write()” permite o envio de mensagens no formato EIB para a rede de dispositivos. Este método recebe como argumentos, uma mensagem EIB e o respectivo “message code”, tendo este, em mensagens de grupo, o valor “0x11”. Este método constrói posteriormente o octeto relativo ao tamanho da mensagem e respectiva paridade.

O passo seguinte deste método consiste no envio do octeto de tamanho para o BCU. Este processo consiste, como foi anteriormente referido, num “software handshake” necessário para o envio de qualquer mensagem. Após a confirmação do sucesso deste “software handshake”, será então enviada a mensagem, sendo entre cada byte, respeitado o “Hardware handshake” característico do protocolo PEI Type 16. Em caso de falha no “Software handshake”, a aplicação aguarda um curto espaço de tempo e tenta o seu reenvio.

O método público “readBufMsg()” retorna uma mensagem recebida pela tarefa de recepção de mensagens. A tarefa de recepção de mensagens recebe todas as mensagens provenientes dos dispositivos, colocando-as num buffer FIFO. Este método é então responsável por retirar do buffer a próxima mensagem, devolvendo-a para a aplicação. No caso de este buffer se

encontrar vazio, o método ficará bloqueado até a recepção de uma nova mensagem por parte da tarefa responsável por tal. O acesso a este buffer é controlado por semáforos, implementando um mecanismo semelhante ao conhecido como “produtor-consumidor”. A diferença em relação ao “produtor-consumidor” consiste no facto do processo consumidor (neste caso a tarefa de recepção de mensagens), não ficar bloqueado quando o buffer estiver cheio, descartando então as mensagens mais antigas nele contidas.

O acesso à porta de série é controlado por um mecanismo de trincos lógicos. Este mecanismo garante a exclusão mútua na comunicação com o BCU, evitando assim conflitos na recepção e envio de mensagens em simultâneo.

O método público “close()” é responsável por terminar a comunicação com a rede de dispositivos. Este método termina a execução da tarefa de recepção de mensagens, e troca de seguida um conjunto de mensagens com o BCU, repondo assim o estado em que este se encontrava antes de iniciada a comunicação. Este método deverá ser invocado aquando do término da aplicação, de forma a não deixar o BCU de ligação à rede de dispositivos num estado inconsistente.

De referir que esta comunicação com a rede de dispositivos EIB/KNX, constituiu a tarefa mais complexa e demorada no desenvolvimento desta solução, não só pela complexidade do protocolo de comunicação utilizado, mas também pela inexistência de documentação detalhada para o efeito.

4. TESTES E RESULTADOS

4.1. Rede de Dispositivos EIB/KNX

Para estudo da tecnologia EIB/KNX e posterior desenvolvimento e teste da solução apresentada nesta tese, foi utilizado um protótipo constituído por vários dispositivos EIB/KNX. Este protótipo, demonstrado na figura seguinte foi montado por um grupo de alunos da cadeira de Edifícios Inteligentes e Domótica.



Figura 29 – Protótipo de Dispositivos EIB/KNX

Este protótipo é constituído pelos seguintes dispositivos do fabricante *hager*:

1. Interruptor quádruplo acoplado ao modulo de 4 entradas TX304
2. Controlador de ambiente TX450
3. Interruptor horário electrónico de 2 canais TX022
4. Módulo de 4 estores TX223
5. Módulo de iluminação de intensidade variável TX210

6. Módulo de iluminação de intensidade variável TX210
7. Módulo de rádio frequência TX350
8. Módulo RS232 TH002
9. Disjuntor
10. Fonte de alimentação TX111

Uma vez que estes dispositivo encontravam-se apenas parcialmente programados, pelos alunos que montaram o protótipo, optou-se por limpar a sua programação e efectuar a sua reprogramação de raiz, não só com o intuito de tornar o protótipo utilizável na presente solução, mas também por motivos de aprendizagem do seu funcionamento. A programação dos dispositivos foi efectuada a partir da aplicação ETS3.

Foram atribuídos os seguintes endereços físicos aos dispositivos:

Dispositivo	Endereço Físico
1	1.1.1
2	1.1.2
3	1.1.3
4	1.1.4
5	1.1.5
6	1.1.6

Tabela 1 – Endereços Físicos

Após a programação dos endereços físicos dos dispositivos, de forma a possibilitar a interacção com estes, foi necessário programar o seu funcionamento.

A interacção entre dispositivos sensores e actuadores efectua-se, como foi referido no capítulo que detalha a tecnologia EIB/KNX, através da atribuição de endereços de grupo. Se um dispositivo sensor e um dispositivo actuator subscreverem o mesmo endereço de grupo, ao ser activado o dispositivo sensor este irá enviar a respectiva ordem ao dispositivo actuator.

A atribuição de endereços de grupo a dispositivos actuadores permite também a interacção com este através de uma aplicação externa.

Foram atribuídos os seguintes endereços de grupo aos dispositivos:

Endereço Grupo	Dispositivos	Funcionalidade
0/1/01	1	Entrada 1 – On/Off
	5	On/Off
		Indicação de estado On/Off
0/1/02	5	Valor de intensidade
		Indicação de intensidade
0/1/03	1	Entrada 1 – Variação
	5	Variação Intensidade
0/2/01	1	Entrada 2 – On/Off
	6	On/Off
		Indicação de estado On/Off
0/2/02	6	Valor de intensidade
		Indicação de intensidade
0/2/03	1	Entrada 2 – Variação
	6	Variação Intensidade
0/3/01	1	Entrada 3 – On/Off
	5	On/Off
	6	On/Off
0/3/2	5	Valor de Intensidade
	6	Valor de Intensidade
0/3/3	1	Entrada 3 – Variação
	5	Variação Intensidade
	6	Variação Intensidade
0/4/1	3	Date
0/4/2	3	Time

Tabela 2 – Endereços de Grupo

De forma a facilitar a interpretação da tabela anterior, a lista seguinte descreve as acções permitidas para cada endereço de grupo:

0/1/01 – Permite ligar ou desligar o dispositivo 5 (1ª lâmpada), a partir do interruptor 1 do dispositivo 1 ou de uma aplicação externa. A indicação do estado permite reportar a uma aplicação externa alterações ao estado da lâmpada.

0/1/02 – Permite ler ou escrever o valor da intensidade do dispositivo 5 (1ª lâmpada) a partir de uma aplicação externa. A indicação do valor de intensidade permite reportar a uma aplicação externa sempre alterações ao valor da intensidade da lâmpada.

0/1/03 – Permite variar a intensidade do dispositivo 5 (1ª lâmpada) a partir do interruptor 1 do dispositivo 1.

0/2/01 – Permite ligar ou desligar o dispositivo 6 (2ª lâmpada), a partir do interruptor 2 do dispositivo 1 ou de uma aplicação externa. A indicação do estado permite reportar a uma aplicação externa alterações ao estado da lâmpada.

0/2/02 – Permite ler ou escrever o valor da intensidade do dispositivo 6 (2ª lâmpada) a partir de uma aplicação externa. A indicação do valor de intensidade permite reportar a uma aplicação externa sempre alterações ao valor da intensidade da lâmpada.

0/2/03 – Permite variar a intensidade do dispositivo 6 (2ª lâmpada) a partir do interruptor 2 do dispositivo 1.

0/3/01 – Permite ligar ou desligar os dispositivos 5 e 6 (lâmpadas) em simultâneo, a partir do interruptor 3 do dispositivo 1 ou de uma aplicação externa.

0/3/02 – Permite ler ou escrever o valor da intensidade dos dispositivos 5 e 6 (lâmpadas) em simultâneo, a partir de uma aplicação externa.

0/3/03 – Permite variar a intensidade dos dispositivos 5 e 6 (lâmpadas) em simultâneo, a partir do interruptor 3 do dispositivo 1.

0/4/01 – Permite a leitura ou escrita do valor da data do dispositivo 3 (temporizador), apenas a partir de uma aplicação externa.

0/4/02 – Permite a leitura ou escrita do valor do tempo (hh:mm:ss) do dispositivo 3 (temporizador), apenas a partir de uma aplicação externa.

4.2. Aplicação de Monitorização e Controlo DomoBus

Sendo a tecnologia DomoBus proveniente de um estudo académico em desenvolvimento, ainda não existe implementada nenhuma aplicação de monitorização e controlo destinada ao utilizador final. Como referido anteriormente, estas aplicações pretendem-se versáteis e adaptáveis, sendo configuradas a partir de ficheiros XML.

Actualmente, para efeitos de teste existe uma aplicação mais simples, implementada por alunos da cadeira de Edifícios Inteligentes e Domótica, de seu nome “DomoDevSimObserver”, que permite a troca de mensagens com dispositivos DomoBus. Esta foi a aplicação utilizada para testar esta solução.

A figura seguinte mostra o ecrã destinado ao envio de mensagens para dispositivos DomoBus:

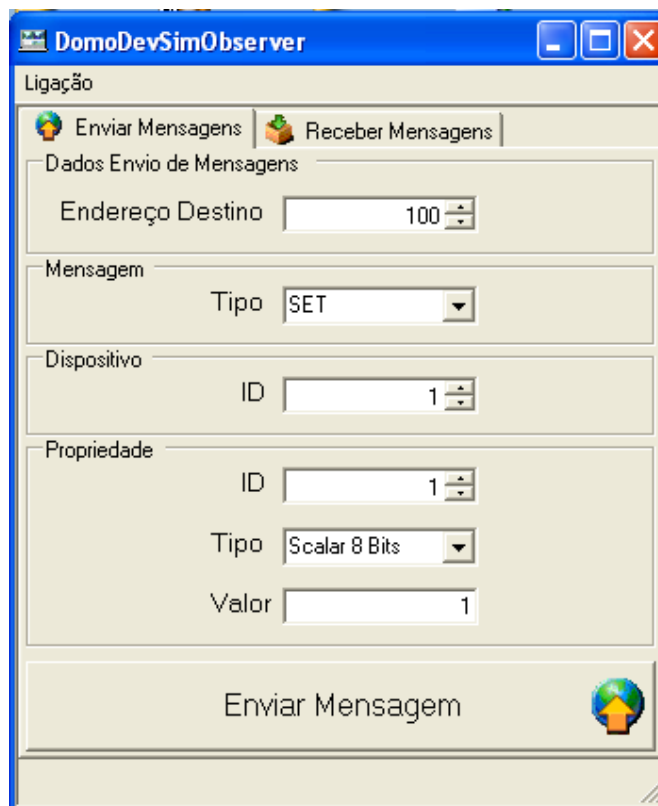


Figura 30 – DomoDevSimObserver: ecrã envio de mensagens

Existem quatro áreas principais no envio de mensagens:

1. Endereço Destino – Endereço da aplicação que controla o dispositivo.
2. Tipo da Mensagem. Existem dois tipos de mensagens:
 - GET – Permite obter informação do valor de uma propriedade.
 - SET – Permite definir o valor de uma propriedade.
3. Dispositivo – ID do Dispositivo cuja propriedade se pretende alterar/obter informação.
4. Propriedade – Definição da Propriedade que se pretende alterar/obter informação. É necessário definir o seu ID, o tipo de dados (Scalar 8bits, Scalar 16bits, Enumerator ou Array) e, caso o tipo de mensagem seja SET, o novo valor da propriedade. Este campo não precisa de estar preenchido caso o tipo de mensagem seja GET.

Para enviar a mensagem clica-se no enviar mensagem.

A figura seguinte mostra o ecrã destinado à recepção de mensagens provenientes de dispositivos DomoBus:

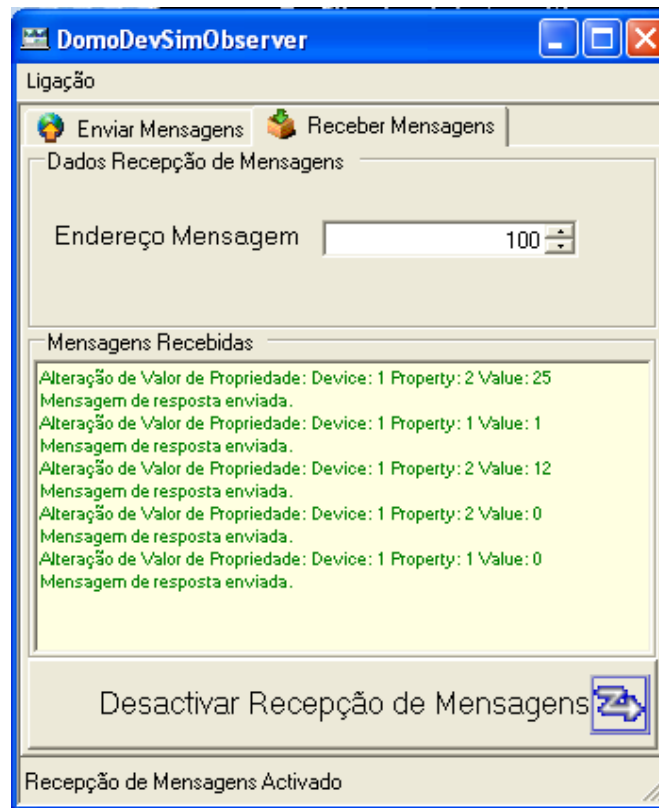


Figura 31 – DomoDevSimObserver: ecrã recepção de mensagens

Para receber mensagens de dispositivos é necessário activar esse modo. Para isso clica-se no botão “Activar Recepção de Mensagens”.

Existem duas áreas principais neste modo:

1. Endereço da Mensagem – Endereço da aplicação que enviou a mensagem. Este endereço é actualizado automaticamente, não é definido pelo utilizador.
2. Mensagens Recebidas – Caixa de texto com todas as mensagens enviadas pelos dispositivos que foram recebidas pelo DomoDevSimObserver. Contém também a informação das mensagens de GET e SET efectuadas por esta aplicação. A imagem apresentada contém alguns exemplos deste tipo de mensagens.

4.3. Procedimentos de teste

4.3.1. Configuração do Adaptador EIB

Como foi referido na descrição da solução, o funcionamento do adaptador EIB é configurado a partir de dois ficheiros XML.

Para os testes efectuados, o conteúdo do ficheiro “Config.xml” foi o seguinte:

```

<ConfigFile>
    <Port Name="COM1"/>
    <ReportTo Address="0x100100"/>
</ConfigFile>

```

Este ficheiro define que será utilizada a porta COM1 para comunicação com os dispositivos, e que o endereço da aplicação para o qual deverão ser reportadas as alterações aos dispositivos será "0x100100".

O conteúdo do ficheiro "DeviceMapping.XML" foi o seguinte:

```

<DomoBus-KNX_mapping>
    <DeviceMappingList>
        <DeviceMapping DomoBusAddress="0x00010001" DeviceTypeID="1"
KNXPhysicalAddress="1105">
            <Property ID="1" KNXGroupAddress="0101"/>
            <Property ID="2" KNXGroupAddress="0102"/>
        </DeviceMapping>
        <DeviceMapping DomoBusAddress="0x00010002" DeviceTypeID="1"
KNXPhysicalAddress="1106">
            <Property ID="1" KNXGroupAddress="0201"/>
            <Property ID="2" KNXGroupAddress="0202"/>
        </DeviceMapping>
        <DeviceMapping DomoBusAddress="0x00010003"
DeviceTypeID="1">
            <Property ID="1" KNXGroupAddress="0301"/>
            <Property ID="2" KNXGroupAddress="0302"/>
        </DeviceMapping>
        <DeviceMapping DomoBusAddress="0x00020001" DeviceTypeID="2"
KNXPhysicalAddress="1103">
            <Property ID="1" KNXGroupAddress="0401"/>
            <Property ID="2" KNXGroupAddress="0402"/>
        </DeviceMapping>
    </DeviceMappingList>
    <PropertyMappingList>

```

```

        <PropertyMapping DeviceTypeID="1" PropertyID="1"
KNXFunctionType="EIS1"/>
        <PropertyMapping DeviceTypeID="1" PropertyID="2"
KNXFunctionType="EIS6"/>
        <PropertyMapping DeviceTypeID="2" PropertyID="1"
KNXFunctionType="EIS4"/>
        <PropertyMapping DeviceTypeID="2" PropertyID="2"
KNXFunctionType="EIS3"/>
    </PropertyMappingList>
</DomoBus-KNX_mapping>

```

Este ficheiro XML representa os quatro dispositivos que serão conhecidos pelo DomoBus, com os respectivos endereços físicos e endereços de grupo KNX. O terceiro dispositivo representa, do ponto de vista do DomoBus, um dispositivos virtual, uma vez que é na realidade um grupo composto pelos dois módulos de iluminação do KNX.

Na segunda parte deste ficheiro XML, podemos observar a existência de quatro tipos diferentes de funcionalidades EIB/KNX. Embora os módulos de iluminação possuam a função “EIS2 – Dimming”, esta é composta por duas subfunções que na realidade possuem o mesmo funcionamento que a função “EIS1 – Switching” e a função “EIS6 – Scaling”, tendo-se então optado por utilizar antes estas duas designações.

Os dispositivos conhecidos pelas aplicações de monitorização e controlo DomoBus seriam então os seguintes:

- Lâmpada 1
 - o Propriedade 1 – estado ligado/desligado
 - o Propriedade 2 – Intensidade (0-100)
- Lâmpada 2
 - o Propriedade 1 – estado /desligado
 - o Propriedade 2 – Intensidade (0-100)
- Lâmpadas 1 e 2 (dispositivo virtual)
 - o Propriedade 1 – estado /desligado
 - o Propriedade 2 – Intensidade (0-100)
- Relógio
 - o Propriedade 1 – Data
 - o Propriedade 2 – Hora

4.3.2. Configuração do ComServ

Como anteriormente referido, todas as aplicações DomoBus para se poderem registar no ComServ, necessitam de ter o seu endereço previamente configurado neste. Esta registo garante que aplicações DomoBus não autorizadas não se conseguiram registar no DomoBus.

O ficheiro utilizado para configuração do ComServ possuía o seguinte conteúdo:

:0x000100 1 65535 172.20.81.132

:0x000200 1 65535 172.20.81.132

:0x100100 1 65535 172.20.81.132

Segundo este ficheiro, o ComServ irá aceitar o registo por parte de três aplicações DomoBus.

O endereço “0x100100” irá ser o endereço utilizado pelo “DomoDevSimObserver” que irá representar a aplicação de monitorização e controlo. O valor “1” indica que o canal de comunicação a utilizar será por IP. O valor 0 indicaria que a comunicação iria ser feita através da porta de série. Os valores “65535” e “172.20.81.132” indicam o porto do socket associado a esta aplicação e o seu endereço IP respectivamente.

Os endereços “0x000100” e “0x000200” serão os endereços pelos quais o adaptador EIB irá estar responsável. Foram escolhidos dois endereços diferentes de forma a testar esta capacidade do adaptador EIB.

4.3.3. Resultados

Após a configuração dos vários componentes integrantes desta solução, foi executado o serviço de comunicações ComServ, a aplicação de monitorização e controlo DomoDevSimObserver e o adaptador EIB.

Seguidamente serão apresentados alguns testes tipo executados ao protótipo desta solução. Estes testes abrangem notificações à aplicação externa de alteração de estados dos dispositivos, tal como a interacção com os dispositivos, quer para escrita e leitura dos valores das suas propriedades.

- **Teste 1 – Notificações**

Este teste teve como finalidade testar a capacidade de receber na aplicação de monitorização e controlo DomoBus, report's relativos a mudanças de estado ligado/desligado de um dispositivo EIB/KNX (Mensagens do tipo REPORT). Para tal efectuaram-se os seguintes passos.

1. Carregou-se no interruptor 1 do dispositivo 1, de forma a acender a primeira lâmpada.

2. Carregou-se no interruptor 1 do dispositivo 1 durante cerca de 2 segundos, de forma a variar a intensidade da primeira lâmpada.
3. Carregou-se no interruptor 1 do dispositivo 1, de forma a desligar a primeira lâmpada.
4. A figura seguinte mostra as mensagens observadas pela aplicação DomoDevSimObserver:

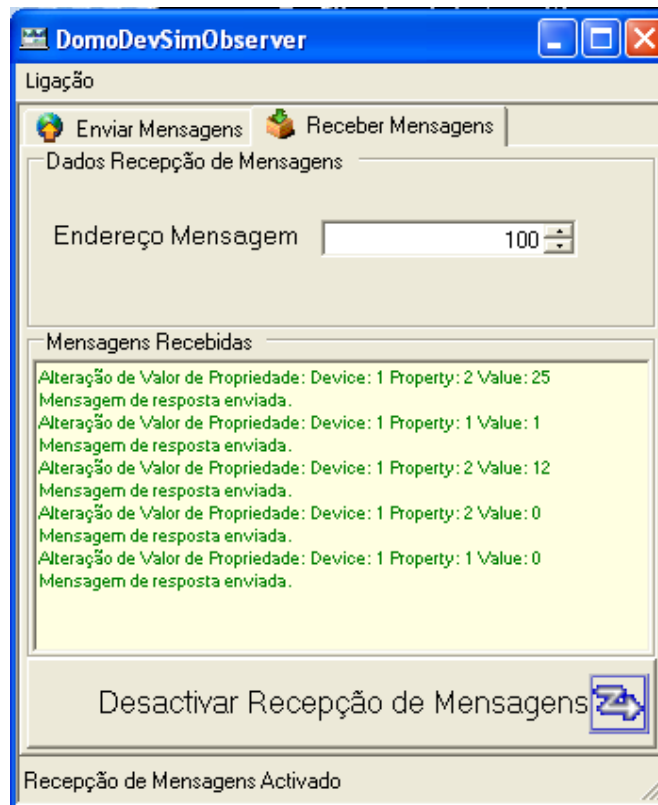


Figura 32 – DomoDevSimObserver: mensagens de *report*

Podemos observar que quando se ligou a lâmpada, a aplicação de monitorização recebeu informação proveniente do endereço “100” de que a propriedade com o ID 2 (propriedade relativa à intensidade) possui o valor 25 e a propriedade com o ID 1 (propriedade relativa ao estado ligado/desligado) do dispositivo com o ID 1 possui o valor 1. Estas duas mensagens são enviadas pelo dispositivo em simultâneo, portanto a ordem pela qual são recebidas pela aplicação é irrelevante e aleatória.

Seguidamente ao variar a intensidade da lâmpada, foi recebida informação de que a propriedade relativa a tal possuía o valor 12.

Ao desligar a lâmpada, foi recebida a informação que ambas as propriedades deste dispositivo possuem o valor 0.

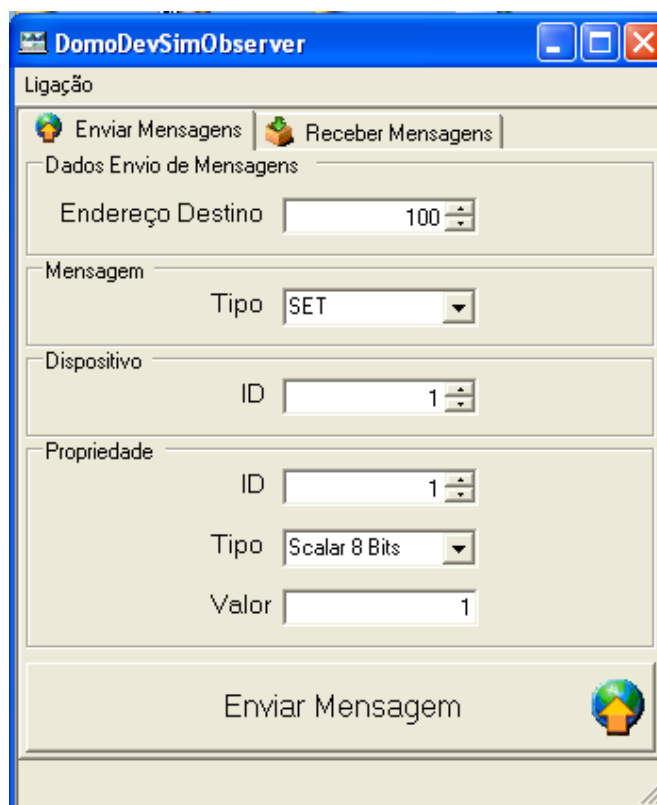
Seguidamente foi efectuado o mesmo teste para a segunda lâmpada. Os resultados foram idênticos aos observados anteriormente para a primeira lâmpada.

Quanto ao report de informação por parte de grupos de dispositivos, o seu funcionamento não é ideal. Ao interagir com o grupo constituído pelas duas lâmpadas, através do interruptor 3 do dispositivo 1, cada lâmpada reporta o seu estado separadamente. Uma vez que as aplicações de monitorização e controlo não possuem a noção de que um dispositivo virtual é composto por dois ou mais dispositivos físicos, esta recebe o report independente de cada dispositivo mas não o interpreta como report de um dispositivo virtual.

- **Teste 2 – Alteração de Propriedades**

Este teste teve como finalidade testar a capacidade de ligar e desligar um dispositivo EIB/KNX através da aplicação de monitorização e controlo DomoBus (Mensagens do tipo SET).

1. No ecrã de envio de mensagens da aplicação DomoDevSimObserver, preencheram-se os campos relativos à funcionalidade ligar da primeira lâmpada, como apresentado na figura seguinte:



The screenshot shows the 'DomoDevSimObserver' application window. It has a title bar with standard Windows controls. Below the title bar is a 'Ligação' (Connection) section with two tabs: 'Enviar Mensagens' (Send Messages) and 'Receber Mensagens' (Receive Messages). The 'Enviar Mensagens' tab is active. Under this tab, there are four sections: 'Dados Envio de Mensagens' (Send Message Data) with an 'Endereço Destino' (Destination Address) field set to 100; 'Mensagem' (Message) with a 'Tipo' (Type) dropdown set to 'SET'; 'Dispositivo' (Device) with an 'ID' field set to 1; and 'Propriedade' (Property) with an 'ID' field set to 1, a 'Tipo' (Type) dropdown set to 'Scalar 8 Bits', and a 'Valor' (Value) field set to 1. At the bottom of the form is a large 'Enviar Mensagem' (Send Message) button with a globe icon.

Figura 33 – DomoDevSimObserver: Ligar lâmpada 1

2. Após o dispositivo se ter ligado, foi enviada uma para o mesmo dispositivo, mas definindo o valor 50 para a propriedade 2.
3. Seguidamente enviou-se a mensagem relativa ao desligar da primeira lâmpada.
4. Repetiram-se os três passos anteriores para a segunda lâmpada, e seguidamente para o dispositivo virtual relativo ao grupo constituído pelas duas lâmpadas.
5. Os resultados apresentaram-se iguais aos esperado como se pode constatar pela seguinte imagem relativa ao ecrã de recepção de mensagem da aplicação DomoDevSimObserver.

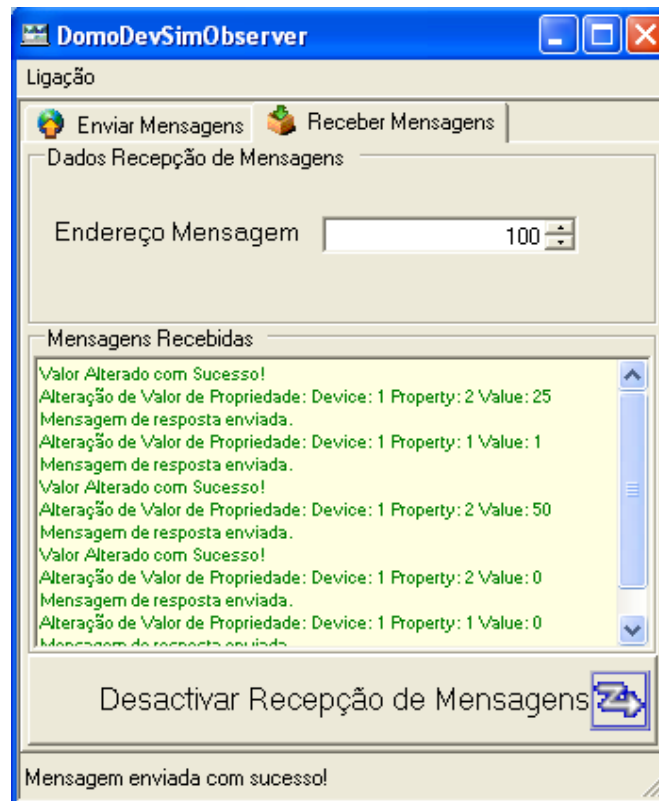


Figura 34 – DomoDevSimObserver: Respostas aos comandos enviados

Podemos observar as alterações aos estados das propriedades 2 e 1 do dispositivo 1 (primeira lâmpada), aquando do envio da mensagem respectiva à funcionalidade “ligar” para este dispositivo.

Podemos também observar a alteração ao estado da propriedade 2 (intensidade) do dispositivo 1 para o valor 50, aquando do envio da mensagem relativa a tal.

Por fim, ao enviar a mensagem relativa á funcionalidade “desligar” da primeira lâmpada, podemos observar a recepção das duas mensagens relativas à alteração das propriedades ligar/desligar e intensidade deste dispositivo para o valor 0.

Dado o tamanho reduzido da janela não é possível ver na imagem as mensagens relativas à segunda lâmpada nem ao dispositivo virtual constituído pelas duas lâmpadas. No entanto os resultados para a segunda lâmpada foram iguais.

Em relação ao grupo virtual, a aplicação foi notificada de que as mensagens foram enviadas com sucesso, mas cada lâmpada reporta as alterações ao seu estado individualmente.

- **Teste 3 – Leitura de Propriedades**

Este teste tem como finalidade testar a capacidade de inquirir os dispositivos relativamente ao seu estado (Mensagens do tipo GET)

1. Com todos os dispositivos desligados, no ecrã de envio de mensagens da aplicação DomoDevSimObserver, preencheram-se os campos relativos à funcionalidade de inquirir a primeira lâmpada quanto ao seu estado ligado/desligado (propriedade 1), como apresentado na figura seguinte:

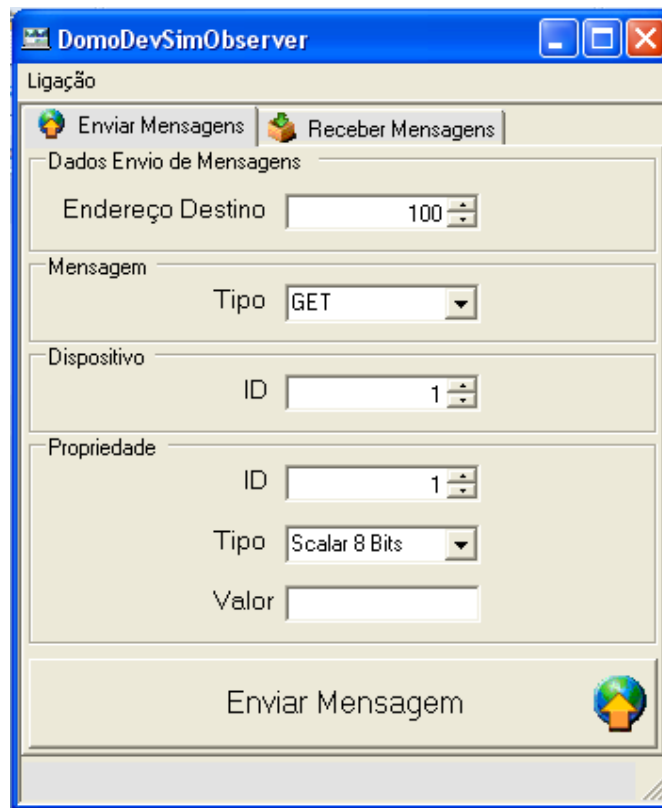


Figura 35 – DomoDevSimObserver: Mensagem GET

2. Seguidamente enviou-se uma mensagem idêntica mas relativa à propriedade 2 do mesmo dispositivo.
3. Após recepção das respostas, ligou-se o dispositivo 5 (primeira lâmpada) através do interruptor do dispositivo 1.

4. Já com o dispositivo 5 (primeira lâmpada) activado repetiu-se o envio das mensagens GET. As respostas enviadas pelos dispositivos no decorrer deste teste podem ser vistas na imagem seguinte

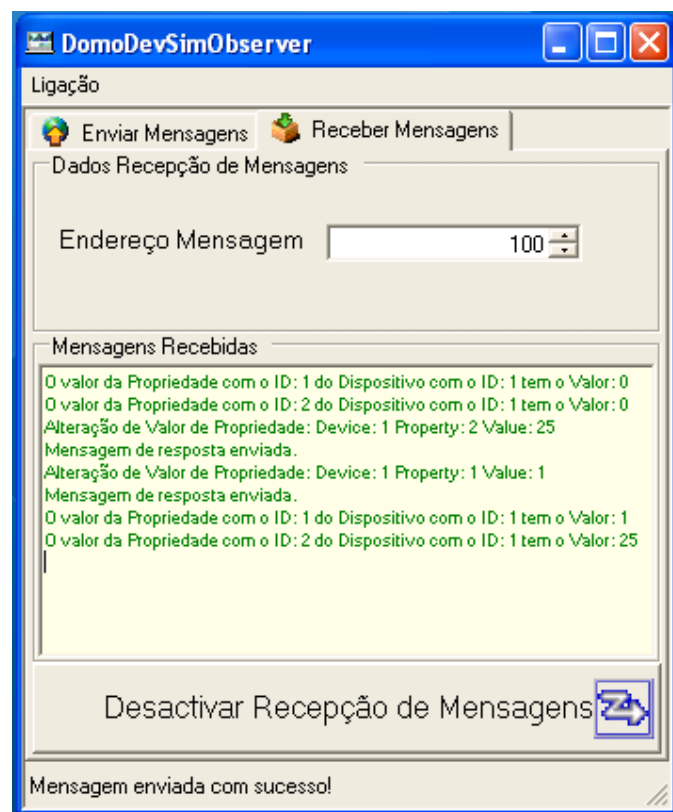


Figura 36 – DomoDevSimObserver: Respostas mensagens GET

Pode-se observar nas duas primeiras linhas da imagem as respostas às duas primeiras mensagens GET enviadas. Como o dispositivo se encontrava desligado ambas as propriedades devolveram o valor 0. Quando foi ligado o dispositivo podemos observar as mensagens de *report* enviadas por este quando o seu estado foi modificado. Nas duas últimas linhas podemos ver as respostas às últimas mensagens GET enviadas. Aqui podemos verificar que o dispositivo já se encontrava ligado (propriedade 1 com o valor 1) e que a sua intensidade era de 25% (propriedade 2 com o valor 25).

Repetiu-se o processo para o dispositivo 6 (segunda lâmpada) e obtiveram-se resultados idênticos. No caso do dispositivo virtual constituído pelas duas lâmpadas, como é comportamento normal nos grupos de dispositivos EIB/KNX, apenas um dos dispositivos respondeu.

5. CONCLUSÃO

Como referido nesta dissertação de tese, a domótica é uma ciência que se tem vindo a expandir ao longo dos últimos anos a um ritmo mais lento do que seria de esperar à partida. Este ritmo reduzido deve-se em grande parte à dispersão e confusão existente no mercado devido à existência de múltiplas tecnologias incompatíveis entre si. Outro das principais causas para este ritmo lento de expansão é a inexistência de aplicações de monitorização e controlo versáteis e adaptáveis, tornando as soluções existentes actualmente insatisfatórias para os seus utilizadores a longo prazo.

A solução apresentada nesta tese cria assim a possibilidade de desenvolvimento de sistemas baseados em diferentes tecnologias, até agora inexistentes. Ao mesmo tempo, esta solução permite também a possibilidade de criação de aplicações versáteis para a utilização com diferentes tecnologias domóticas existentes no mercado, que podem ser configuradas perante qualquer alteração à rede de dispositivos de uma habitação ou alterações ao comportamento de dispositivos existentes. Estas aplicações criam valor para os seus utilizadores, na medida em estas alterações á rede de dispositivos, e consequentemente à aplicação não implicam a necessidade de aquisição de uma nova versão aplicação reduzindo drasticamente os seus custos de manutenção.

Apesar da solução aqui apresentada, propor a possibilidade de utilização de múltiplas tecnologias domóticas numa rede integrada, para o desenvolvimento de um protótipo optou-se pela tecnologia EIB/KNX. A escolha desta tecnologia deveu-se ao facto de ser actualmente a tecnologia dominante no mercado, sendo também a tecnologia mais completa e versátil, permitindo assim uma utilização mais abrangente.

A tecnologia escolhida para a interacção com o utilizador foi o DomoBus. Esta escolha deveu-se ao facto de esta ser uma tecnologia simples, versátil e extremamente poderosa. É com base nesta tecnologia que o desenvolvimento desta solução se tornou possível. O seu modelo de abstracção permite virtualmente a representação de dispositivos de qualquer tecnologia, criando assim a possibilidade de desenvolvimento de aplicações de monitorização e controlo versáteis tecnologicamente independentes.

O sistema de comunicações da tecnologia DomoBus desenhado inicialmente com o intuito de permitir a comunicação entre diferentes aplicações e redes de dispositivos domóticos DomoBus, cria também, através de adaptadores ou gateways como o apresentado nesta solução, a possibilidade de integração de redes de dispositivos de outras tecnologias existentes no mercado.

A tecnologia DomoBus, desenvolvida como um projecto académico, tendo sido utilizada nos últimos anos como ferramenta de aprendizagem e de teste para novas ideias, embora ainda

em fase de desenvolvimento, mostrou-se uma tecnologia bastante versátil e inovadora. Dada a complexidade da tecnologia EIB/KNX, ao constituir o principal pilar de suporte à solução apresentada nesta dissertação, o DomoBus demonstra uma enorme flexibilidade e vem preencher algumas lacunas das tecnologias actualmente existentes no mercado.

Este protótipo foi desenvolvido com base na utilização de um conjunto limitado de dispositivos, que foi apresentado no capítulo anterior. Estes dispositivos utilizados apresentam as funcionalidades mais comuns em qualquer tipo de dispositivos, o que implica que o protótipo desenvolvido permita a interacção com a maioria das funcionalidades dos dispositivos existentes no mercado. No entanto existem outras funcionalidades não suportadas pelos dispositivos utilizados tendo sido impossível explorá-las neste protótipo sem interacção directa. Como trabalho futuro fica então a expansão deste protótipo a outras funcionalidades da tecnologia EIB/KNX que não foram possíveis de implementar nesta solução.

Esta solução introduz ainda a possibilidade de criação de um sistema multi tecnológico. O protótipo desenvolvido permite a utilização de sistemas compostos pela tecnologia DomoBus e EIB/KNX, tendo este sido um importante passo, dada a forte utilização desta segunda tecnologia no mercado actualmente. Como trabalho futuro, fica ainda a criação de adaptadores ou gateways, como o apresentado nesta solução, que permitam a interacção da presente solução com redes de dispositivos de outras tecnologias.

Por fim, este protótipo apresenta ainda algumas deficiências ao nível do protocolo de comunicação com o sistema EIB/KNX. Estas deficiências devem-se à inexistência de documentação detalhada sobre as mensagens EIB/KNX, que levam à incapacidade de tradução e compreensão de algumas mensagens provenientes da rede de dispositivos. Espera-se que, devido à expansão dos sistemas domóticos, seja publicada num futuro próximo, documentação que permitirá corrigir as deficiências existentes neste protótipo.

6. BIBLIOGRAFIA

- [1] The EIB Association: The EIBA Handbook Series – Release 3.0, Brussels, 1998
- [2] The Konnex Association, <http://www.konnex.org>
- [3] Renato Nunes, "Modelo de Especificação e Programação de um Sistema Domótico", IADIS Conferência Ibero-Americana WWW/Internet 2004, Madrid, Spain, October 2004, pp. 377-384.
- [4] Renato Nunes, "DomoBus - A New Approach to Home Automation", 8CLEEE - 8th International Congress on Electrical Engineering, Portugal, July 2003, pp.2.19-2.24.
- [5] The Konnex Association: KNX, The World's First Open Standard For Home and Building Control – System Architecture, Brussels, July 2004.
- [6] The Konnex Association: KNX, The World's First Open Standard for Home and Building Control – Interworking, Brussels, July 2004.
- [7] Wolfgang Kastner and Bernd Thallner: "A GPL Linux Device Driver for the EIB", Technische Universität Wien Institut für Rechnergestützte Automation, Treitlstraße 1, A-1040 Wien
- [8] Marc Goossens – The EIB Association: The Fast Lane to EIB - The EIB System for Home & Building Electronics, Brussels, Belgium, February 1998
- [9] Renato Nunes, 2004, "A Communication Infrastructure for Home Automation". CCCT 2004 - International Conference on Computer, Communications and Control Technologies, Austin, USA.
- [10] Renato Nunes, "Sistema DomoBus - Especificação XML de um Sistema DomoBus", V1.5 - 22/11/2006, Instituto Superior Técnico / Inesc-Id
- [11] Wolfgang Kastner, Georg Neugschwandtner, Martin Kögler, "An Open Approach to EIB/KNX Software Development", TU Wien, Institute of Automation, Automation Systems Group, Treitlstraße 1, A-1040 Vienna, Austria.
- [12] Vienna TU Wien, Institute of Automation, Automation Systems Group: <http://www.auto.tuwien.ac.at/knx/>
- [13] University of Applied Sciences Regensburg, EIB Team: <http://homepages.fh-regensburg.de/~scg39398/eibteam/eibteam.htm>
- [14] Renato Nunes, "DomoBus Comm-API: Comunicação no sistema DomoBus – Application Programming Interface", V1.2, 19 Outubro, 2005
- [15] <http://www.acasainteligente.com/index.asp>
- [16] <http://www.smarthomeusa.com/info/x10theory/#theory>
- [17] Kenneth P. Wacks, Ph.D. "Home Automation and Utility Customer Services", Cutter Information Corporation