

FEUP KNX

Domótica KNX/EIB de Baixo Custo

2007 / 2008

**Departamento de Engenharia Electrotécnica e
de Computadores**

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Faculdade de Engenharia da Universidade do Porto

Diana Sobreiro da Costa Palma – ee05241

<http://www.feupknx.pt.vu>



Faculdade de Engenharia da Universidade do Porto

FEUP KNX
Domótica KNX/EIB de Baixo Custo

Diana Sobreiro da Costa Palma

Dissertação realizada no âmbito do
Mestrado Integrado em Engenharia Electrotécnica e de
Computadores

Dissertação realizada sob a orientação do

Professor Doutor Mário de Sousa,

do Departamento de Engenharia Electrotécnica e de Computadores

da Faculdade de Engenharia da Universidade do Porto

(O Presidente do Júri, Professor Doutor Américo Lopes de Azevedo)

Porto, Março 2008

Resumo

A implantação de um sistema de automação numa casa ou num edifício, permite aumentar o conforto, segurança e obter uma melhor gestão de gastos de energia. Por exemplo, uma luz só liga se for necessário e não ocorre o problema de esquecimento de luzes ligadas. Hoje em dia, os sistemas utilizados para automatizar um edifício têm uma hierarquia distribuída, ao contrário dos sistemas de controlo industriais cuja lógica de controlo se encontra centralizada nos autómatos programáveis. Um sistema de automação de um edifício assenta numa rede de comunicação de dados sobre a qual também é possível realizar tarefas de gestão da mesma rede.

Para interligar todos os dispositivos de uma casa é necessário utilizar uma rede com alguma complexidade. Um dos protocolos de rede mais utilizados em sistemas de automação de casas ou edifícios é o *European Installation Bus* (EIB/KNX). Utiliza o meio físico proprietário Twisted Pair (TP), mas este protocolo permite a utilização do mesmo sob RS-232, USB, Rádio – Frequência (RF) e *Ethernet*. Com o aumento do uso do protocolo de rede TCP/IP sob *Ethernet* em edifícios, é grande a vantagem de utilizar o *backbone* da rede TCP/IP para ligar a rede KNX/EIB e assim aproveitar a cablagem já existente. Com esta solução, poder controlar uma casa remotamente utilizando a Internet, deixa de ser uma utopia.

O objectivo deste trabalho é tornar o projecto Domoitech realizado em KNX/EIB, compatível com redes KNX/EIB sob TCP/IP *Ethernet*. Serão estudadas possíveis arquitecturas e será criado um servidor que efectua a interface entre uma rede TCP/IP *Ethernet* e a rede KNX/EIB implementada no Domoitech. A norma KNX/EIB, inclui uma parte que especifica como este servidor deve ser criado. Essa parte tem o nome de KNXnet/IP e é um protocolo que permite o envio de tramas KNX/EIB sob o TCP/IP.

O servidor KNXnet/IP a ser criado, tem a vantagem de ser multi-plataforma e a aplicação do mesmo em sistemas embebidos de baixo custo, torna a solução muito económica. Serão efectuados testes e validação do servidor KNXnet/IP depois da implementação do mesmo.

Abstract

The deployment of a system of automation in a house or a building can increase the comfort, safety and better management of spending power. For example, a light turn on only if necessary and is not the problem of forgetfulness of lights on. Today, the systems used to automate a building have a hierarchy distributed, as opposed to industrial control systems whose logic of control is centralized in a programmable logic controller (PLC). A system automation of a building based on a data communication network, on which it is also possible to perform management tasks on the same network.

To connect all devices of a house is necessary to use a network with some complexity. One of the Network Protocols used in most systems automation of houses or buildings, is the European Installation Bus (KNX/EIB). Uses the media owner Twisted Pair (TP), but the network protocol allows the use of even under RS-232, USB, Ethernet and RF. With the increase in the use of the protocol TCP/IP on Ethernet in buildings, it is a great advantage of using the backbone of the TCP/IP network to connect the network KNX/EIB and thus takes up the existing wiring. With this solution, able to manage a home remotely using the Internet, ceases to be a dream.

The objective of this work is to make the project Domoitech held in KNX/EIB, compatible with networks KNX/EIB under TCP/IP Ethernet. Will be studied possible architectures and will be set up a server that performs the interface between a TCP/IP Ethernet network and KNX/EIB implemented in Domoitech. The standard KNX/EIB includes a section that specifies how this server must be created. This piece is named KNXnet/IP and is a protocol that allows the transmission of frames KNX/EIB under the TCP/IP.

The server KNXnet/IP to be created, has the advantage of being multi-platform and the application of it in embedded systems at low cost, makes the solution very economic. After the implementation of the server KNXnet / IP, tests and validation will be carried out.

Prefácio

Os sistemas de automação de edifícios, mais conhecidos por sistemas Domóticos, têm um custo elevado. Até o sistema mais simples que utiliza um protocolo de rede elementar como o X-10, tem um custo que não é acessível a todos os bolsos. Por exemplo, um simples interruptor de parede que funciona em X-10, tem um custo superior a 40€. Quando comparado com o KNX/EIB, o X-10 é o protocolo “pobre” para Domótica e se um simples interruptor X-10 é caro, um interruptor KNX/EIB é exageradamente caro (custo superior a 70€).

A motivação para a realização desta dissertação surgiu na ideia de dar seguimento ao projecto Domoitech que consistiu em criar um sistema Domótico de baixo custo e compatível com outros equipamentos que existam no mercado. O protocolo de Domótica utilizado no Domoitech foi o KNX/EIB.

Ao longo da realização deste trabalho, o contacto com o mercado foi mantido sempre, para uma constante actualização sobre as novidades e acompanhar o crescimento e necessidades dos sistemas domóticos no mercado nacional.

Em primeiro lugar gostaria de agradecer à minha família, especialmente aos meus pais, por terem estado sempre presentes e me terem apoiado ao longo deste percurso escolar.

Ao orientador, professor Mário de Sousa agradeço a sua disponibilidade e orientação ao longo desta dissertação.

Aos meus amigos e colegas que me apoiaram neste percurso universitário e que partilharam comigo as vivências mais importantes e estiveram sempre ao meu lado.



Índice

1	<i>Introdução</i>	13
1.1	<i>Protocolos de Domótica</i>	14
1.2	<i>Domoitech</i>	16
1.3	<i>Objectivos</i>	18
2	<i>Protocolo KNX/EIB</i>	19
2.1	<i>KNX/EIB Handbook</i>	22
2.2	<i>Arquitectura</i>	24
2.2.1	<i>Modos de configuração</i>	26
2.2.2	<i>Topologia</i>	28
2.2.3	<i>Formato da trama KNX/EIB</i>	30
2.3	<i>Meio Físico</i>	31
2.3.1	<i>Par entrançado</i>	31
2.3.2	<i>Rede eléctrica</i>	33
2.3.3	<i>Rádio Frequência e Infravermelhos</i>	34
2.3.4	<i>Ethernet</i>	35
2.4	<i>Comunicação</i>	36
2.4.1	<i>Camada de Ligação Lógica</i>	36
2.4.2	<i>Camada de Rede</i>	37
2.4.3	<i>Camada de Transporte</i>	38
2.4.4	<i>Camada de Aplicação</i>	38
2.4.4.1	<i>Serviços</i>	39
2.4.4.2	<i>EIS</i>	42
2.5	<i>Interfaces Standard de Comunicação</i>	45



2.5.1	<i>Physical External Interface</i>	45
2.5.1.1	<i>Protocolo FT 1.2</i>	48
2.5.2	<i>External Message Interface</i>	51
2.5.2.1	<i>Mensagens cEMI na camada de ligação lógica</i>	52
2.5.2.2	<i>Mensagens cEMI de gestão e configuração</i>	56
2.6	<i>ETS</i>	59
3	<i>KNXnet/IP</i>	63
3.1	<i>Arquitectura</i>	64
3.2	<i>Módulos do KNXnet/IP</i>	66
3.3	<i>Núcleo</i>	69
3.4	<i>Gestão de dispositivos</i>	71
3.5	<i>Tunnelling</i>	72
3.6	<i>Routing</i>	74
3.7	<i>Resumo dos serviços do KNXnet/IP</i>	75
3.8	<i>Formato das tramas KNXnet/IP</i>	78
4	<i>Análise da arquitectura</i>	79
4.1	<i>Arquitectura Domoitech com KNXnet/IP</i>	80
4.1.1	<i>Projecto de Hardware</i>	80
4.2	<i>Arquitectura Híbrida</i>	85
4.2.1	<i>Projecto de Hardware</i>	86
4.3	<i>Comparação de arquitecturas</i>	87
5	<i>Implementação do Servidor KNXnet/IP</i>	91
5.1	<i>Arquitectura do Software</i>	91
5.2	<i>Validação e Resultados</i>	95
5.2.1	<i>Testes aos serviços do Núcleo</i>	97



5.2.2	<i>Testes aos serviços do Tunneling.....</i>	<i>100</i>
5.2.3	<i>Testes aos serviços da Gestão de Dispositivos.....</i>	<i>102</i>
6	<i>Conclusão.....</i>	<i>103</i>
6.1	<i>Desenvolvimentos futuros.....</i>	<i>104</i>
	<i>Referências.....</i>	<i>106</i>
	<i>Apêndice 1.....</i>	<i>107</i>

Índice de Figuras

<i>Figura 1 - Diagrama de blocos da estrutura de hardware Domoitech.....</i>	<i>16</i>
<i>Figura 2 – Modelo do Sistema KNX.....</i>	<i>22</i>
<i>Figura 3 - Modelo KNX/EIB</i>	<i>24</i>
<i>Figura 4 - Estrutura de uma rede KNX/EIB.....</i>	<i>29</i>
<i>Figura 5 - Formato Trama EIB.....</i>	<i>30</i>
<i>Figura 6 - Sinal balanceado entre dispositivos KNX/EIB.....</i>	<i>31</i>
<i>Figura 7 - Protocolo CSMA/CA.....</i>	<i>32</i>
<i>Figura 8 - Codificação sinal pela rede eléctrica.....</i>	<i>34</i>
<i>Figura 9 – Serviços da camada de ligação lógica</i>	<i>36</i>
<i>Figura 10 – Interface com a camada de rede entre os dispositivos KNX/EIB</i>	<i>37</i>
<i>Figura 11 – Interface com a camada de transporte entre os dispositivos KNX/EIB.....</i>	<i>38</i>
<i>Figura 12 – Códigos dos serviços KNX/EIB modo multicast.....</i>	<i>40</i>
<i>Figura 13 – Exemplo Trama GroupValue_Read.....</i>	<i>40</i>
<i>Figura 14 - Exemplo Trama GroupValue_Res.....</i>	<i>41</i>
<i>Figura 15 – Exemplo Trama GroupValue_Write</i>	<i>41</i>
<i>Figura 16 – EIS Switching.....</i>	<i>42</i>
<i>Figura 17 – EIS Drive Control - Move.....</i>	<i>43</i>
<i>Figura 18 – EIS Drive Control – Step.....</i>	<i>43</i>
<i>Figura 19 – Máquina de estados do Drive Control.....</i>	<i>44</i>
<i>Figura 20 - Diagrama de pinos de um conector PEI</i>	<i>46</i>
<i>Figura 21 - Caractere de controlo de uma trama FT 1.2 de uma estação primária.....</i>	<i>49</i>
<i>Figura 22 - Caractere de controlo de uma trama FT 1.2 de uma estação secundária</i>	<i>50</i>
<i>Figura 23 - Exemplo de transmissão de dados por FT 1.2</i>	<i>51</i>
<i>Figura 24 - Fluxo de mensagens entre cliente e servidor cEMI.....</i>	<i>52</i>
<i>Figura 25 - Estrutura básica de uma mensagem cEMI do tipo L_DATA.....</i>	<i>53</i>
<i>Figura 26 - Control Field 1 do L_DATA.....</i>	<i>54</i>
<i>Figura 27 - Control Field 2 do L_DATA.....</i>	<i>54</i>

<i>Figura 28 - Estrutura básica de mensagem cEMI de gestão e configuração.....</i>	<i>56</i>
<i>Figura 29 - Interface ETS 3.0d Profissional</i>	<i>60</i>
<i>Figura 30 - Interface de pesquisa e inserção de dispositivos KNX/EIB.....</i>	<i>61</i>
<i>Figura 31 - Interface configuração de uma instalação KNX/EIB.....</i>	<i>61</i>
<i>Figura 32 - Modelo em camadas do protocolo KNX e KNXnet/IP</i>	<i>63</i>
<i>Figura 33 - Exemplo de uma configuração KNXnet/IP possível.....</i>	<i>65</i>
<i>Figura 34 - Exemplo de arquitectura com servidor KNXnet/IP implementando tunnelling</i>	<i>67</i>
<i>Figura 35 - Exemplo de arquitectura com servidor servidor KNXnet/IP implementando routing.....</i>	<i>68</i>
<i>Figura 36 - Esquema de um servidor KNXnet/IP e a função do núcleo.....</i>	<i>70</i>
<i>Figura 37 - Endpoints de um servidor KNXnet/IP</i>	<i>71</i>
<i>Figura 38 - Formato das tramas KNXnet/IP.....</i>	<i>78</i>
<i>Figura 39 - Arquitectura Domoitech com KNXnet/IP.....</i>	<i>80</i>
<i>Figura 40 - Esquema do hardware evidenciando as ligações do PIC18F6520 e da XPORT AR.....</i>	<i>82</i>
<i>Figura 41 - Diagrama de blocos da estrutura de hardware</i>	<i>84</i>
<i>Figura 43 - Diagrama de blocos da estrutura de hardware</i>	<i>86</i>
<i>Figura 44 - Gráfico comparativo entre as arquitecturas.....</i>	<i>89</i>
<i>Figura 45 - Arquitectura de software.....</i>	<i>92</i>
<i>Figura 46 - Diagrama UML da estrutura de ficheiros do software</i>	<i>94</i>
<i>Figura 47 - Esquema da instalação da plataforma de testes</i>	<i>96</i>
<i>Figura 48 - Teste Search Request.....</i>	<i>97</i>
<i>Figura 49 - Teste Search Response</i>	<i>98</i>
<i>Figura 50 - Teste Description Response</i>	<i>98</i>
<i>Figura 51- Teste Connect Response</i>	<i>99</i>
<i>Figura 52 - TesteConnection State Response.....</i>	<i>99</i>
<i>Figura 53 - Teste Disconnect Response</i>	<i>100</i>
<i>Figura 54 - Teste Tunneling Request.....</i>	<i>100</i>
<i>Figura 55 - Teste Tunneling Acknowledge.....</i>	<i>101</i>
<i>Figura 56 - Envio de trama KNX/EIB para o Domoitech</i>	<i>101</i>
<i>Figura 57 - Teste Device Configuration Request</i>	<i>102</i>

Figura 58 - Teste Device Configuration Acknowledge..... 102



Índice de Tabelas

<i>Tabela 1 – Protocolos de Rede utilizados na Domótica.....</i>	<i>15</i>
<i>Tabela 2 - Valores de resistência para selecção do tipo de PEI.....</i>	<i>47</i>
<i>Tabela 3 - Custo placa principal Domoitech com KNXnet/IP.....</i>	<i>87</i>
<i>Tabela 4 - Códigos das mensagens cEMI.....</i>	<i>107</i>
<i>Tabela 5 - Propriedades do device object.....</i>	<i>108</i>

Notação e Glossário

Esta secção apresenta os conceitos (glossário de termos) ordenados alfabeticamente e acrónimos utilizados no corpo do texto do relatório.

AVAC	- Aquecimento, Ventilação e Ar Condicionado
Bits/s	- Bits por segundo
cEMI	- Common External Message Interface
Dispositivo KNX/EIB	- Elemento físico que comunica em KNX/EIB e que está ligado a uma rede KNX/EIB. É um objecto concreto que o cliente compra como por exemplo, um interruptor KNX/EIB.
E/S	- Entradas/Saídas
EIS	- EIB Interworking Standards
ETS	- EIB Tool Software
EIBA	- European Installation Bus Association
EIB	- European Installation Bus
IP	- Internet Protocol
KNX/EIB	- Protocolo da associação Konnex
OSI	- Open Systems Interconnection
PDA	- Personnel Digital Assistente, é uma agenda electrónica com sistema operativo e realiza funções idênticas a um computador
PLC	- Autómato Programável
PL	- Power Line
PEI	- Physical External Interface
RF	- Rádio Frequência
TP	- Twisted Pair
Trama	- Sequencia de octetos sempre com a notação little-endian

1 Introdução

A Domótica é uma tecnologia recente que permite a gestão de todos os recursos habitacionais. O termo “Domótica” resulta da junção da palavra “Domus” (casa) com “Telemática” (electrónica + informática). São estes dois últimos elementos que, quando utilizados em conjunto, rentabilizam o sistema, simplificando a vida diária das pessoas satisfazendo as suas necessidades de comunicação, de conforto e segurança. Quando a Domótica surgiu (com o primeiros edifícios, nos anos 80) pretendia-se controlar a iluminação, condições climáticas, a segurança e a interligação entre os 3 elementos. [1]

Nos nossos dias, a ideia base é a mesma, a diferença é o contexto para o qual o sistema está pensado: já não um contexto militar ou industrial mas doméstico. Apesar de ainda ser pouco conhecida e divulgada, pelo conforto e comodidade que pode proporcionar, a Domótica promete ter muitos adeptos.

A Domótica alia as vantagens da electrónica à informática, de forma a obter uma utilização e uma gestão dos diversos equipamentos de uma casa. Torna a vida mais confortável, mais segura e até mais divertida! Permite que as tarefas mais rotineiras e aborrecidas sejam executadas. A Domótica torna-se ainda mais interessante se a aplicarmos para melhorar o nível de vida de uma pessoa incapacitada. Poderá optar por um manuseamento mais ou menos automático. Nos sistemas passivos o elemento reage só quando lhe é transmitida uma ordem, dada directamente pelo utilizador (interruptor). Nos sistemas mais avançados, com mais inteligência, não só interpreta parâmetros, como reage às circunstâncias (informação que é transmitida pelos sensores), por exemplo detectar que uma janela está aberta e avisa o utilizador, ou que a temperatura está a diminuir e ligar o aquecimento.

O controlo remoto de casas de habitação deixa de ser uma utopia. A Domótica permite o acesso às funções vitais da casa, como aquecimento, electrodomésticos, alarme, fechaduras das portas, quer seja através de um comando remoto, da Internet ou do telemóvel.

1.1 Protocolos de Domótica

É grande a variedade de protocolos normalizados direccionados para a Domótica. É necessário diferenciar quais os que são proprietários, os de alianças ou grupos de trabalho e ainda quais são livres e abertos.

No mercado, os sistemas mais usuais utilizam o protocolo X-10 por ser mais económico, mas a grande desvantagem é a sua fraca robustez e ser muito rudimentar. Sistemas que utilizem este protocolo são facilmente encontrados que por utilizarem a rede eléctrica como meio de comunicação, não necessitam de instaladores experientes, nem da instalação de mais cablagem porque a instalação eléctrica existente é aproveitada. É aconselhado o uso deste tipo de sistema no caso de se necessitar um controlo simples.

O protocolo mais fiável e que também é bastante utilizado em sistemas domésticos existentes no mercado, é o KNX/EIB. Este protocolo oferece muita robustez e flexibilidade contudo, os produtos são de elevado custo. Para ter uma noção, um interruptor de pressão quadruplo da Siemens, custa aproximadamente 70,91€. [2]

O protocolo KNX/EIB tem sido desenvolvido dentro do contexto da União Europeia, com o fim de fazer frente aos produtos domóticos já existentes na América e Japão. O protocolo KNX/EIB permite a utilização de vários meios físicos, mas o mais utilizado é o par entrançado onde todos os dispositivos estão ligados a um Bus. Mas com esta solução de par entrançado (TP), era necessária uma fonte de alimentação para o barramento que normalmente tem um custo elevado, na ordem dos 160€. [2]

O KNX/EIB é um protocolo aberto e por esta razão, existem no mercado alguns fabricantes que têm os seus produtos a utilizarem KNX/EIB.

Existem outros protocolos de domótica mas não são muito usuais e raramente encontrados no mercado. A tabela seguinte apresenta uma breve descrição de alguns Protocolos *standards* utilizados na Domótica, distinguindo quais pertencem a organizações e quais são proprietários. [3]

Tabela 1 – Protocolos de Rede utilizados na Domótica

Protocolos de Rede - Alianças e Grupos de Trabalhos		
Standard	Meio Físico	Descrição
BatiBUS	Par Entrançado	Sensores de união e actuadores para construir sistemas que controlem HVAC (Ar Condicionado), sistemas de segurança e acesso. Em convergência com EIB e EHS para KNX.
CEBus (Consumer Electronics Bus)	Todos	O Standard CEBus (EIA-600) é um protocolo criado pela Associação de Industrias Electrónicas (EIA) para ser possível a interligação e comunicação entre dispositivos electrónicos da casa.
EIB (European Installation Bus)	Par Entrançado	Sensores e actuadores para construir sistemas que controlem HVAC (Ar Condicionado), sistemas de segurança e acesso. Em convergência com EHS e BatiBus para KNX.
EHS (European Home System)	Todos	Uma colaboração entre industrias e governos Europeus sobre Domótica. Entre alguma das suas missões a EHSA tem o objectivo de ser um standard na Europa de um BUS comum (EHS). Em convergência com EIB e BatiBus para KNX.
HomeRF (Home Radio Frequency Working Group)	RF	A missão do grupo de trabalho HomeRF é tornar possível uma vasta gama de produtos electrónicos de consumo que operem entre si, estabelecendo uma especificação aberta para comunicações digitais de RF (sem licença), para PC,s e produtos electrónicos de consumo em qualquer sitio e arredores da casa.
LonMark Interoperability Association	Todos	A associação LonMark tem a missão de integrar facilmente dispositivos baseados em redes LonWorks, fazendo uso de ferramentas e componentes standards.
ZIGBEE	RF	Pensa-se que este pode ser um dos standards que irá ser bastante utilizados no mundo da domótica. É uma versão melhorada do HomeRF e destinada a ambientes industriais.
Protocolos de Rede - Proprietários		
Lonworks Echelon Corp.	Todos	Redes de controlo comerciais e para a casa. Uma rede LonWorks é grupo de dispositivos trabalhando juntos para sensorizar, monitorizar, comunicar, e de algumas maneiras controlar. É muito parecido com uma LAN de PC,s.
X-10	Corrente eléctrica/RF	O protocolo mais utilizado na domótica, utiliza a rede eléctrica e facilita o controlo de dispositivos domóticos sem instalação de qualquer fio em casa, por utilizar a instalação de rede eléctrica já existente.

1.2 Domoitech

O projecto Domoitech realizado no ano lectivo de 2006/2007, consistiu em criar uma configuração e equipamento para domótica que fosse compatível com soluções que facilmente se encontrem no mercado. Outro requisito muito importante era que o custo deveria ser o mínimo possível.

A solução apresentada foi baseada numa arquitectura descentralizada, composta por dispositivos com os sensores e actuadores que estarão ligados ao barramento da rede de domótica. Cada um desses dispositivos, permite ligar várias entradas e saídas. É composto por uma placa principal, onde é feito o processamento do protocolo de domótica e a esta placa é possível conectar até 7 placas secundárias que contêm as interfaces as sensores e actuadores (p.ex: relés, optoacopladores, ...). A figura 1 ilustra a arquitectura de hardware da placa principal, e dos dois tipos de placas secundárias possíveis.

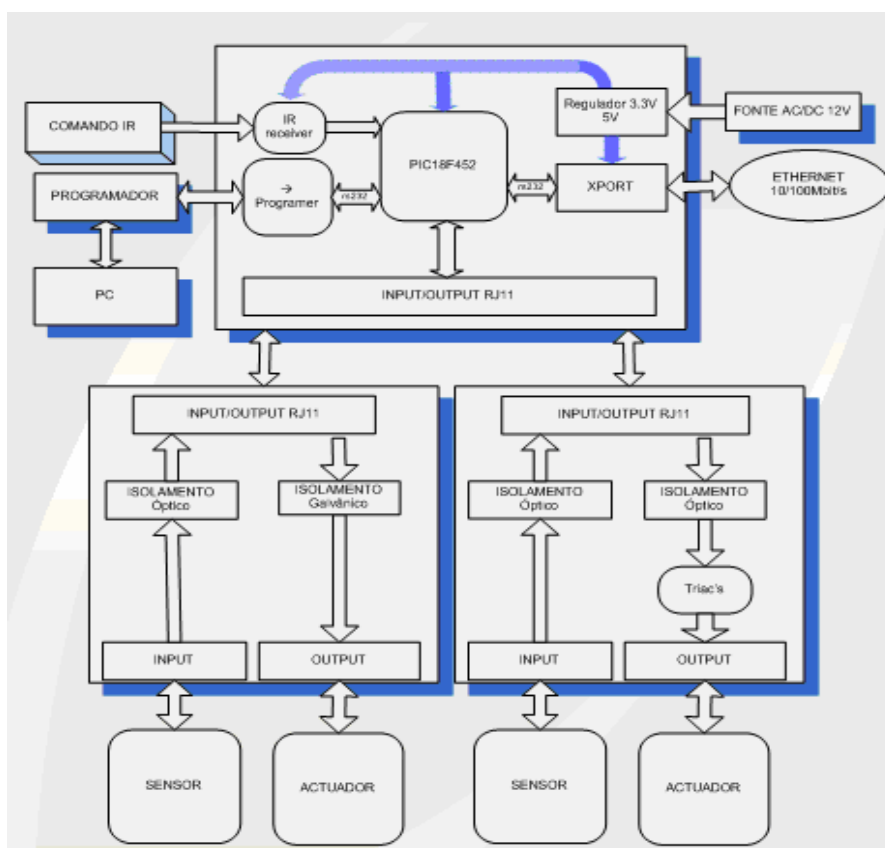


Figura 1 - Diagrama de blocos da estrutura de hardware Domoitech

Analisando a figura 1, começando pelo topo, encontramos o diagrama de blocos da estrutura para placa principal que contém; um receptor de infravermelhos, reguladores de tensão, uma interface para o programador da placa, entradas/saídas RJ11 para ligar as placas secundárias, um microcontrolador PIC18F452 [11] e uma XPORT [10]. O microcontrolador é a unidade de processamento que contém o protocolo de domótica e efectua leituras/escritas de dados das placas secundárias e da XPORT. Este último componente referido, efectua uma conversão bidireccional entre o protocolo RS232 e TCP/IP sob *Ethernet*. É a solução integrada mais compacta para permitir uma interface *web* a qualquer componente com uma interface série. Posteriormente será explicado que o barramento físico utilizado para a rede de domótica foi o TCP/IP sob *Ethernet*. As placas secundárias podem ser de dois tipos; utilizando relés ou *triac's* para os actuadores. Cada placa secundária contém duas interfaces para entradas e duas para as saídas e os respectivos isolamentos ópticos e galvânicos.

Com a arquitectura descentralizada do Domoitech, é possível ter um computador central que poderá funcionar como servidor *web*, permitindo assim o acesso pelo exterior de forma a controlar remotamente os equipamentos, e realizar algumas tarefas de gestão da rede.

Depois de ter sido efectuado um estudo aprofundado de protocolos de domótica existentes e os mais comuns no mercado, foi escolhido desenvolver o projecto utilizando o protocolo EIB. Apesar do projecto Domoitech ter sido desenvolvido com o protocolo EIB, não é totalmente compatível com outros dispositivos KNX/EIB devido ao meio físico que utiliza não estar em conformidade com a norma.

No mercado é usual encontrar a sigla KNX/EIB indicativa do nome do protocolo porque desde 14 de Abril de 1999, o EIB evoluiu para KNX que resultou da convergência entre EIB, EHS e o BatiBUS. O KNX veio acrescentar ao EIB outras funcionalidades tais como: permite a utilização de mais meios físicos e o controlo de mais sistemas, como é o exemplo de sistemas HVAC. É utilizada a sigla KNX/EIB em vez de utilizar só KNX, porque o KNX é totalmente compatível com EIB, visto a sua base ser EIB onde apenas foram acrescentados mais meios físicos que os protocolos EHS e o BatiBUS suportavam.

No projecto Domoitech, foi utilizado como meio físico o Ethernet e o protocolo de rede TCP/IP sob *Ethernet*. O KNX também suporta esse meio físico e tem uma



norma denominada por KNXnet/IP para a utilização desse mesmo meio em redes IP. Como na altura o protocolo utilizado era o EIB, o Domoitech não utilizou a norma KNXnet/IP e utilizou o *Ethernet* TCP/IP para simular um barramento *Twisted-Pair* do EIB. Contudo isto torna uma incompatibilidade com outros equipamentos KNX/EIB porque o acesso ao meio físico não respeita a norma. Apesar de o Domoitech codificar/descodificar tramas KNX/EIB, falta-lhe saber as regras para envio/recepção dessas mesmas tramas sob o protocolo de rede TCP/IP.

Outra lacuna do Domoitech é a falta de alguns serviços inerentes ao KNX/EIB. Esses serviços serão importantes para poder configurar os dispositivos KNX/EIB.

1.3 Objectivos

O objectivo deste trabalho é encontrar a melhor solução para tornar o projecto Domoitech compatível com equipamento KNX/EIB que se encontra no mercado. É necessário efectuar um estudo do protocolo KNX/EIB e encontrar também uma arquitectura de baixo custo melhorando o Domoitech mas aproveitando a estrutura do mesmo.

2 Protocolo KNX/EIB

O EIB é um protocolo de comunicação desenvolvido por um conjunto de empresas líderes no Mercado Europeu do material eléctrico com o objectivo declarado de criar um sistema que constitua uma barreira às importações de produtos e sistemas semelhantes que são produzidos nos mercados Japonês e dos Estados Unidos da América onde estas tecnologias possuem um grau de maturidade superior ao produzido na Europa. O objectivo era criar uma norma Europeia que permitisse a comunicação entre todos os dispositivos de uma instalação, esteja ela numa casa ou num edifício de escritórios. O EIB possui uma arquitectura descentralizada. Ele define uma relação elemento – a – elemento entre os dispositivos, o que permite distribuir a inteligência entre os sensores e actuadores instalados.

A EIBA é uma associação actualmente com 114 membros, empresas lideres no mercado do material eléctrico, que se iniciou em 1990, com 15 membros apenas, para implantar o uso do sistema de instalação inteligente EIB.

A EIBA tem sede em Bruxelas e os seus membros cobrem cerca de 90% do negócio do material eléctrico na Europa. As principais áreas de actuação desta associação são:

- Estabelece as directrizes técnicas para o sistema e produtos EIB, assim como estabelecer os procedimentos de ensaio e certificação de qualidade e mandar realizar esses mesmos ensaios.
- Divulga o conhecimento e experiências das empresas que trabalham com o EIB, assim como dos novos produtos e/ou inovações.
- Única entidade a poder atribuir a marca registada EIB aos produtos e aos fabricantes seus associados.
- Colabora activamente com outros organismos europeus e internacionais em todas as fases de normalização e adaptação do EIB às normas internacionais.
- Participa na iniciativa de "convergência" KONNEX, juntamente com o BCI (Batibus) e a EHSA (EHS). [5]

O KONNEX, conhecido pelas siglas KNX foi criado em 14 de Abril de 1999 com o objectivo de obter um único *standard* Europeu para a automação das casas e edifícios.

Os objectivos desta iniciativa, com o nome de "Convergência", são:

- Criar um único standard para a domótica e automação de edifícios que cubra todas as necessidades e requisitos das instalações profissionais e residenciais no âmbito europeu;
- Melhorar as prestações dos diversos meios físicos de comunicação sobretudo na tecnologia de radiofrequência, fundamental para a efectiva consolidação da domótica;
- Introduzir novos modos de funcionamento que permitam aplicar uma filosofia *Plug&Play* a muitos dispositivos típicos de uma casa;
- Envolver as empresas fornecedoras de serviços como as de telecomunicações e de electricidade, com o objectivo de desenvolver a telegestão nas casas;

Resumindo, partindo dos sistemas EIB, EHS e Batibus, trata-se de criar um único standard europeu que seja capaz de competir em qualidade, prestações e preços, com outros sistemas norte-americanos como o Lonworks ou CEBus.

Actualmente a associação KONNEX está a terminar as especificações do novo *standard* (versão 1.1) o qual será compatível com os produtos EIB instalados. Pode afirmar-se que o novo *standard* terá o melhor do EIB, do EHS e do Batibus e que aumentará consideravelmente a oferta de produtos para o mercado residencial.

A versão 1.1 contempla três modos de funcionamento:

1. S-mode (*System mode*): a configuração do modo sistema usa a mesma filosofia que o EIB, isto é, os diversos dispositivos ou modos da nova instalação, são instalados e configurados por profissionais com ajuda de um *software* (ETS) especialmente concebido para este propósito. Este é o modo mais utilizado.
2. E-mode (*Easy mode*): na configuração do modo simples os dispositivos são programados em fábrica para realizar uma função concreta. Mesmo assim devem ser configurados alguns detalhes no local da instalação mediante o uso de um controlador (como uma porta residencial) ou mediante micro-

interruptores alojados nos dispositivos (semelhante ao X-10 ou outros dispositivos proprietários que há no mercado).

3. A-mode (*Automatic mode*): na configuração do modo automático, com uma filosofia *Plug&Play*, nem o instalador nem o utilizador final têm de configurar o dispositivo. Este modo será especialmente indicado para ser usado em electrodomésticos e equipamentos de entretenimento (consolas, *set-top* boxes, áudio e vídeo).

Respeitante ao meio físico o novo standard poderá funcionar sobre:

- Par de condutores (TP1): que aproveita a norma EIB equivalente.
- Par de condutores (TP0): que aproveita a norma Batibus equivalente.
- Corrente eléctrica (PL100): que aproveita a norma EIB equivalente.
- Corrente eléctrica (PL132): que aproveita a norma EHS equivalente.
- Ethernet: utiliza a norma KNXnet/IP.
- Radiofrequência: que aproveita a norma EIB.RF
- Existe também o EIB.IR que transmite o sinal por infravermelho, até uma distância máxima de aproximadamente 12 metros. É Ideal para o uso com comandos à distância em salas ou salões onde se pretende controlar os dispositivos KNX/EIB instalados e o número destes ou as distâncias a cobrir estão dentro do limite indicado. [5]

A figura 2 ilustra que o sistema de base do KNX é o EIB e que para todos os modos de configuração, existe um *software* criado pela KONNEX para configurar os dispositivos KNX/EIB denominado por ETS (*Engineering Tool Software*).

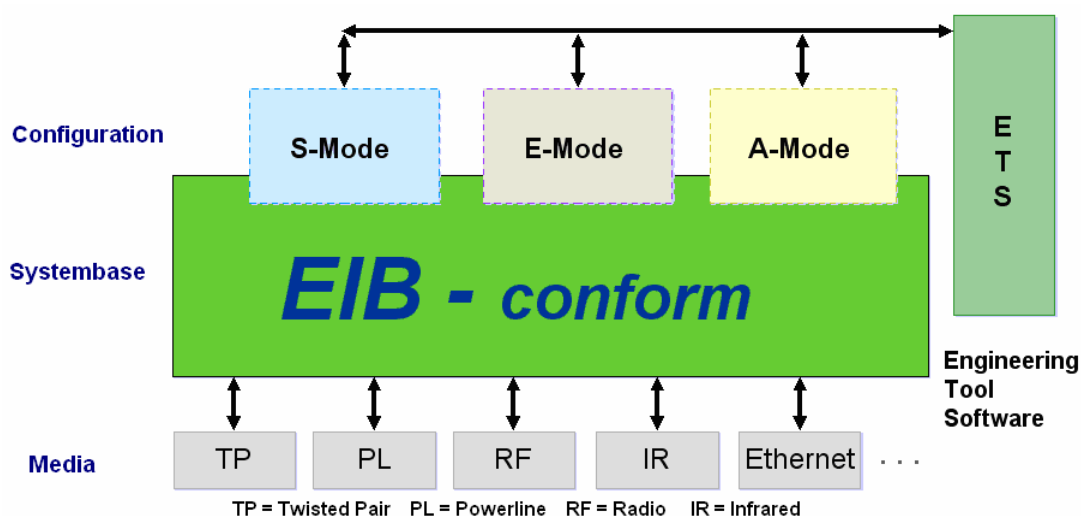


Figura 2 – Modelo do Sistema KNX [8]

A figura ilustra também os vários meios físicos possíveis de utilizar e o significado dos três pontos (...) indica que é possível a utilização de outros meios físicos, mas apenas o TP, PL, RF, IR e *Ethernet* é que estão definidos na norma.

2.1 KNX/EIB Handbook

Para desenvolver um produto KNX/EIB, é necessário adquirir o *KNX/EIB Handbook*, que contém a norma ou o Standard EN 50090 que define o KNX/EIB. Existe também a possibilidade de no caso de ser-se membro da associação, adquirir o *KNX/EIB Handbook* gratuitamente. [7]

O *KNX/EIB Handbook* encontra-se ainda na versão 1.1 desde 2004. A norma está estruturada em 10 volumes. Como a norma KNX/EIB encontra-se ainda em construção, existem alguns documentos anexados que ainda não estão terminados e aprovados. Encontram-se igualmente anexados 49 notas de aplicação.

Os volumes 1 e 2 não contêm nenhum documento e foram criados para a inclusão de documentos numa futura versão da norma.

O volume 3 é o principal porque contém as especificações do sistema. Tem oito partes divididas por: arquitectura, meio físico, comunicação, ambiente de

aplicação, gestão, interfaces standard, inter funcionamento e a última parte contém a norma KNXnet/IP.

O volume 4 contém os requisitos de hardware e testes ao *hardware*. É aqui que são definidos os requisitos eléctricos para cada meio físico e como efectuar testes aos mesmos.

O volume 5 contém o manual de certificação do produto desenvolvido em KNX.

O volume 6 contém um conjunto de perfis que especificam o comportamento dos dispositivos KNX, a fim de garantir o inter-funcionamento entre eles.

O volume 7 contém alguns exemplos de aplicação e descreve pormenorizadamente como se desenvolve essas aplicações. Por exemplo, descreve como o KNX é utilizado em sistemas de ventilação e ar condicionado.

O volume 8 contém especificação de testes a efectuar ao sistema todo e verificar se está em conformidade. Este volume é essencial e importante realizar antes da certificação.

O volume 9 descreve os componentes do sistema e dispositivos básicos como por exemplo, o BCU e BIM. O BCU (*Bus coupling units*), é um dispositivo que inclui o circuito de interface ao meio físico *Twisted-Pair* do tipo TP1. É composto por um microprocessador que implementa quase todas as camadas KNX/EIB exceptuando as de nível de aplicação. O colaborador apenas precisa de desenvolver a aplicação num microcontrolador à parte e utilizar o BCU como interface ao barramento TP1. O BIM (*Bus Interface Modules*), é basicamente um BCU mas com portas adicionais de entradas e saídas.

O volume 10 contém especificações de domínios de aplicação. A integração do KNX em outros domínios de aplicação é um dos grandes objectivos do KNX.

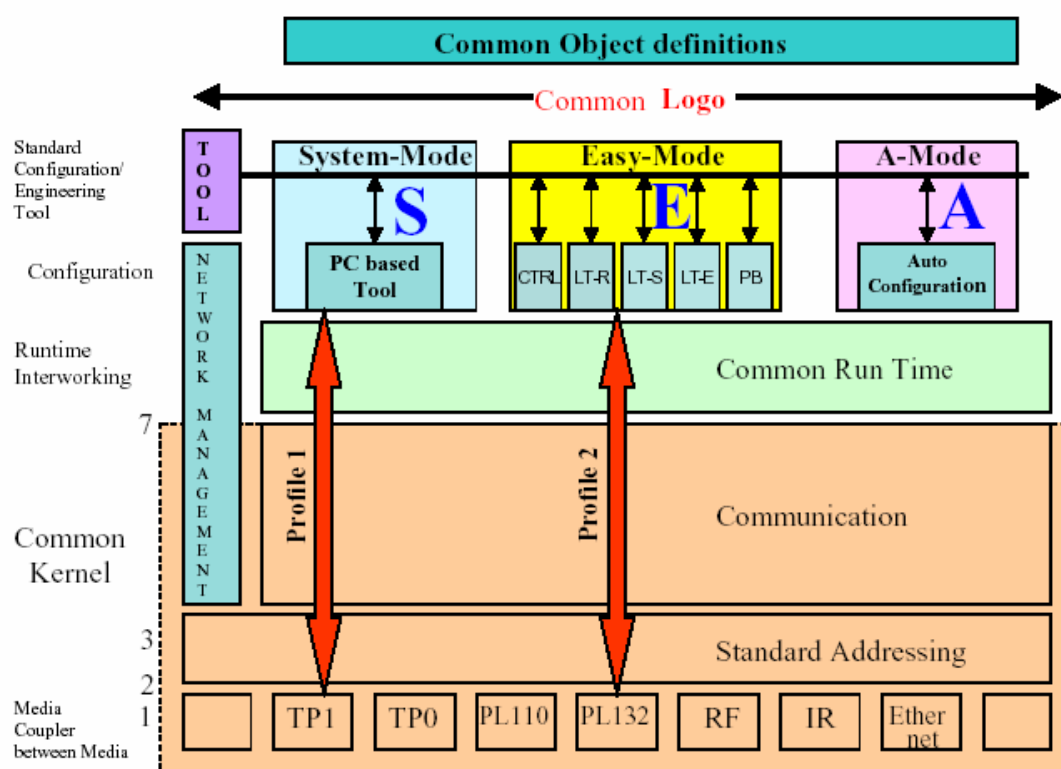
Para além destes volumes, notas de aplicação e documentos ainda em progresso, contém também 26 suplementos que dizem respeito ao volume 3.

2.2 Arquitectura

Este sub capítulo resume os principais elementos de um sistema KNX/EIB e alguns conceitos.

O KNX/EIB define a tecnologia de controlo de edifícios como uma forma especializada de automação de processos dedicados às necessidades de uma habitação. A sua arquitectura é descentralizada e distribuída, daí que se utiliza o termo de Rede KNX/EIB.

O KNX/EIB é bastante completo e contém imensos mecanismos e ingredientes que possibilitam os fabricantes de escolherem a sua própria configuração adequada para o mercado pretendido. A figura 3 ilustra como é possível optar-se por 3 métodos de configuração e escolher igualmente o meio físico a utilizar.



Ctrl = Controller Approach LT= Logical Tag (e.g. Code Wheel) PB = Push Button approach LTE= Logical Tag extended

Figura 3 - Modelo KNX/EIB [8]

Os componentes essenciais e que todos os fabricantes têm de desenvolver são os seguintes:

- *Runtime Interworking*, é um componente essencial porque contém os *Datapoints*. Os *Datapoints* representam as variáveis do processo e controlo (*inputs, outputs,...*). A definição desses *Datapoints* está contida em tabelas denominadas por *Group Objects* e *Interface Objects Properties* em que esta última contém as propriedades dos *Datapoints*. É necessário também criar *Standardized Datapoint Types* para definir os tipos das variáveis, como por exemplo: a variável de saída lâmpada é do tipo 1 Bit em que tem apenas dois estados (ligado/desligado).

- *Network Management*: é necessário criar esquemas de configuração e gestão para gerir todos os recursos da rede. Existem duas formas de configurar uma instalação KNX/EIB: uma é ao nível da rede e a outra é a configuração individual dos nós ou dispositivos KNX/EIB. A configuração pode ser feita localmente nos dispositivos, como por exemplo: carregando num botão, ou ligar o dispositivo directamente a um computador que com um software configura o mesmo, etc. O outro tipo de configuração pode ser feita através de uma rede de gestão que utiliza o barramento de comunicações KNX para enviar tramas de configuração para todos os dispositivos. Esta é a configuração mais comum em que utiliza o software ETS (que será abordado posteriormente) para enviar tramas pela rede e configurar todos os dispositivos KNX/EIB.

- *Communication System*: contém a pilha KNX/EIB desde o meio físico até à camada de Aplicação. É caracterizado pelo *KNX Common Kernel*. Ao nível mais baixo, o de meio físico, o fabricante pode optar por utilizar um destes meios físicos: TP0, TP1, PL110, PL132, RF, IR e Ethernet.

Das camadas do modelo OSI - 7, apenas a de sessão e de apresentação é que estão vazias, porque as restantes são utilizadas pelo KNX/EIB. Começando pelo nível mais baixo do modelo de camadas OSI:

- Camada Meio Físico: permite a utilização de vários meios físicos, sendo mais utilizado o de TP1. O TP0 foi adoptado do protocolo BatiBus e o TP1 foi adoptado do protocolo EIB, daí que a sua utilização seja mais comum. Estes dois meios físicos, utilizam um cabo com dois fios entrançados, onde passa a alimentação e dados, com transferências assíncronas *half-duplex* e taxas de

transferência de 2400bit/s para TP0 e 9600bit/s para o TP1. Em ambos os meios, é implementado o mecanismo de evitar colisões CSMA/CD.

Os restantes meios raramente são utilizados, mas a norma KNX/EIB permite a utilização dos mesmos e especifica os mesmos. O KNX acrescentou um novo meio, o de *Ethernet* com o intuito de permitir a sua utilização com o protocolo TCP/IP. O intuito da utilização do protocolo TCP/IP é permitir utilizar as redes *Ethernet* (IEEE 802.2), *Bluetooth* (IEEE 802.15), *Wifi* (IEEE 802.11) e *FireWire* (IEEE 1394).

- Camada Ligação de Dados: encontra-se acima de cada meio físico e permite o acesso entre o meio físico e a ligação lógica.
- Camada de Rede: é feito um *acknowledged* das tramas e contém também um contador das tramas por cada *router* que passam (*hop count*). Esta camada tem interesse em utilização de nós que tenham *routers*.
- Camada de Transporte: permite quatro tipos de comunicação, um-para-muitos (*multicast*), um-para-todos (*broadcast*), um-para-um sem conexão e um-para-um orientado à conexão.
- Camada de Aplicação: tem uma variedade enorme de serviços que permitem realizar todo o tipo de acções ou configurações nos dispositivos KNX/EIB. Estes serviços dependem dos tipos de comunicação da camada de transporte, porque por exemplo, para uns serviços é necessário uma ligação de um-para-todos e para outros serviços é necessário uma ligação de um-para-um orientado à conexão.

2.2.1 Modos de configuração

Uma das opções que o fabricante tem, é adoptar por um modo de configuração do sistema KNX/EIB adequado ao mercado destino. A opção de utilização de vários modos de configuração, é possível graças ao software ETS que utilizando a função de leitura de auto-descrição que os dispositivos KNX/EIB têm, consegue saber qual o modo de configuração que utilizam e adequa os procedimentos de configuração.

Os três modos de configuração são: *System*, *Easy* e *A-mode*.

O **System mode** é o mais versátil e mais usual encontrar porque tem uma implementação simples. A complexidade de configuração deixa de estar no dispositivo KNX/EIB passando para um software de configuração como o ETS. Este tipo de configuração, requer que o ETS tenha uma base de dados dos dispositivos KNX/EIB de cada fabricante com respectivas funcionalidades. Os fabricantes dos seus produtos são responsáveis por criarem os *templates* dos seus dispositivos e por os actualizarem. Normalmente, na compra de um dispositivo KNX/EIB, o fabricante fornece ao cliente o *template* do mesmo para inserir no ETS e assim poder configurar o dispositivo.

O **Easy mode** como podemos ver na imagem 3, representa um conjunto de vários modos:

- *Controller Mode*: este modo só suporta um número limitado de dispositivos num segmento do meio físico. Necessita de um controlador instalado nesse segmento físico que é o responsável por fazer a configuração dos dispositivos. Esse controlador normalmente só suporta uma ou poucas aplicações como por exemplo: iluminação. Este controlador faz a leitura das funcionalidades de cada dispositivo e realiza a configurações de cada um. A sua funcionalidade pode-se assemelhar ao ETS mas com funções de configuração muito limitadas.

- *Push-button Mode*: este modo, tal como o anterior, só funciona para um número limitado de dispositivos num segmento físico. Ao contrário do anterior modo, não necessita de um controlador especializado, porque cada dispositivo KNX/EIB contém um botão de configuração. Os dispositivos têm capacidade de serem configurados dinamicamente. Com isto a complexidade na configuração passa a estar no lado dos dispositivos e quando é necessário alterar algum parâmetro, terá de ser feito localmente no dispositivo.

- *Logical Tag Mode*: neste modo não é necessário uma ferramenta de configuração. Os dispositivos KNX/EIB e as suas funcionalidades, são activados através de, por exemplo; *dip-switches* para atribuir os valores a funções de configuração.

- *Logical Tag Extended Mode*: este modo é igual ao anterior em que a configuração é feita localmente através de interruptores, mas este é específico para aplicações AVAC porque estas necessitam de tramas estendidas. (ver capítulo 2.2.3).

O **A- mode** ao contrário de todos os modos anteriores que são dedicados a instalações que normalmente são fixas (não sofrem alterações regularmente), este modo é destinado a utilizadores inexperientes. Inclui mecanismos de auto configuração permitindo assim a fácil mobilidade de uma aplicação para outra.

2.2.2 Topologia

O KNX/EIB é uma rede distribuída que pode ter até 65536 dispositivos com endereços individuais de 16 bit. Em KNX/EIB existem dois tipos de endereços: de grupo ou individual. Normalmente a denotação dos endereços é feita com dois pontos e números decimais, ou seja na forma X . X . X, em que dois primeiros números decimais fazem 8 bits e o último número decimal tem o tamanho de 8 bits. O endereço de grupo não necessita de ser único pelo que um dispositivo pode ter mais do que um endereço de grupo. O endereço individual é único em cada dispositivo KNX/EIB e na denotação, o primeiro número decimal corresponde ao endereço da área, que são os 4 bits MSB do octeto 0 e o segundo número decimal da denotação, corresponde ao endereço da linha e são os 4 bits LSB do octeto 0. O octeto 1 é o que tem o endereço do dispositivo e corresponde ao terceiro número decimal na denotação dos endereços com dois pontos.

Em cada sub-rede é possível ter até 256 dispositivos. A figura 4 ilustra um exemplo de uma rede KNX/EIB com o *backbone* onde são ligadas várias áreas que contém as diversas sub-redes.

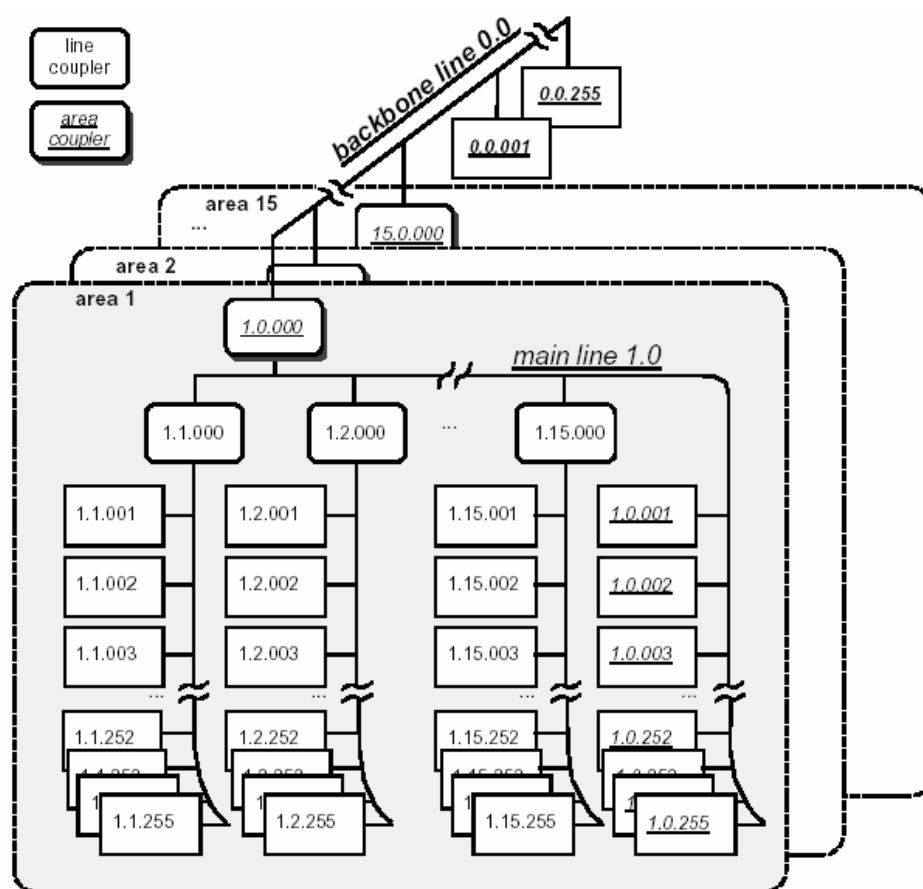


Figura 4 - Estrutura de uma rede KNX/EIB [8]

Cada área tem uma linha principal onde vão ser ligadas as várias sub-redes que se denominam por linhas. Tendo em conta que os oito primeiros bits do endereço KNX/EIB são reservados para identificação da área e linha, restam 8 bits para endereçar dispositivos, o que perfaz as 256 combinações de endereços diferentes nessa linha. Em cada área é possível ter até 15 linhas porque o endereçamento das linhas é feito com 4 bits e o endereço 0 é reservado para o acoplador entre a linha principal e o *backbone*.

Ligados ao *backbone* é possível termos no máximo 15 áreas porque o endereçamento das áreas é feito com 4 bits e o endereço 0 é reservado para definir os dispositivos KNX/EIB que se ligam directamente ao *backbone*.

2.2.3 Formato da trama KNX/EIB

A trama KNX/EIB, não tem qualquer preâmbulo que diga respeito ao meio físico porque esta é independente do meio físico. Esta trama é composta por vários cabeçalhos de quatro camadas do modelo OSI do protocolo que foram referidas anteriormente.

Uma trama KNX/EIB tem o seguinte formato:

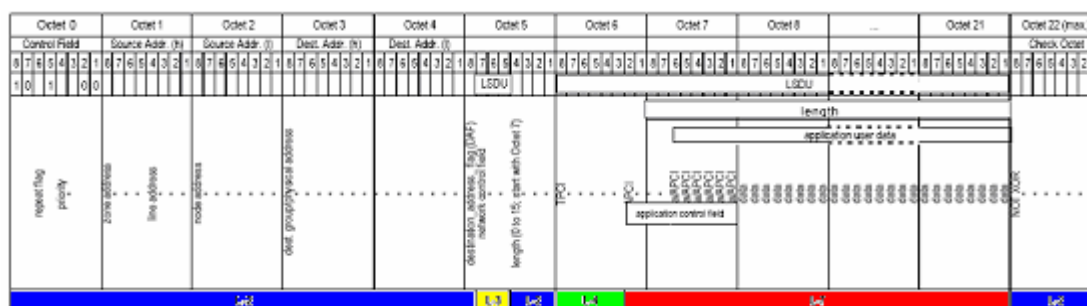


Figura 5 - Formato Trama EIB [8]

Na parte inferior da trama encontra-se uma barra de cores que facilita a visualização dos preâmbulos das várias camadas.

A barra azul, desde o octeto 0 até ao octeto 4, mais 5 bits do octeto 5 e último octeto, diz respeito à camada de ligação lógica. Neste cabeçalho, estão indicados os endereços de destino e origem, um caractere inicial de controlo e no final um octeto de controlo com um NOT XOR de todos os octetos anteriores do 0 até este octeto de controlo excluindo o próprio. A este octeto terminal, chamamos de *check octet* e serve para detecção de erros na trama. O caractere de controlo determina a prioridade da trama e distingue se a trama é normal ou se é estendida. O uso de tramas estendidas é muito raro pelo que a norma refere que este tipo de tramas ficou reservado para utilizações futuras e não definem este tipo de tramas. É neste cabeçalho que é indicado o número de octetos que a trama tem, desde o octeto 7 até o penúltimo octeto, visto que o *check octet* não entra na contagem. Assim para se saber o tamanho exacto da trama KNX/EIB soma-se a este valor, 8 octetos que serão sempre fixos.

A barra amarela, os 3 bits do octeto 5 pertencem à camada de rede e indicam se os endereços que vêm são do tipo endereços de grupo ou individuais.

A barra verde, o octeto 6 contém o cabeçalho da trama de transporte.

Na barra a vermelho, os octetos a partir do 7 excepto o último, pertencem à camada de aplicação. O tamanho máximo de uma trama KNX/EIB é de 23 octetos.

2.3 Meio Físico

Um dos grandes trunfos do KNX/EIB é a possibilidade de utilização do mesmo sob vários tipos de meios físicos. Apesar de a norma especificar cinco meios físicos, o KNX/EIB refere que é possível utilizar outros que o fabricante pretenda. Contudo, *softwares* de configuração como o ETS que será explicado posteriormente, apenas funcionam para os meios físicos descritos na norma. Se o fabricante pretender utilizar outro meio físico, deverá solicitar a sua certificação junto da Konnex, para a introdução do mesmo no software ETS. Os cinco meios físicos serão explicados resumidamente nos próximos sub-sub-capítulos.

2.3.1 Par entrançado

Existem dois tipos de rede para este meio físico: TP0 e TP1. O primeiro foi herdado do protocolo BatiBus e funciona com uma taxa de transferência de 2400 bit/s enquanto que o TP1 foi herdado do EIB e funciona com uma taxa de transferência de 9600 bits/s.

Os dados são transmitidos simetricamente através de par entrançado e a transmissão de sinais é feita por meio da diferença de tensão entre os dois condutores do cabo como é representado na figura 6.

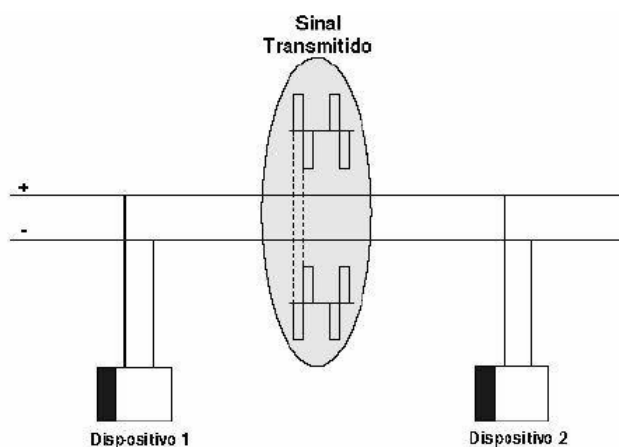


Figura 6 - Sinal balanceado entre dispositivos KNX/EIB

Uma trama KNX/EIB é enviada para o meio físico TP, no modo de caracteres série. Cada octeto da trama é enviado para o barramento TP utilizando o modo série que consistem em enviar um bit de início do caractere, seguido dos oito bits do octeto, um bit de paridade para controlo de erros e no final um bit de identificação do fim do caractere série.

O sinal transmitido no barramento é em corrente alternada com uma componente em corrente contínua. A tensão da componente corrente contínua, pode tomar os valores de 21 a 32 Vdc. O tempo de duração da transmissão de um bit é de 104 μ s. O sinal lógico '1' pode ser visto como estado inactivo da linha, o que significa que transmitir um 1 é o mesmo que não transmitir nada, assim sendo, para enviar um '1' é necessário enviar um '0' que normalmente é o bit de início do caractere. O sinal lógico '0' é feito quando a linha fica a ON durante 35 μ s, e nos restantes 105 μ s (até o final de transmissão do bit), a linha fica no estado OFF.

O acesso ao barramento é baseado no protocolo CSMA/CA (*Carrier Sense Multiple Access with Collision Avoidance*), que funciona do seguinte modo:

- Um dispositivo que pretenda transmitir uma mensagem pode começar a fazê-lo imediatamente se encontrar o barramento inactivo (desocupado), caso contrário, terá de aguardar até este ficar livre.
- Os dispositivos escutam o barramento, enquanto transmitem, com o objectivo de detectarem qualquer colisão, que ocorrerá se dois dispositivos transmitirem simultaneamente uma mensagem. Quando um dispositivo tenta impor o estado lógico "1" e detecta o estado lógico "0" (sinal de corrente na linha), pára de transmitir para permitir que o dispositivo com a mensagem prioritária o continue a fazer, como ilustra a figura 7;

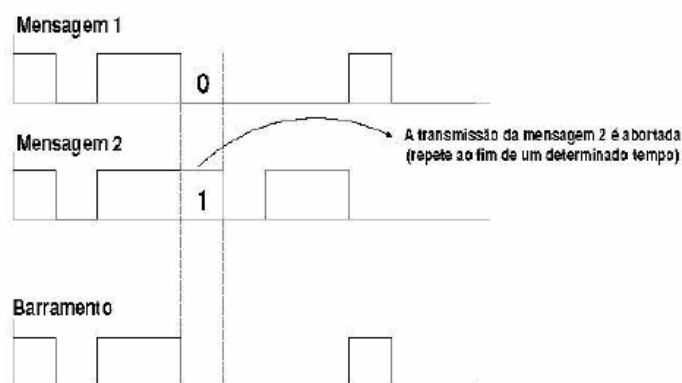


Figura 7 - Protocolo CSMA/CA

- O dispositivo com a mensagem de prioridade mais baixa, continua a escutar o barramento para quando a mensagem prioritária terminar, poder então transmitir os seus dados. Desta forma, se vários dispositivos tentarem transmitir simultaneamente, o protocolo CSMA/CA assegura que a rede só será ocupada por um destes dispositivos.

2.3.2 Rede eléctrica

Este meio físico utiliza a rede eléctrica como meio para comunicação. O protocolo X-10 também direccionado para a domótica, é conhecido pela utilização deste meio físico e muito adoptado porque permite a utilização da instalação eléctrica já existente. Contudo este meio físico tem muitas interferências e as taxas de transmissão são baixas.

O KNX/EIB tem duas especificações para este meio físico: o PL110 e o PL132.

O PL110 tem uma taxa de transferência de 1200 bits/s e foi herdado do EIB pelo que dispositivos EIB PL110 podem comunicar com dispositivos KNX/EIB PL110.

O PL132 tem uma taxa de transferência de 2400 bits/s e foi herdado do EHS que ainda o utiliza, no entanto, dispositivos KNX/EIB PL132 não podem comunicar com dispositivos EHS PL132 porque o protocolo de domótica é diferente.

A rede eléctrica funciona com sinais de corrente alternada de 230 Vac a uma frequência de 50 Hz. Para enviar sinais digitais pela rede varia-se a frequência, a modulação é do tipo *Spread frequency Shift Keying (SFSK)* onde a frequência para o valor lógico '1' é 115,2 kHz e para o valor lógico '0' é de 105,6 kHz. A duração de um bit é de 833,33 µs.

A figura 8 ilustra a codificação dos bits com valor lógico '1' e '0' num sinal de 1,8 Vp e cuja frequência varia mediante o tipo de valor lógico.

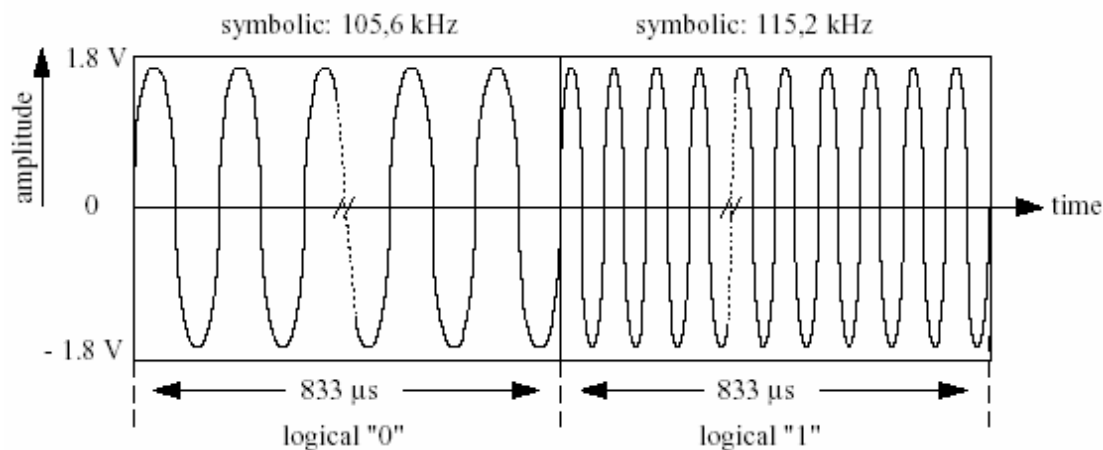


Figura 8 - Codificação sinal pela rede eléctrica

Estes sinais são sobrepostos ao sinal principal de 230 Vac / 50 Hz. A amplitude máxima do sinal transmitido não pode ultrapassar os 116 dBμV.

2.3.3 Rádio Frequência e Infravermelhos

O KNX/EIB permite a utilização do ar como meio físico de comunicação. Na norma estão definidos dois tipos de redes que utilizam o ar como meio de transmissão dos dados: a rádio frequência e os infravermelhos.

A rádio frequência transmite as tramas em sinais de rádio na banda de frequência de 868 MHz. São chamados dispositivos de curto alcance com um máximo de 25 mW de potência emitida e uma taxa de transmissão de 16,384 kBits/s. Para a utilização da rádio frequência em KNX/EIB, podem ser utilizados componentes de baixo custo que permitam comunicação bidireccional ou unidireccional.

A modulação dos sinais em rádio frequência é do tipo *Frequency-shift keying* (FSK) ou modulação por comutação da frequência. Como estes meios são abertos é necessário proteger a informação codificando-a. A codificação que a rádio frequência utiliza é a codificação de Manchester.

Devido à propriedade de este meio físico ser aberto, é necessário alterar o domínio dos endereços KNX/EIB, para endereços maiores. Por exemplo, no caso de existirem duas redes KNX/EIB vizinhas, podem ocorrer interferências tais como: ao enviar um pedido de ligar a lâmpada no endereço 1/1/2 na rede KNX/EIB de uma

casa, além de ligar esta, poderá ligar também outro dispositivo com o mesmo endereço na rede KNX/EIB da casa vizinha. Estes endereços maiores, são uma combinação de endereços KNX/EIB e o número de série de cada emissor/receptor de rádio frequência. Com esta solução é garantido que o endereço é único.

A especificação da utilização dos infravermelhos como meio de comunicação, está definida na norma EIB e que a norma KNX/EIB manteve sem efectuar nenhuma alteração. Este meio de comunicação normalmente é utilizado nos comandos de infravermelhos que permitem controlar dispositivos KNX/EIB. Por exemplo, é possível ter um comando de infravermelhos que envie tramas KNX/EIB sob infravermelhos, para uma *gateway* que contém um receptor de infravermelhos e que passa as tramas para a rede KNX/EIB que poderá utilizar por exemplo o meio físico TP1. A transmissão de dados por infravermelhos é assíncrona e pode ser unidireccional ou bidireccional mas em *half-duplex*. A frequência do sinal que é emitido pelo emissor de infravermelhos é de 447,5 kHz e o tipo de modulação é em amplitude. A taxa de transmissão é de aproximadamente 7000 bits/s e a comunicação pode ser bidireccional ou unidireccional como no caso dos comandos de infravermelhos. Os endereços KNX/EIB ao contrário da rádio frequência, não são estendidos, são os endereços KNX/EIB normais.

2.3.4 Ethernet

Este meio físico ao contrário dos anteriores, não tem nenhum documento na norma que o especifique. Apenas tem uma referência de que o meio físico *Ethernet* é utilizado sub o protocolo de rede IP (*Internet Protocol*). O KNX/EIB, referencia uma norma, herdada do EIB, a EIB.net, que permite a utilização do KNX/EIB sob redes TCP/IP. A norma denominada por KNXnet/IP está definida na parte 3/8 do *KNX Handbook*. Está estruturada em cinco capítulos, sendo que o primeiro é uma introdução e uma visão geral do que é a norma e como está estruturada e os quatro restantes capítulos são a descrição dos módulos do KNXnet/IP que serão explicados no capítulo 3 desta dissertação.

A utilização deste meio físico é utilizada em paralelo com outro meio físico como por exemplo o KNX/EIB TP1. A norma KNXnet/IP define um servidor que funciona como uma *gateway* e que interliga a rede KNX/EIB a uma rede IP.

2.4 Comunicação

Como foi referido no sub capítulo de Arquitectura, o sistema de comunicação do KNX/EIB é constituído por uma pilha protocolar estruturada num conjunto de camadas semelhante ao modelo de 7 camadas OSI.

No caso do protocolo KNX/EIB, apenas são implementadas cinco camadas: Física, Ligação lógica, Rede, Transporte e por último, a de Aplicação.

A camada física como foi dito anteriormente, é independente das camadas superiores porque o KNX/EIB tem um mecanismo que o permitiu ser independente do meio físico. Esse mecanismo será explicado posteriormente. As restantes camadas serão explicadas neste sub capítulo.

2.4.1 Camada de Ligação Lógica

A camada de ligação lógica, efectua a interface entre o nível físico e a camada de rede. Esta camada costuma vir associada ao meio físico e é aqui que é feito o mecanismo que permite que todas as camadas superiores a esta, sejam independentes do meio físico. Os serviços existentes estão representados na figura 9.

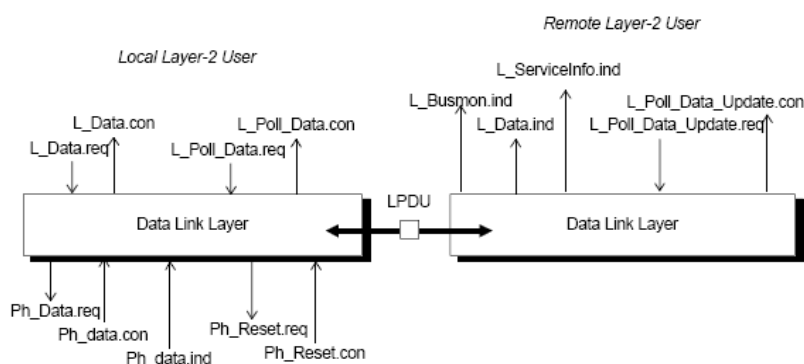


Figura 9 – Serviços da camada de ligação lógica [8]

A trama que é tratada neste nível, denomina-se por LPDU e encapsula os dados que vão para as camadas superiores. Numa trama LPDU, o cabeçalho que é tratado neste nível da ligação lógica, examina o formato da trama, endereço de destino e decide se essa trama lhe é destinada. Esta camada, detecta erros de

transmissão e efectua repetições de tramas. Os serviços desta camada serão explicados em pormenor no sub-sub-capítulo 2.5.2.

Neste nível é feito uma verificação da consistência dos dados recebidos através do teste que é feito utilizando o último octeto da trama, o *check octet*, que realiza um NOT XOR aos restantes octetos da trama.

2.4.2 Camada de Rede

A camada de rede, efectua a interligação entre a camada de ligação lógica e a camada de transporte. Esta camada, recebe tramas enviadas pela camada de ligação lógica, utilizando o serviço *L_Data* e disponibiliza os serviços; *N_Data*, *N_GroupData* e *N_Broadcast* para a camada de rede como ilustra a figura 10.

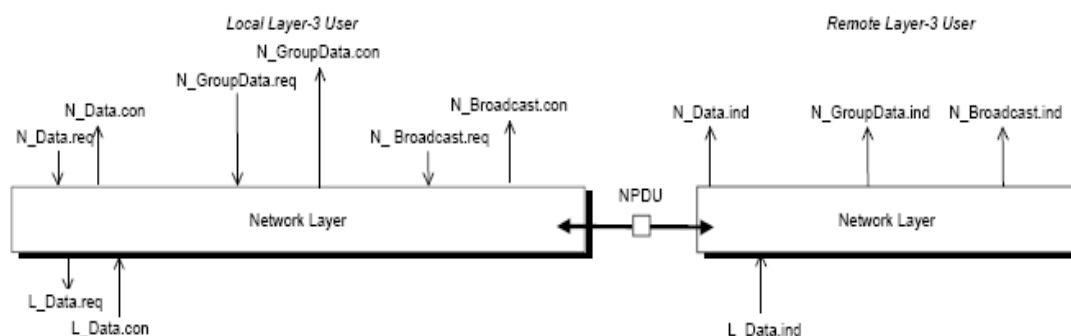


Figura 10 – Interface com a camada de rede entre os dispositivos KNX/EIB [8]

A trama que circula no nível de rede, é denominada por NPDU e corresponde a uma trama que tem encapsulado os dados que serão enviados às camadas superiores. Em relação ao nível de ligação lógica, esta trama é a resultante de uma trama LPDU sem o cabeçalho do nível de ligação lógica que contém: o endereço destino, origem, a informação do comprimento e o octeto de controlo de erros. O cabeçalho da trama NPDU é constituído por apenas 3 bits situados no octeto 5 e cuja funcionalidade é de indicar se os endereços KNX/EIB de destino são do tipo endereços de grupo, individuais ou de *broadcast*. Se o endereço de destino é do tipo individual, a trama é enviada para a camada de transporte utilizando os serviços do *N_Data*. Se o endereço de destino KNX/EIB é de grupo, a trama é enviada utilizando os serviços *N_GroupData*. Quando o endereço KNX/EIB de destino é 0 e o tipo de

endereço é *broadcast*, a camada de rede envia a trama para a camada de transporte, utilizando os serviços de *N_Broadcast*.

2.4.3 Camada de Transporte

A camada de transporte realiza a interface entre a camada de rede e a camada superior de aplicação. Implementa os vários modos de comunicação: o modo *multicast*, *broadcast*, um-para-um sem conexão e um-para-todos orientada à conexão. Cada modo de comunicação especifica um ponto de acesso que contém vários serviços. Tal como os outros serviços, estes contêm três primitivas, o *request (req)*, *confirm (con)* e o *indication (ind)*. A figura 11 ilustra os diferentes pontos de acesso dos modos de comunicação e os serviços que contêm.

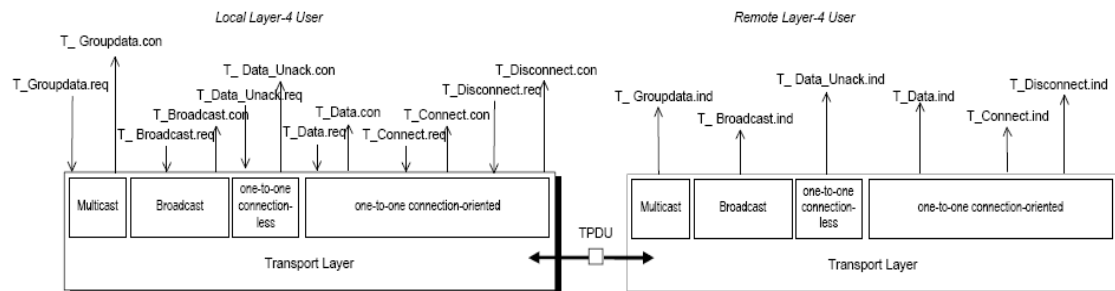


Figura 11 – Interface com a camada de transporte entre os dispositivos KNX/EIB [8]

A trama da camada de transporte, é denominada por TPDU e corresponde à trama NPDU mas reduzida, pois não tem o cabeçalho de controlo de rede. Esta trama contém os dados da camada de aplicação e o cabeçalho da camada de transporte. Este cabeçalho é composto por um campo de controlo de com o tamanho de 5 bits situados no octeto 5 da trama KNX/EIB. Este campo indica o código do serviço da camada de transporte.

2.4.4 Camada de Aplicação

A camada de aplicação corresponde à última camada superior do modelo OSI. Os serviços que esta camada tem, são respectivos aos quatro modos de comunicações da camada de transporte: o modo *multicast*, *broadcast*, um-para-um sem conexão e um-para-todos orientada à conexão.

Como foi referido anteriormente, uma trama KNX/EIB pode ter até um máximo de 23 octetos. A aplicação utiliza dois bits do octeto 6 e o octeto 7 para identificar o serviço. Os restantes octetos são para dados com um limite de 14 octetos. A trama de aplicação é denominada por APDU e vai encapsulada na trama de transporte.

Os serviços desta camada serão explicados a seguir assim como os objectos de comunicação do EIB, denominados por EIS.

2.4.4.1 Serviços

Como foi explicado anteriormente, os serviços da camada de aplicação estão organizados em quatro modos de comunicação.

Os serviços do modo de comunicação *broadcast*, servem para configurar a rede KNX/EIB. Os serviços do modo um-para-um orientado à conexão servem para aceder à memória do microcontrolador que implementa o protocolo. Quanto ao modo um-para-um sem conexão, serve para alterar ou ler as propriedades dos objectos KNX/EIB como por exemplo; os *device objects* cujo conceito será explicado num sub capítulo posterior.

O modo de comunicação mais utilizado é o de *multicast* que envia tramas de escrita para alterar os valores das saídas do dispositivo KNX/EIB ou trama de leitura para saber qual o estado das entradas. O Domoitech só utiliza os serviços deste modo, por isso apenas serão explicados em pormenor os três tipos de serviços: *GroupValue_Read*, *GroupValue_Write* e *GroupValue_Response*.

Estes serviços alteram ou lêem estados de saídas/entradas de um ou mais dispositivos KNX/EIB que pertençam ao endereço de grupo destino da trama recebida. O modo de comunicações *multicast* só funciona com endereços de grupo e não com endereços individuais. Os endereços KNX/EIB individuais são utilizados para realizar gestão e configuração dos dispositivos KNX/EIB, porque a actuação/leitura das saídas/entradas, é feita através de endereços KNX/EIB de grupo.

O código do serviço que é transportado no cabeçalho da trama da camada de aplicação, pode ter até 10 bits e em alguns dos casos só são necessários 4 bits. Nos serviços do *multicast*, só serão necessários 4 bits para os identificar. Assim sendo,

os dados poderão preenchidos a partir do bit 6 do octeto 7 e os restantes bits tomam o valor 0.

Os códigos para os serviços são os seguintes:

[illegible]

Figura 12 – Códigos dos serviços KNX/EIB modo multicast [8]

É de notar que para estes serviços o cabeçalho da trama de rede terá de indicar que pelo menos o endereço de destino é um endereço de grupo. Por esta razão, o código do serviço só é validado em conjunto com o cabeçalho de rede.

Serviço GroupValue Read

Este serviço serve para enviar tramas com pedidos de leitura de estados de variáveis ou E/S. O código deste serviço tem o valor decimal 0 quando retirados os bits 8, 7, 2 e 1 do octeto 6 e o octeto 7. Para realizar essa validação do código, tem de ser aplicada uma máscara de bits com o valor hexadecimal *0xC3FF* aos octetos 6 e 7.

O receptor da trama requerente deste serviço, deverá enviar uma resposta com os estados pedidos, utilizando o serviço *GroupValue_Res*.

Um exemplo de uma trama KNX/EIB com este serviço a efectuar um pedido de *GroupValue_Read* para o endereço de grupo 0/0/23 é o seguinte:

Octeto 0	Octeto 1	Octeto 2	Octeto 3	Octeto 4	Octeto 5	Octeto 6	Octeto 7	Octeto 8
188	0	1	0	23	241	0	0	163

Figura 13 – Exemplo Trama GroupValue_Read

Serviço GroupValue_Res

Este serviço serve para enviar tramas de resposta aos pedidos de leitura de estados de variáveis ou E/S. O código deste serviço tem o valor decimal 1 quando retirados os bits 8, 7, 2 e 1 do octeto 6 e os bits 8 e 7 do octeto 7. Para realizar essa validação do código, aplica-se uma máscara de bits com o valor hexadecimal 0xC3C0 aos octetos 6 e 7.

Os estados das variáveis vão guardados nos dados, ou seja, são colocados a partir do octeto 7. Se um dos estados é um valor binário então é possível colocar no octeto 7 porque sobram 6 bits para dados.

Um exemplo de uma trama de resposta à anterior da Figura 13, em que só existe um dispositivo binário com estado 0 e com o endereço de grupo 0/0/23, é o seguinte:

Octeto 0	Octeto 1	Octeto 2	Octeto 3	Octeto 4	Octeto 5	Octeto 6	Octeto 7	Octeto 8
188	0	23	0	1	241	0	64	227

Figura 14 - Exemplo Trama GroupValue_Res

Serviço GroupValue_Write

Este serviço serve para enviar tramas de escrita em que mudam o estado das variáveis ou E/S que estejam no endereço de grupo destino. O valor do estado vai no octeto 7 para o caso de ser binário ou vai no octeto seguinte para o caso de ser um valor decimal de 8 bits. O código deste serviço tem o valor decimal 2.

A figura 15 ilustra o exemplo de uma trama KNX/EIB com este serviço que envia um pedido para o endereço de grupo 0/0/23, para modificar um estado que tomará o valor binário 1.

Octeto 0	Octeto 1	Octeto 2	Octeto 3	Octeto 4	Octeto 5	Octeto 6	Octeto 7	Octeto 8
188	0	1	0	23	241	0	129	37

Figura 15 – Exemplo Trama GroupValue_Write

2.4.4.2 EIS

Os objectos de comunicação do EIB, são os chamados EIS (*EIB Interworking Standards*), que na norma KNX/EIB não têm essa denominação, mas têm a denominação de *Datapoint Types*. Esta nova denominação faz mais sentido visto que os EIS são tipos de dados.

O Domoitech reconhece apenas dois tipos de EIS: *SWITCHING* e *DRIVE CONTROL*. No KNX/EIB estes dois tipos de EIS resumem-se a apenas um *Datapoint Type*, o *boolean*. Como o Domoitech foi realizado com os EIS, serão explicados os dois EIS que ele implementa.

EIS 1 - Switching

Este objecto, implementa uma função com 1bit, logo é possível ter dois estados para actuar numa carga do tipo *ON/OFF* ligada ao actuador. É utilizada esta função para as lâmpadas. No entanto as saídas das lâmpadas podem ser utilizadas para qualquer carga do mesmo tipo, como por exemplo: televisões, portas eléctricas, etc.

O EIS 1, tem o valor de 1 bit e o código dele é 10 e que pode ter os seguintes estados: (*on / off*), (*enable / disable*), (*alarm / no alarm*) e (*true / false*).

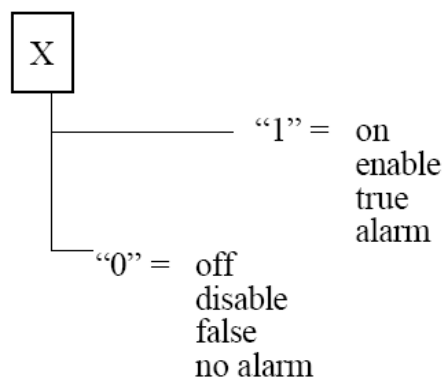


Figura 16 – EIS Switching [8]

EIS 7 – Drive Control

Este objecto consiste em duas subfunções denominadas por “*move*” e “*step*”. Com a subfunção “*move*” é possível colocar o *drive* em movimento para um sentido ou para o outro. O tempo de movimento é escolhido pelo utilizador e é facilmente configurável. Com a subfunção “*step*” é possível realizar movimentos graduais em ambos os sentidos. Este tempo de movimento gradual é também facilmente configurável. Este objecto é utilizado para controlar por exemplo, os estores.

A saída do actuador é também binária, mas neste caso existem duas subfunções: uma coloca a saída num determinado estado a um determinado tempo e a outra subfunção, também com os dois estados, mas o tempo é bastante inferior em relação ao outro.

A subfunção *Move* é utilizada para fechar ou abrir totalmente o estore e pode ter dois estados possíveis: mover para baixo ou mover para cima. A figura 17 ilustra os dois estados possíveis.

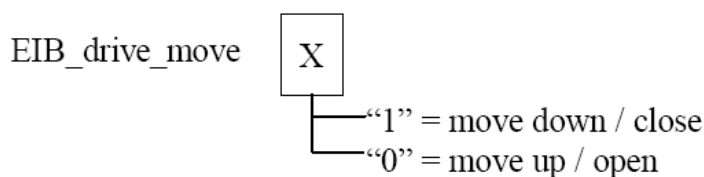


Figura 17 – EIS Drive Control – Move [8]

A subfunção *Step* é utilizada para fechar ou abrir gradualmente o estore e pode ter dois estados possíveis: mover para baixo ou mover para cima. A figura 18 ilustra os dois estados possíveis.

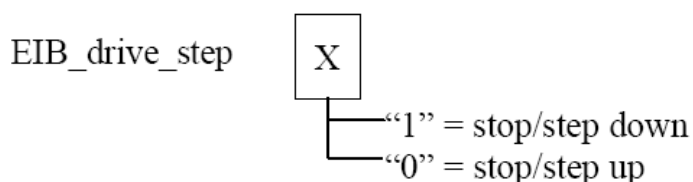


Figura 18 – EIS Drive Control – Step [8]

Para este EIS será necessário uma máquina de estado mais complexa em que são necessários quatro estados possíveis: em movimento, parado, gradualmente para cima e gradualmente para baixo. A figura 19 ilustra a máquina de estados.

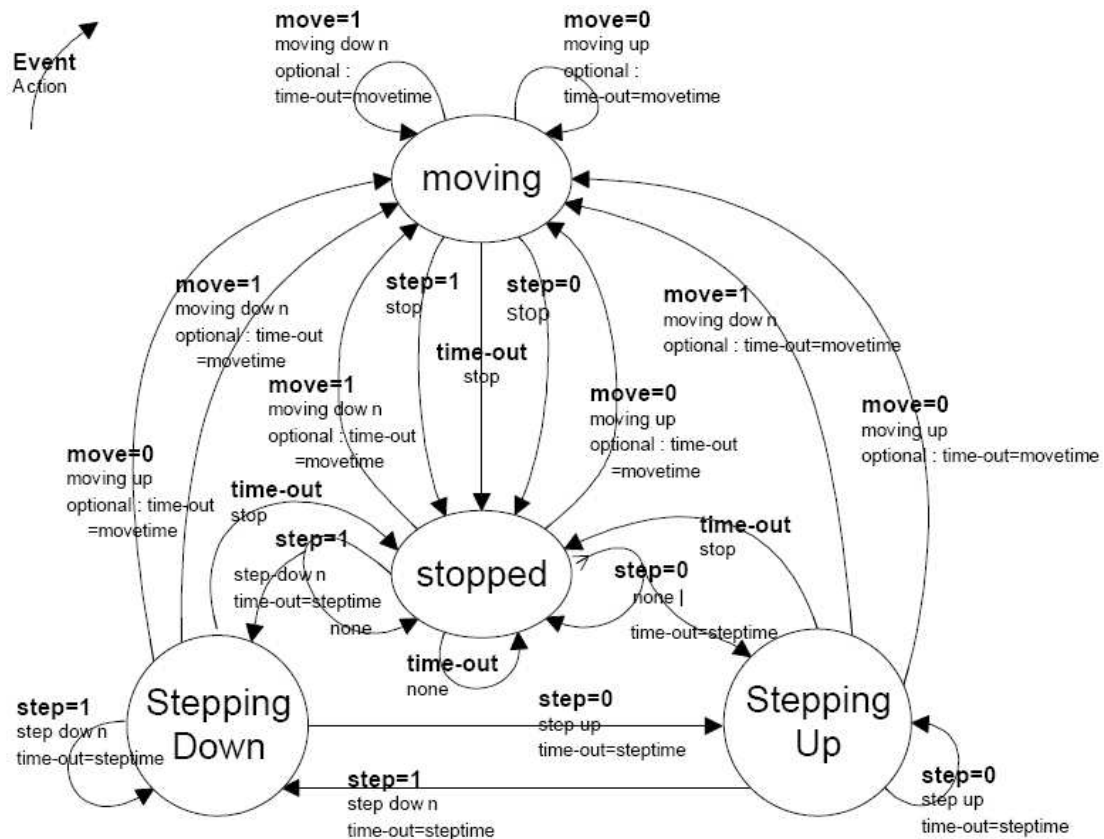


Figura 19 – Máquina de estados do Drive Control [8]

2.5 Interfaces Standard de Comunicação

O KNX/EIB especifica dois tipos de *interfaces standards*: um para acesso ao meio físico e outro para acesso à camada de ligação lógica. O primeiro *standard* denominado de *Physical External Interface (PEI)*, estabelece regras de comunicação entre a unidade que contém o *interface* físico (BCU, BIM,...) e as restantes camadas do protocolo KNX/EIB. O segundo *standard*, é denominado por *External Message Interface (EMI)* e define as regras de comunicação entre a camada de ligação lógica e a camada de rede. Este *standard* permitiu que o KNX/EIB fosse independente do meio físico.

2.5.1 Physical External Interface

O PEI contém especificações eléctricas/mecânica e de software que permitem efectuar o acesso ao meio físico. Esse acesso pode ser feito através de uma transmissão de dados em paralelo ou série.

Existem duas versões para a especificação eléctrica/mecânica: a versão de *interface* de *hardware* com 10 pinos e a versão de 12 pinos. Devido à existência de resistências que podem tomar diferentes valores, entre os pinos 5 e 6 em ambas as versões de *hardware*, permitiu criar 21 tipos de ligações PEI.

Os 21 PEI podem ser agrupados em quatro categorias diferentes:

- Utilizações especiais:
 - PEI tipo 0 – para utilizar em aplicações onde não existe nenhum adaptador, por exemplo, sem a resistência entre os pinos 5 e 6;
 - PEI tipo 1 – este PEI é reservado para utilizar quando não existe outro tipo de PEI ou enquanto o outro tipo de PEI não se inicializar;
 - PEI tipo 20 – serve para o fabricante efectuar o *download* de configurações para a unidade que contém o *interface* físico (BCU, BIM,...);
- Reservados para futuras extensões: PEI tipos 3, 5, 7, 9, 11, 13, 15, 18;
- Comunicações em Paralelo: PEI tipo 2, 4, 6, 8, 17 e 19;
- Comunicações em Série: PEI tipo 10, 12, 14 e 16.

Os tipos de PEI mais utilizados são os de comunicação série. Dentro desses tipos, os PEI tipo 14 e 12 servem para comunicações série síncronas e os restantes para comunicações assíncronas à excepção do PEI tipo 10 que permite comunicação síncrona ou assíncrona. Este último PEI é o mais comum devido à sua versatilidade apesar de que em quase 100% dos casos é utilizado no modo assíncrono.

Como foi referido anteriormente, nas especificações eléctricas/mecânicas existem dois tipos de *interface* sendo que o mais utilizado é o de 10 pinos. A figura 20 ilustra os sinais de cada pino numa configuração de *interface* de 10 pinos.

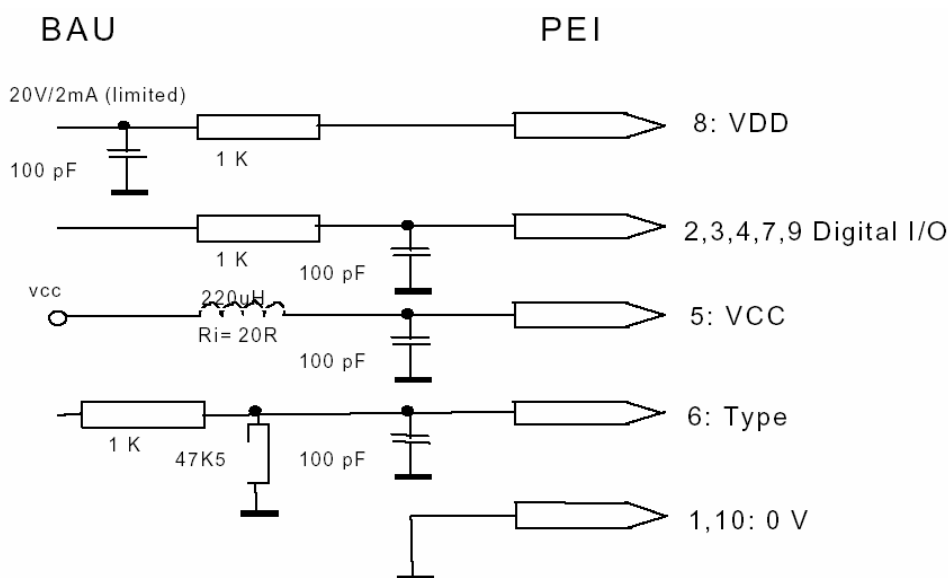


Figura 20 - Diagrama de pinos de um conector PEI [8]

O pino 5 tem uma tensão de +5 Vdc com uma corrente máxima de 10mA e o pino 8 tem uma tensão de +24 Vdc com uma corrente máxima de 2mA.

Os pinos 2, 3, 4, 7 e 9, são de Entradas/Saídas dos dados, que numa comunicação série correspondem directamente aos sinais: SCLK, RXD/RDI, TxD/TDO, CTS e RTS.

O pino 6 é para selecção do tipo de PEI, através da colocação de uma resistência entre o pino 6 e pino 5 e que mediante o valor dessa resistência, é seleccionado o PEI. A tabela seguinte contém os valores possíveis dessa resistência e que tipo de PEI definem e ainda uma breve descrição da função de cada tipo de PEI.

Tabela 2 - Valores de resistência para selecção do tipo de PEI

PEI Tipo	Descrição do tipo de PEI	Resistência (k Ω) + tolerância
0	Sem adaptador	Sem resistência
1	Utilizado quando a aplicação está parada	910 5%
2	4 entradas, +1 saída (LED) - Paralelo	430 5%
3	Reservado	255 1%
4	2 entradas / 2 saídas, +1 saída (LED) - Paralelo	187 1%
5	Reservado	140 1%
6	3 entradas / 1 saída, +1 saída (LED) - Paralelo	107 1%
7	Reservado	84.5 1%
8	Reservado	66.5 1%
9	Reservado	54.9 1%
10	Protocolo de comunicações série síncrono ou assíncrono tendo como protocolo por defeito p FT 1.2 no modo assíncrono. É o mais utilizado.	45.3 1%
11	Reservado	37.4 1%
12	Comunicação série síncrona	30.1 1%
13	Reservado	24.3 1%
14	Comunicação série síncrona com protocolo <i>data block</i>	19.1 1%
15	Reservado	14.7 1%
16	Comunicação série síncrona	11.0 1%
17	Entradas/Saídas programáveis	7.50 1%
18	Reservado	4.53 1%
19	4 saídas, +1 saída (LED) - Paralelo	2.00 1%
20	<i>Download</i> de configurações para o BCU/BIM	0

Tendo em conta que o tipo de PEI mais utilizado é o tipo 10 no modo de comunicação assíncrono, apenas este será apresentado explicando o funcionamento do protocolo que é utilizado por defeito, o FT1.2.

2.5.1.1 Protocolo FT 1.2

Este protocolo é baseado na norma internacional: IEC 870-5-1. É utilizada uma transmissão de dados balanceada: isto quer dizer que cada estação pode ser simultaneamente uma estação primária (inicia uma transmissão) e uma estação secundária (recebe a mensagem). Qualquer estação pode iniciar uma transmissão porque têm iguais direitos de acesso ao meio, portanto a relação é do tipo Mestre/Mestre.

São utilizados três fios: um para recepção de dados (RxD), um para transmissão de dados (TxD) e outro é a massa com sinal de 0 V. A transmissão de dados é feita com 8 bits de dados e 1 *stop* bit com paridade par. A taxa de transmissão pode ter um dos seguintes valores: 1200, 2400, 4800, 9600 e 19200, mas por defeito o valor utilizado é 19200.

O FT1.2 tem três tipos de tramas: de comprimento fixo, de comprimento variável e de um só caractere.

As tramas de comprimento fixo contêm: um caractere inicial cujo valor é sempre 10 em hexadecimal e serve para identificar o início da trama, um caractere de controlo, um caractere de *checksum* e caractere que identifica o final da trama e que tem sempre o valor 16 em hexadecimal.

As tramas de comprimento variável são as mais utilizadas porque só nestas tramas é que é possível enviar dados de utilizador (tramas KNX/EIB). Estas tramas são iniciadas pelo caractere de início com o valor 68 em hexadecimal, depois têm dois caracteres que contêm o tamanho dos dados de utilizador, um segundo caractere de início de trama com o valor 68 em hexadecimal, o caractere de controlo procedido pelos dados de utilizador, e no final da trama, um caractere de *checksum* e para finalizar, o valor 16 em hexadecimal.

Outra trama também muito utilizada para realizar o *acknowledge*, é a trama de um só caractere. O valor do caractere é de E5 em hexadecimal e é enviado para dar conhecimento de uma trama bem recebida.

Cada caractere das tramas é enviado pela linha (o nível binário da linha é 1), utilizando um bit de início com o valor binário 0, seguido dos 8 bits de codificação do caractere, um bit de paridade e termina com um bit de fim de trama que tem o valor binário de 1.

Em todas a tramas existe o caractere de *checksum*, que realiza a soma aritmética em módulo de 256 para os casos de ocorrer overflow, sobre todos os

octetos dos dados de utilizador. No caso das tramas de comprimento fixo, este caractere é igual ao caractere de controlo.

O caractere de controlo contém informação que caracteriza a direcção da mensagem, o tipo de serviço e suporta mecanismos de controlo de forma a eliminar duplicação de mensagens. Este caractere é diferente para tramas enviadas pela estação primária e tramas enviadas pela estação secundária.

A figura 21 ilustra o caractere de controlo de uma trama enviada pela estação primária.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIR	PRM = 1	FCB	FCV	Function code			

Figura 21 - Caractere de controlo de uma trama FT 1.2 de uma estação primária

O bit 7 do caractere indica a direcção da transmissão da trama: 1 para o sentido de envio por parte do BAU/BIM para o outro dispositivo externo e 0 para o sentido contrário onde a trama parte do dispositivo externo para o BAU/BIM.

O bit 6 indica que a trama parte da estação primária e toma o valor de 1.

O bit 5 serve para implementar o mecanismo de evitar repetições de tramas, funcionando como um contador binário de tramas. Alterna entre 0 e 1 sucessivamente para tramas de Envio/Confirmação. Quando a estação primária envia a primeira trama, coloca o bit contador com o valor 0 e se não receber nenhuma confirmação, volta a enviar uma nova trama com o mesmo bit contador. Só quando receber uma confirmação é que comuta o estado do bit.

O bit 4 encontra-se sempre activado (valor 1) quando a comunicação é feita através do serviço SEND_UDAT. Este serviço serve para enviar dados de utilizador, logo o tipo de trama utilizado é de comprimento variável.

Os restantes 4 bits indicam o código de função e as funções possíveis são: SEND_RESET que efectua o *reset* do dispositivo remoto, SEND_UDAT que serve para enviar tramas com dados de utilizador (tramas KNX) e REQ_STATUS que pede uma informação sobre o estado do dispositivo remoto.

O caractere de controlo de uma trama enviada pela estação secundária é um pouco diferente e a imagem 22 ilustra as diferenças.

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
DIR	PRM=0	RES	DFC	Function code			

Figura 22 - Caractere de controlo de uma trama FT 1.2 de uma estação secundária

Os bits 7 e 6 têm a mesma função que no caractere de controlo da trama enviada por uma estação primária, sendo que o bit 6 agora toma o valor 0.

O bit 5 é reservado e o valor dele é sempre 0. O bit 4 serve para indicar à estação primária que a próxima mensagem que enviar poderá causar um *overflow*.

Os restantes 4 bits tal como no caso anterior, servem para indicar o código da função, mas neste caso as funções possíveis são: CONFIRM_ACK que envia tramas de *acknowledge* positivo, CONFIRM_NACK que envia tramas de *acknowledge* negativo e RESPOND_STATUS que envia tramas de resposta ao REQ_STATUS.

É permitida a transmissão simultânea de dados, visto existirem duas linhas. Contudo, uma estação primária só aceita uma nova mensagem quando a anterior foi entregue com ou sem sucesso. Os erros só são detectados pela estação que recebe a mensagem pois se a estação secundária recebe uma trama corrompida, não envia nenhuma resposta e a estação primária só soube que a trama não foi entregue devido ao tempo de espera de resposta ter terminado. No entanto, a estação primária pode receber a resposta com erros e neste caso volta a retransmitir a trama.

A figura 23 ilustra um exemplo de uma transmissão de dados de utilizador com sucesso e outro com distúrbios.

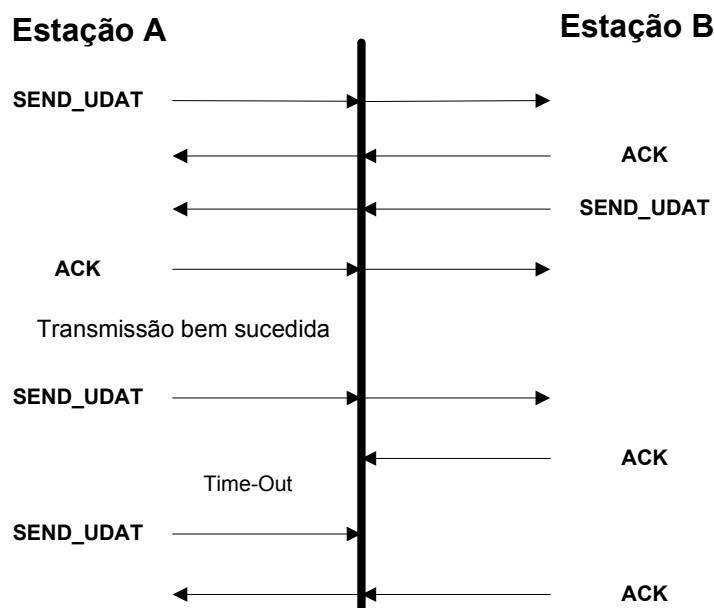


Figura 23 - Exemplo de transmissão de dados por FT 1.2

2.5.2 External Message Interface

O EIB criou o conceito da *External Message Interface (EMI)* que permitiu o EIB ser independente do meio físico. A EMI serve para troca de tramas entre a camada de ligação lógica e a camada de rede. Contudo, como o EIB definiu a existência de dois tipos de EMI, a EMI1 e EMI2, o KNX/EIB corrigiu essa diversidade que podia levar a incompatibilidades entre equipamentos de diferentes fabricantes. O KNX/EIB introduziu a *Common External Message Interface* criando assim uma única especificação para a EMI com a nova denominação de cEMI, garantindo a compatibilidade total entre produtos de fabricantes diferentes.

Uma mensagem cEMI é uma estrutura genérica que é independente do meio físico e que pode conter informação adicional como o *timestamp* ou outro.

A figura 24 ilustra em que camadas é feita o fluxo de mensagens cEMI.

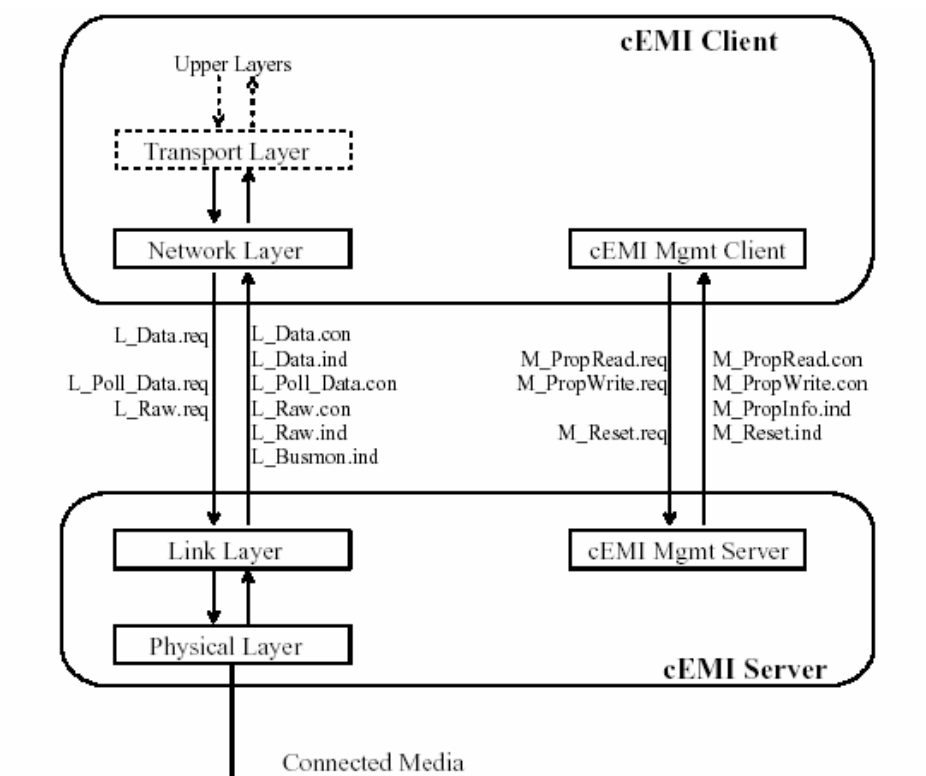


Figura 24 - Fluxo de mensagens entre cliente e servidor cEMI [8]

A figura mostra os dois tipos de mensagens cEMI que existem: mensagens de troca de dados na camada de ligação lógica e mensagens de gestão e configuração do dispositivo KNX/EIB.

2.5.2.1 Mensagens cEMI na camada de ligação lógica

Uma mensagem cEMI da camada de ligação lógica, tem uma estrutura básica constituída por: um octeto inicial que contém o código da mensagem, procedido de um octeto que indica o tamanho do campo que contém informação adicional, os octetos seguintes são a informação adicional (se existir) e a informação que diz respeito ao serviço referido no código da mensagem. Esses serviços são os diferentes tipos de mensagens cEMI. O apêndice 1 contém a tabela com os códigos de mensagem de todo o tipo de mensagens cEMI.

As mensagens cEMI na camada de ligação lógica podem ser de três tipos de serviços: L_DATA, L_POOL_DATA e L_RAW.

2.5.2.1.1 Serviço L_DATA

As mensagens do tipo L_DATA são obrigatórias e utilizadas para enviar os dados (Tramas KNX/EIB). A figura 25 ilustra a estrutura básica de uma mensagem do tipo L_DATA.

Message Code	Additional Info Length	Additional Information	Control field 1	Control field 2	Src. High	Src. Low	Dest. High	Dest. Low	NPDU	
MC	AddIL	...	Ctrl1	Ctrl2	SAH	SAL	DAH	DAL	L	TPCI/APCI & data
1 octet	1 octet	var. length	1 octet	1 octet	2 octets		2 octets		1 octet	var. length

Figura 25 - Estrutura básica de uma mensagem cEMI do tipo L_DATA [8]

Até ao octeto *Control field 1*, é o cabeçalho de uma estrutura cEMI básica. Todos os restantes octetos são respectivos de cada serviço. No serviço L_DATA, o Ctrl1 e Ctrl2 são campos de controlo e cada um tem o seu significado:

Como podemos observar na figura 26, o bit 7 do Ctrl1 representa o tipo de trama KNX/EIB, se é estendida (FT=0) ou normal (FT=1).

O bit 5 indica se a trama pode ser ou não repetida, r=0 para repetição ou r=1 para não repetição.

O bit 4 só é aplicável se o meio físico for aberto como é o caso de redes sem fios. Para o caso de ser *system broadcast*, SB=0 e para o caso de ser apenas *broadcast*, SB=1.

Os bits 2 e 3 indicam a prioridade da mensagem: 00 para prioridade igual à do sistema, 10 para urgente, 01 para normal e 11 para prioridade baixa.

O bit 1 indica se é necessário um *acknowledge* para indicar que a trama L_DATA foi bem entregue. Se for necessário um *acknowledge*, a=1 senão toma o valor de 0.

O bit 0 é utilizado pela função L_DATA.con e serve para confirmar se houve erros ou não. No caso se ocorrer erros, este bit toma o valor de 1.

Ctrl1							
7	6	5	4	3	2	1	0
Frame Type	reserved	repeat flag	system broadcast	priority		acknowledge	Confirm
FT	0	r	SB	p1	p0	a	C

Figura 26 - Control Field 1 do L_DATA [8]

A imagem seguinte ilustra a constituição do Control Field 2:

Ctrl2							
7	6	5	4	3	2	1	0
Address Type	hop count			Extended Frame Format			
AT	h	h	h	f	f	f	f

Figura 27 - Control Field 2 do L_DATA [8]

O bit 7 indica o tipo de endereço KNX/EIB a quem a mensagem se destina. O valor deste bit é 0 para o caso do endereço ser individual e 1 para endereço de grupo.

Os bits 4, 5 e 6 contêm o valor do *hop count* que é decrementado cada vez que passa por um *router*.

Os restantes quatro bits têm a mesma função que o octeto 7 do Ctrl1, indicam se a mensagem é normal ou estendida. Toma o valor de 0000 para mensagem normal e 1xxx para mensagem estendida.

Voltando a analisar a figura 25, depois dos octetos de controlo, seguem-se dois octetos que indicam o endereço KNX/EIB de origem e outros dois que indicam o endereço KNX/EIB de destino. O octeto L contém o número de octetos da trama NPDU. Este tamanho é igual ao já referido no capítulo onde foi explicado o formato de uma trama KNX/EIB. Os restantes octetos, são o que resta de uma trama KNX/EIB deste o octeto que contém o TPCI até ao *check octet*.

Existem 3 tipos de mensagens do serviço L_DATA: L_DATA.req, L_DATA.con e L_DATA.ind.

O L_DATA.req serve para enviar tramas KNX/EIB para o meio físico. Se o servidor cEMI enviou correctamente a trama KNX/EIB para o meio físico, envia uma mensagem L_DATA.con para o cliente cEMI a indicar se a trama foi entregue com erro (C=0) ou com sucesso (C=1).

A mensagem L_DATA.ind é utilizada quando é recebido pelo meio físico uma trama KNX/EIB e o servidor cEMI, passa-a ao cliente cEMI.

2.5.2.1.2 Serviço L_POOL_DATA

Este tipo de serviço é aplicável apenas quando se utiliza o meio físico *Twisted-Pair*. Este serviço é opcional.

Tal como no serviço L_DATA, uma mensagem L_POOL_DATA tem praticamente a mesma estrutura, exceptuando que a partir do endereço KNX/EIB de destino, em vez de conter a NPDU, tem um octeto que define o número de *slots* e para o caso de mensagens L_POOL_DATA.con ainda acrescenta no final um conjunto de octetos que contêm os dados. O número de *slots* indica o tamanho dos dados que a mensagem transporta.

Ao contrário do L_DATA, este serviço apenas tem as mensagens: L_POOL_DATA.req e L_POOL_DATA.con.

2.5.2.1.3 Serviço L_RAW

Este tipo de serviço tal como o anterior é opcional e a sua utilização é para casos especiais, tais como: teste e diagnóstico. Nunca deve ser utilizado quando a rede está a funcionar normalmente, só deve ser utilizado quando a rede se encontra no modo *offline*.

A estrutura de uma mensagem L_RAW é mais simples que as anteriores visto só ter o octeto do código da mensagem, a informação adicional e os dados que contêm a *raw frame*.

Este serviço tem as mensagens: L_RAW.req, L_RAW.con, L_RAW.ind e L_BUSMON.ind.

À excepção da L_BUSMON.ind todas as outras mensagens têm a mesma função que as do serviço L_DATA. O L_BUSMON.ind serve para transportar os *acknowledges* vindos do meio físico que a mensagem L_RAW.ind não pode transportar por motivos de diferenciação de tipos de *raw frames*.

2.5.2.2 Mensagens cEMI de gestão e configuração

A gestão e configuração são feitas localmente no dispositivo KNX/EIB e utilizam-se os seguintes serviços: M_PropRead, M_PropWrite, M_PropInfo e M_Reset.

O serviços de gestão local são confirmados como por exemplo, o M_PropRead.req assim como o M_PropWrite.req, devem ser confirmados se entregues com ou sem sucesso, através dos serviços M_PropRead.con e M_PropWrite.con respectivamente. Mensagens geradas pelo servidor cEMI, como por exemplo: notificações de eventos, utilizam o M_PropInfo.ind que é um serviço que não necessita de confirmação.

A figura 28, ilustra a estrutura básica de uma mensagem cEMI de gestão e configuração.

Message Code	Interface Object Type		Object Instance	Property ID	number of elements	start index	Data
MC	IOTH	IOTL	OI	PID	NoE	SIx	Data
1 octet	2 octets		1 octet	1 octet	4 bits	12 bits	var. length

Figura 28 - Estrutura básica de mensagem cEMI de gestão e configuração[8]

O primeiro octeto contém o código da mensagem. Os códigos de mensagens encontram-se numa tabela do prefácio 1.

Os dois octetos seguintes, indicam o tipo de *Interface Object*. Existem 14 tipos, sendo que o que é mais utilizado é o *Interface Object* tipo 0 que toma o nome de *Device Object*. Os restantes *Interface Objects* são utilizados para outras aplicações que não a de gestão e configuração.

O octeto que tem o nome de *Object Instance*, pode tomar valores de 1 até n e indicam quantas instâncias existem do objecto.

O octeto denominado por PID contém o identificador da propriedade do objecto. No prefácio 1 encontra-se uma tabela com os PID e a respectiva propriedade do *Device Object* a que se refere.

Os dois octetos seguintes, definem o NoE e o Slx, onde o primeiro é formado pelos 4 bits iniciais e os restantes 12 são o Slx. O NoE indica o número de elementos do *array* de uma propriedade, se esta tiver uma estrutura de *arrays*. No caso de a propriedade não for do tipo *array*, o NoE toma o valor de 1. O Slx é o apontador para a posição do primeiro elemento desse *array* e o primeiro elemento do *array* encontra-se na posição 1. Os octetos seguintes, contêm os dados.

2.5.2.2.1 Serviço M_PropWrite

Este serviço serve para alterar valores da propriedades de *Device Objects*. Existem dois tipos de mensagens: M_PropWrite.req e M_PropWrite.con.

O M_PropWrite.req é enviado pelo cliente cEMI para modificar um valor de uma propriedade do objecto que o servidor cEMI conhece. A estrutura desta mensagem é igual à estrutura genérica apresentada anteriormente e no campo dos dados, vai o valor ou valores que serão para escrever na propriedade indicada através do octeto PID.

Como resposta à mensagem anterior, o servidor cEMI de gestão, deve confirmar a operação de escrita através da mensagem M_PropWrite.con.

Quando a resposta é positiva (escrita com sucesso), a mensagem não deverá contar dados, sendo a sua estrutura igual à da mensagem recebida, com os mesmos valores, alterando apenas o código da mensagem e removendo o campo de dados.

Quando ocorre algum erro de escrita ou quando o objecto ou propriedade não existe no servidor cEMI, é gerido uma mensagem de erro e a resposta ao cliente cEMI deve ser enviada utilizando na mesma a mensagem M_PropWrite.con. Ao contrário da situação anterior, o campo de dados é enviado e nesse campo vai o código do erro da operação. O valor do NoE também é alterado para o valor fixo de 0 reforçando a ideia de que no campo de dados apenas vai o código do erro.

2.5.2.2.2 Serviço *M_PropRead*

Este serviço serve para o cliente cEMI efectuar leituras de propriedades de *Device Objects* contidos no servidor cEMI.

Tal como no serviço anterior, existem dois tipos de mensagens: *M_PropRead.req* e *M_PropRead.con*.

A mensagem *M_PropRead.req* é enviada pelo cliente cEMI para ler o valor de uma propriedade do objecto contido no servidor cEMI. A estrutura desta mensagem é igual à estrutura genérica de mensagens de gestão e configuração, exceptuando que o campo de dados não é utilizado.

Em resposta à mensagem anterior, o servidor cEMI envia uma mensagem do tipo *M_PropRead.con* com a resposta positiva ou negativa. No caso do servidor não encontrar nenhum erro na leitura, envia a resposta positiva em que a mensagem é igual à recebida anteriormente (*M_PropRead.req*), mudando apenas o código da mensagem e acrescentado o campo de dados que contém o valor ou valores lidos da propriedade.

No caso do servidor cEMI encontrar um erro na leitura como por exemplo, a propriedade ou objecto não existem, ele envia uma mensagem do tipo *M_PropRead.con*, mas no campo de dados em vez de ir o valor da leitura, vai o código do erro. Para reforçar que no campo de dados vai um código de erro, o NoE toma o valor 0.

2.5.2.2.3 Outros serviços de gestão e configuração cEMI

Como foi referido no início, existem mais dois serviços utilizados pelo cliente e servidor cEMI. Um deles é o *M_ProplInfo* e o outro é o *M_Reset*.

O *M_ProplInfo* utiliza apenas uma mensagem do tipo *M_ProplInfo.ind* que deve ser aplicada pelo servidor cEMI para enviar uma notificação de evento para o cliente cEMI. Pode utilizar esta mensagem por exemplo: para avisar o cliente cEMI de que o valor de uma propriedade foi alterado. A estrutura desta mensagem é igual à estrutura genérica que se encontra na imagem 28.

O serviço *M_Reset* tem dois tipos de mensagens: *M_Reset.req* e *M_Reset.ind*.

A mensagem M_Reset.req deve ser utilizada para reiniciar o servidor cEMI por indicação do cliente cEMI. Esta operação normalmente é feita quando o cliente cEMI inicializa. Esta mensagem é constituída apenas por um octeto que indica o código da mensagem.

Como resposta à mensagem anterior, o servidor cEMI deve enviar a mensagem M_Reset.ind ao cliente, avisando-o de que a reinicialização foi feita com sucesso. Tal como a mensagem anterior, esta também é formada por um só octeto que contém o código da mensagem.

2.6 ETS

O ETS como já foi referido anteriormente, é o software criado e fornecido pela organização KONNEX. Tal como o nome indica: *Engineering Tool Software*, é uma ferramenta para engenharia de sistemas KNX/EIB.

O ETS tem várias utilizações, sendo que a mais utilizada é a de desenho e configuração de uma instalação ou projecto KNX/EIB. Com esta ferramenta é também possível, fazer a gestão da rede KNX/EIB.

Como já foi referido num sub capítulo anterior, um dos modos de configuração: o *S-Mode*, utiliza um computador com o software ETS para configurar a rede KNX/EIB. Este software contém uma base de dados de todos os produtos e que deve ser fornecida por cada fabricante. Essa base dada contém todas as informações sobre cada dispositivo KNX/EIB o que permite saber que funções ou aplicações tem. A pessoa que pretende configurar a instalação KNX/EIB, pode desenhar e configurar toda a rede no modo *offline* e posteriormente carregar toda a configuração para os dispositivos KNX/EIB. Quando ligado ao barramento KNX/EIB, o ETS comporta-se como um Mestre de Configuração.

Até à data, o ETS vai na versão 3.0d e apresenta duas configurações: uma para iniciados e outra para profissionais. A versão apresentada é a 3.0d para profissionais. Esta é a versão mais utilizada pelos instaladores por permitir funções mais avançadas.

A figura 29 ilustra a interface inicial na abertura do programa.

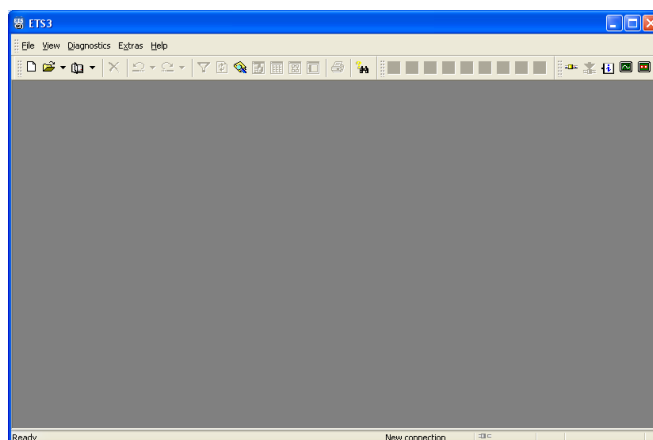


Figura 29 - Interface ETS 3.0d Profissional

No menu *File*, aparecem as funções de abrir ou criar um novo projecto que contém o desenho e configuração da instalação KNX/EIB.

No menu *View*, para além de ser possível configurar as barras e janelas que devem aparecer na interface, é possível seleccionar as bases de dados dos fabricantes e efectuar uma pesquisa de produtos nas bases de dados.

No menu *Diagnostics* aparecem as funções que permitem verificar se a configuração está bem feita e permite efectuar testes aos dispositivos KNX/EIB.

No menu *Extras*, surgem as opções de configurar o tipo de ligação entre o ETS e o barramento KNX/EIB, sendo que é possível utilizar o USB, RS232 e o TCP/IP Ethernet. Tem outras funções tais como: manutenção da base de dados, configurar os botões a aparecerem na barras, verificar as licenças, e a função de colocar o ETS *online* com a rede KNX/EIB.

O software ETS torna-se inútil na falta das bases de dados dos fabricantes. Para poder ter acesso a uma base de dados, foi necessário efectuar o contacto com uma empresa instaladora de sistemas KNX/EIB como é o exemplo da empresa Casa das Lâmpas, revendedora de artigos KNX/EIB da *Albrecht Jung* que forneceram a base de dados dos artigos deste fabricante.

Depois da criação de um novo projecto e da base de dados do fabricante se encontrar inserida no ETS, é possível iniciar a configuração de uma instalação. A figura 30 ilustra a interface obtida quando se pretende procurar e inserir um dispositivo KNX/EIB.

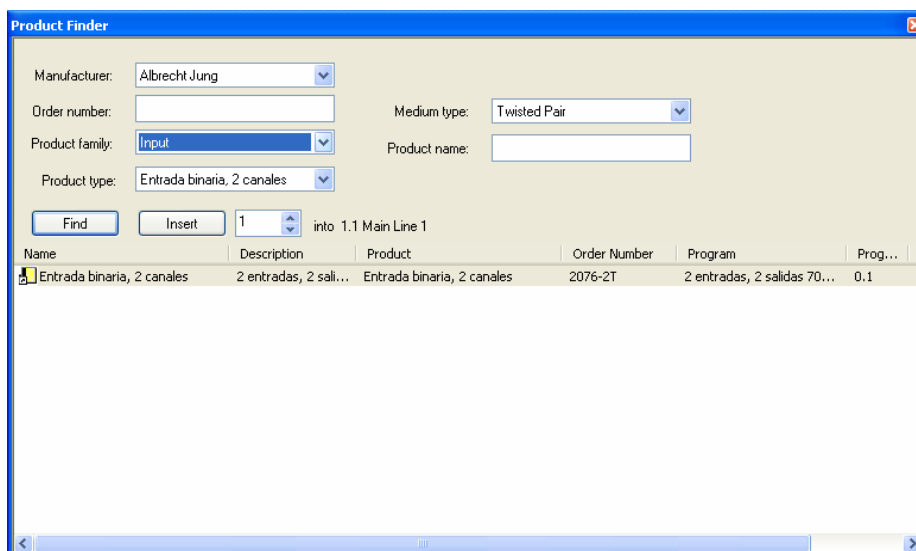


Figura 30 - Interface de pesquisa e inserção de dispositivos KNX/EIB

Nesta janela é possível escolher o fabricante, o tipo de meio físico que a rede utiliza, a família de produtos e o tipo de produto. Depois de escolhidas as categorias pretendidas, efectua-se uma pesquisa seleccionando o botão *Find* e surge uma lista com todos os dispositivos KNX/EIB que tenham as características seleccionadas. Depois de seleccionado o dispositivo, insere-se o mesmo no projecto carregando no botão *Insert*.

A figura 31 ilustra um exemplo de configuração de uma instalação KNX/EIB.

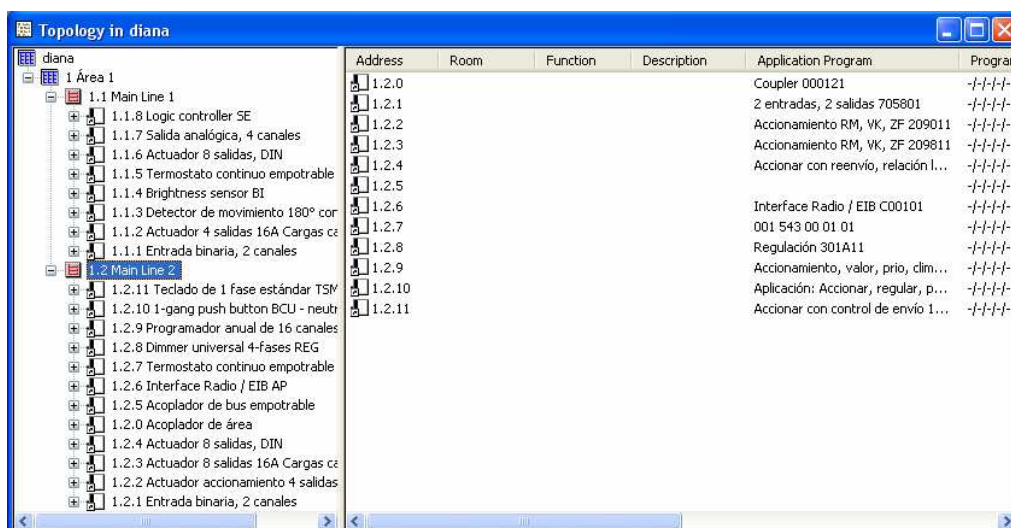


Figura 31 - Interface configuração de uma instalação KNX/EIB

A instalação do exemplo só contém uma área e duas linhas. Na criação do projecto, insere-se ou remove-se áreas e linhas com muita facilidade e os endereços



KNX/EIB são atribuídos automaticamente. Em cada linha, inserem-se os dispositivos KNX/EIB da forma que foi explicada anteriormente. Os endereços dos dispositivos são também atribuídos automaticamente. É possível inserir dispositivos de fabricantes diferentes.

Depois de configurada a instalação, basta seleccionar a opção de *download* e o ETS configura automaticamente cada dispositivo KNX/EIB que esteja na instalação. O instalador apenas tem de ter atenção à selecção dos dispositivos KNX/EIB no projecto para não existir a situação de escolher o dispositivo diferente ao que se encontra na instalação física. A simplicidade deste software, permite a qualquer pessoa ser capaz de configurar uma rede KNX/EIB sem grandes conhecimentos.

3 *KNXnet/IP*

A utilização das redes de telecomunicações e das Tecnologias de Informação e Comunicação, tem vindo a aumentar e o custo desse tipo de redes tem diminuído. Actualmente é usual existir numa habitação comum uma rede IP onde é possível estar ligado à Internet ou até mesmo criar uma PAN (*Personal Area Network*). O custo de uma rede IP é muito reduzido e por essa razão o protocolo Konnex (KNX) permite a sua utilização como meio físico. Para poder utilizar KNX sobre uma rede IP foi criado o protocolo KNXnet/IP.

O KNXnet/IP é uma continuação do EIB.net e esta nova actualização está definida na norma EN 13321-2.

Para uma melhor compreensão, o KNXnet/IP situa-se na camada Física do modelo em camadas do KNX e define a integração do protocolo KNX sobre o protocolo de rede *Internet Protocol* (IP).

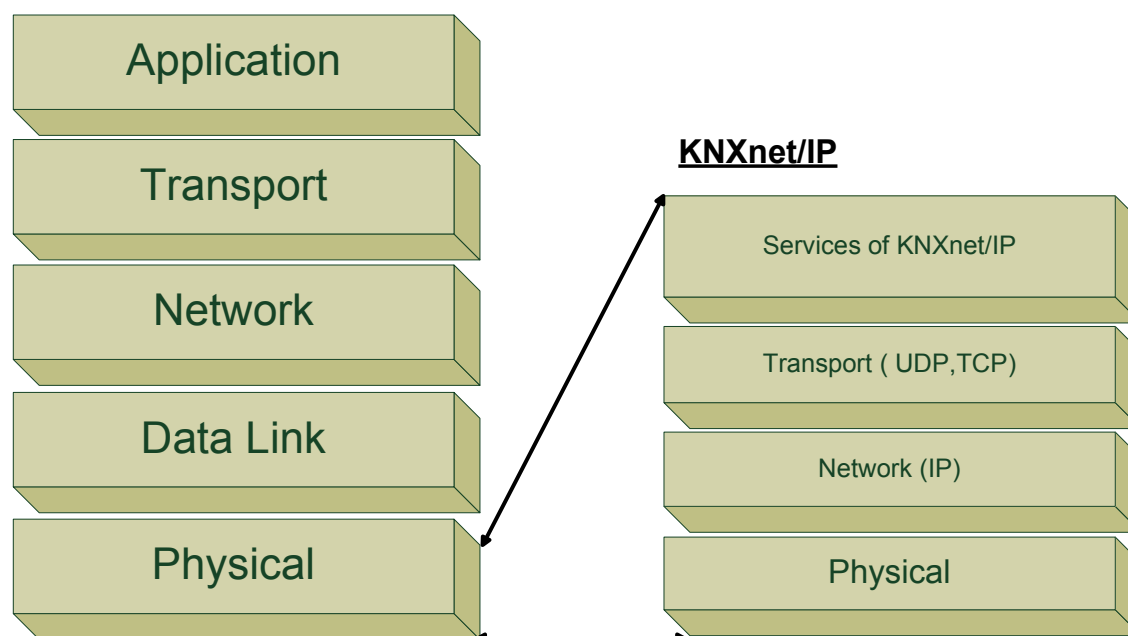


Figura 32 - Modelo em camadas do protocolo KNX e KNXnet/IP

Na figura 32 é compreensível em que camada do protocolo KNX se situa o KNXnet/IP. Os serviços do KNXnet/IP são vários e serão descritos posteriormente. As restantes camadas do KNXnet/IP: transporte, rede e física, são as mesmas que o protocolo TCP/IP utiliza. A especificação do KNXnet/IP apenas descreve em pormenor a camada dos serviços e indica que se pode utilizar o transporte UDP ou o TCP e a rede é a IP.

Existem inúmeras vantagens em utilizar KNX sobre IP:

- permite configurações remotas da rede KNX;
- como a rede IP é de velocidades elevadas em comparação com a velocidade num *backbone* de uma rede KNX (TP1, TP0,...), não provoca atrasos nas comunicações;
- permite a supervisão e controlo dos dispositivos remotamente.

3.1 *Arquitectura*

A figura 33 ilustra um cenário típico de um sistema KNXnet/IP onde um dispositivo KNXnet/IP cliente que executa um software de configuração como por exemplo o ETS, acede via uma rede IP aos múltiplos dispositivos KNX instalados em sub redes KNX. Neste exemplo, as sub redes são em *Twisted-Pair* tipo 1 (KNX-TP1), mas poderão ter outro meio físico que a norma KNX permita. O cliente KNXnet/IP pode aceder a um ou todos os servidores KNXnet/IP ao mesmo tempo. No caso dos servidores implementarem também o *routing*, é possível que os servidores possam comunicar entre si. Mais à frente será explicado o que é o *routing*.

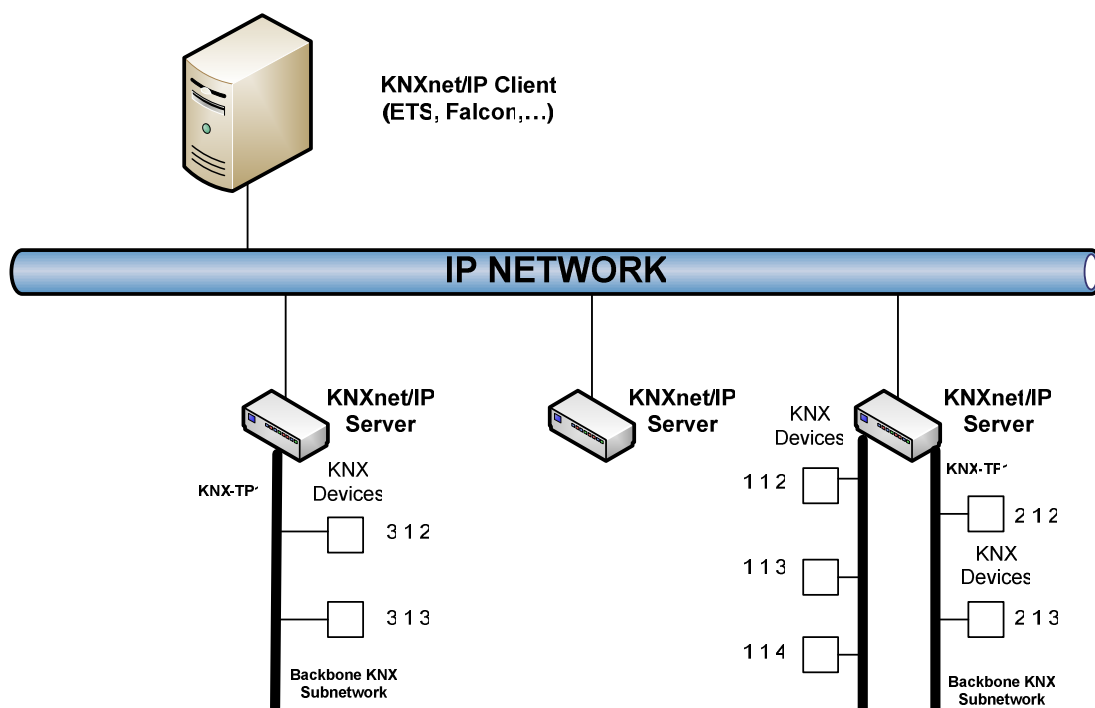


Figura 33 - Exemplo de uma configuração KNXnet/IP possível

Um sistema KNXnet/IP terá de conter no mínimo os seguintes elementos:

- uma sub rede KNX (KNX-TP1, KNX-TP0, KNX-RF,KNX-PL110,KNX-PL132);
- um dispositivo que faça interface entre uma rede KNX e uma rede IP que é denominado por servidor KNXnet/IP
- e adicionalmente poderá ser utilizado software para funções remotas num computador ligado à rede IP.

3.2 Módulos do KNXnet/IP

O KNXnet/IP está organizado por módulos, o que permite o criador de produtos KNXnet/IP implementar apenas o que lhe interessa diminuindo a complexidade e consequentemente o custo.

Os módulos são os seguintes:

- Núcleo;
- Gestão de dispositivos;
- *Tunnelling*;
- *Routing*.

O módulo obrigatório, é o núcleo do protocolo que contém as especificações das tramas KNXnet/IP, os serviços básicos de funcionamento e contém um capítulo descritivo do protocolo de rede IP.

O módulo da gestão de dispositivos apenas define serviços para efectuar configurações e gestão de servidores KNXnet/IP.

O módulo de *tunnelling* consiste em criar um túnel IP, entre um cliente KNXnet/IP e um servidor KNXnet/IP (uma conexão de um-para-um) por onde são trocadas tramas KNX. É essencial para poder utilizar o software ETS e configurar os dispositivos KNX que estão ligados a servidores KNXnet/IP.

O módulo de *routing* é utilizado para efectuar o encaminhamento de tramas pela rede. Este módulo é essencial no caso de se pretender que os servidores KNXnet/IP possam trocar mensagens directamente entre si. Essas tramas podem ter origem do próprio servidor, bem como na rede KNX TP1 ligada a esse servidor. Por exemplo, na figura 33, o dispositivo KNX 1.1.2 para comunicar com dispositivos KNX de outras sub-redes (como é o exemplo do 3.1.2), é necessário que os servidores de interface à rede IP, implementem o *routing*.

Em suma, se for pretendido um servidor KNXnet/IP que possa interligar uma ou mais sub redes a um cliente KNXnet/IP, apenas é necessário implementar o núcleo, gestão de dispositivos e o *tunnelling*. Um exemplo desta arquitectura é a imagem 34.

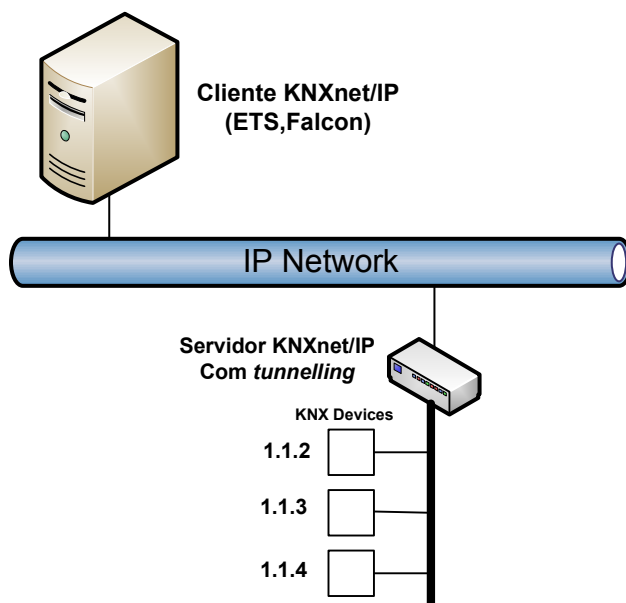


Figura 34 - Exemplo de arquitectura com servidor KNXnet/IP implementando tunnelling

Com esta arquitectura, o software ETS pode configurar um dispositivo KNX como por exemplo: configurar o 1.1.2 utilizando o servidor KNXnet/IP para efectuar a interface entre a rede IP e a rede KNX.

É de salientar que este tipo de servidor KNXnet/IP não pode comunicar com os outros servidores KNXnet/IP, apenas pode comunicar com clientes KNXnet/IP. Esta solução é utilizada quando se pretende conectar o *backbone* de uma rede KNX em TP1 a um cliente KNXnet/IP que poderá ser um computador que gere e monitoriza a rede KNX. Começa a ser usual encontrar servidores KNXnet/IP destes para poder controlar e monitorizar os dispositivos da casa remotamente utilizando a Internet.

Adicionando o módulo de routing a um servidor KNXnet/IP permite que este possa comunicar com outros servidores KNXnet/IP espalhados pela rede IP.

A figura 35 ilustra um exemplo de uma arquitectura com este tipo de servidores.

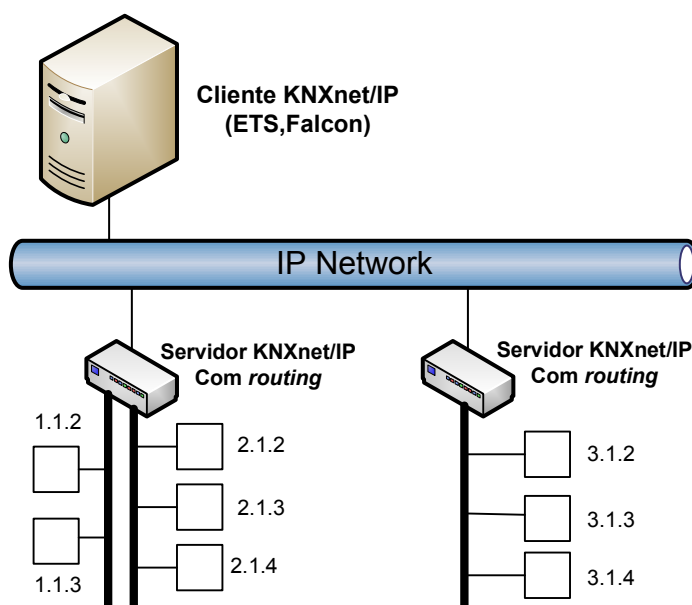


Figura 35 - Exemplo de arquitectura com servidor servidor KNXnet/IP implementando routing

Com esta solução é possível por exemplo: o dispositivo KNX 3.1.2 comunicar com os dispositivos KNX 1.1.2 e 2.1.2.

Desta forma, é possível interligar uma ou mais sub redes KNX através de uma rede IP e assim o *backbone* da rede KNX passa a ser a rede IP. Estes servidores também podem comunicar com os clientes. Este tipo de solução não é tão usual visto que aumenta significativamente o custo do projecto KNX. Uma possível utilização era em edifícios de grandes dimensões e complexidade onde se pretendia interligar todos os dispositivos do edifício através de uma rede IP.

Analisando estas arquitecturas, a solução ideal por ser a mais completa era de criar um servidor KNXnet/IP que implementasse o núcleo, gestão de dispositivos, o *tunnelling* e o *routing*. Por motivos de complexidade e de prazos curtos, decidiu-se implementar apenas o núcleo, gestão de dispositivos e o *tunnelling*.

3.3 Núcleo

A especificação do núcleo, contém uma descrição dos serviços básicos de funcionamento do KNXnet/IP assim como as tramas KNXnet/IP, e os conceitos básicos. Este módulo é obrigatório implementar num dispositivo KNXnet/IP porque contém a camada básica do protocolo KNXnet/IP. É responsável pelos serviços de *discovery* e *self-description* de um servidor.

Especialmente para as redes que suportam hot-plug e onde provavelmente a atribuição do seu endereço é em runtime (ex: DHCP), é importante que haja uma procura dos dispositivos nas sub redes. Por esta razão, foi criado o serviço *discovery* e depois o *self-description* para saber informações mais pormenorizadas do dispositivo.

O núcleo introduz alguns conceitos como o *endpoint*, o *service container* e o HPAI.

O *endpoint* é uma abstracção de uma entrada/saída de mensagens KNXnet/IP e KNX. Existem os *endpoint*: de *discovery*, controlo e dados. Cada servidor de KNXnet/IP tem de ter obrigatoriamente um *endpoint* de *discovery* e poderá ter vários *endpoints* de dados e controlo. Os *endpoints* de dados que existem são: o *endpoint* configuração de dispositivos e o *endpoint* de encaminhamento de dados que serão explicados posteriormente.

O conjunto dos dois *endpoints* de dados e um *endpoint* de controlo, denomina-se por *service container*. É de referenciar que para cada ligação a uma sub rede KNX é necessário um *service container* para cada ligação pois estes não podem ser partilhados.

Um *Host Protocol Address Information* (HPAI), está associado a um *endpoint* e contém um endereço IP, porta de comunicação IP e o protocolo de transporte (TCP, UDP). Cada *endpoint* tem de ter um HPAI para fornecer aos dispositivos KNXnet/IP o seu endereçamento. Por exemplo: para o *endpoint* de dados é possível utilizar o protocolo UDP com a porta IP 5001 e com o endereço IP 192.168.105.28. Desta forma, todas as mensagens enviadas pela rede IP como destino o endereço IP 192.168.105.28 e a porta UDP número 5001, serão direccionadas para o *endpoint* de dados.

Um canal de comunicação é uma conexão realizada entre o *endpoint* de dados do cliente KNXnet/IP e o *endpoint* de dados do servidor KNXnet/IP.

Para estabelecer e manter um canal de comunicação, um servidor deve suportar pelo menos um *endpoint* de controlo e um *endpoint* de dados. A figura 36 ilustra a estrutura básica de um servidor KNXnet/IP e a função do núcleo.

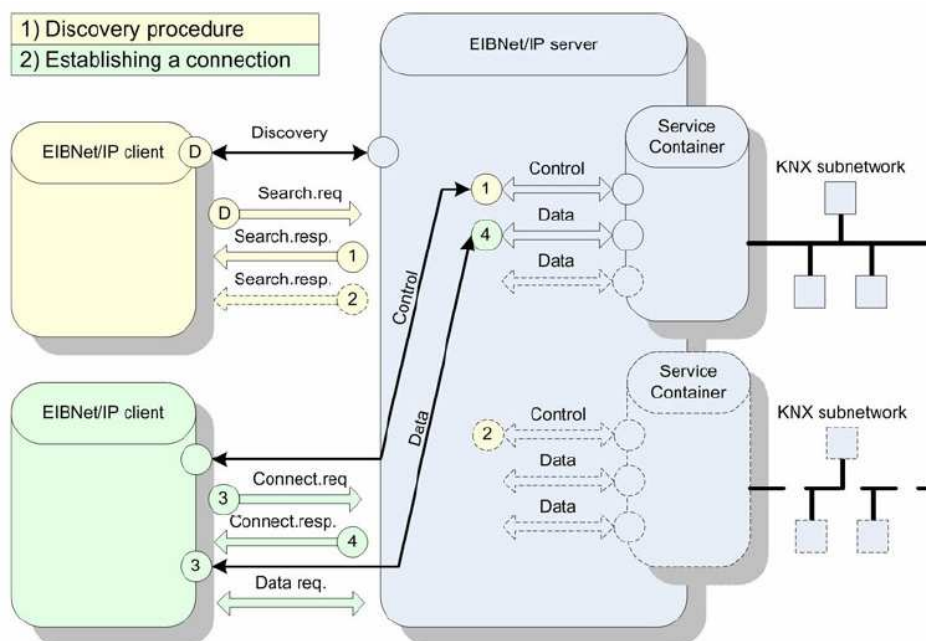


Figura 36 - Esquema de um servidor KNXnet/IP e a função do núcleo [8]

Para o *discovery* de um servidor KNXnet/IP, o cliente envia um `SEARCH_REQUEST` para o *discovery endpoint* dos servidores KNXnet/IP, com o endereço IP de configuração de sistema que existe por defeito em todos os dispositivos KNXnet/IP. Esse endereço é de multicast com o IP 224.0.23.12, a porta IP utilizada por defeito é a 3671 e o protocolo de transporte utilizado é o UDP. Nesta mensagem o cliente envia o HPAI do seu endpoint de controlo para poder receber as respostas ao pedido. Cada servidor KNXnet/IP terá de responder imediatamente com um `SEARCH_RESPONSE`, para o *endpoint* de controlo do cliente contendo o protocolo suportado, potencialidades, informação do estado e opcionalmente o nome desse servidor. Nesta mensagem o servidor deve incluir o HPAI do seu *endpoint* de controlo para comunicações futuras. A figura 36 contém um exemplo deste serviço onde se identificam as trocas das mensagens.

Para estabelecer uma conexão, o cliente envia um `CONNECT_REQUEST` para o *endpoint* de controlo servidor KNXnet/IP. Esta mensagem inclui o HPAI do

endpoint de dados que o cliente deseja utilizar para criar o túnel. O servidor deverá responder dentro do tempo de *timeout* com um `CONNECT_RESPONSE` contendo o HPAI do *endpoint* de dados para esta conexão. A figura 36 contém um exemplo de um estabelecimento de conexão.

3.4 Gestão de dispositivos

O módulo de gestão de dispositivos é responsável pela configuração e gestão do dispositivo KNXnet/IP. Isto poderá ser feito via rede IP ou KNX.

Para a configuração dos dispositivos é necessário estabelecer uma ligação ponto-a-ponto. Para isso, utiliza-se o *endpoint* de controlo para estabelecer a ligação (como foi referido na descrição do núcleo) e para o canal de comunicação utiliza-se o *endpoint* de dados de configuração de dispositivos. Um exemplo de utilização deste módulo é a troca de informação de configurações com o software ETS.

A figura 37 ilustra os *endpoints* que o servidor KNXnet/IP pode ter, evidenciando os *endpoints* de dados. O *endpoint* de dados de *routing* será explicado posteriormente.

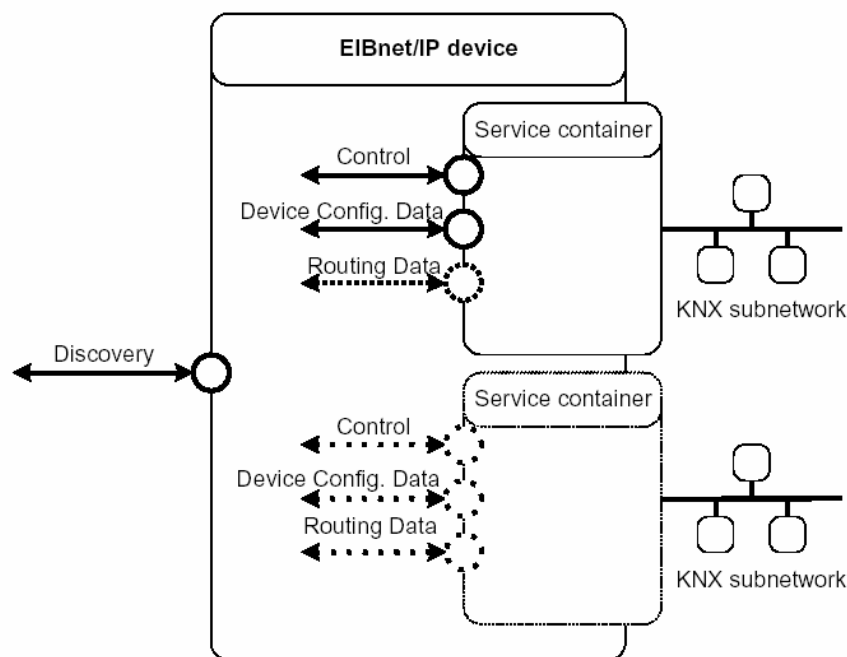


Figura 37 - Endpoints de um servidor KNXnet/IP [8]

O procedimento para a troca de informação deste módulo é o seguinte: o cliente envia uma trama com o serviço `DEVICE_CONFIGURATION_REQUEST` para o HPAI do *endpoint* de dados do servidor. Nessa mensagem vai o HPAI do *endpoint* de dados do cliente. O servidor processa o pedido e responde com um `DEVICE_CONFIGURATION_ACKNOWLEDGEMENT` num prazo de 10 segundos. A mensagem é enviada para o HPAI do *endpoint* de dados do cliente contendo o HPAI do *endpoint* de dados do servidor.

A configuração e gestão utilizam *Interface Objects* do KNX para alterar os valores de variáveis de dispositivos KNXnet/IP e dispositivos KNX. Cada *service container* tem possuir os seus próprios *Interface Objects* e no caso de haver mais do que um *service container*, cada um deles tem de ter os seus próprios *Interface Objects* e endereços IP distintos.

Este módulo de gestão de dispositivos altera valores dos vários *Interface Objects*. A esses valores, utiliza-se o termo de *Property Identifiers* (PID). Exemplos de PID: é a propriedade que contém o endereço MAC do dispositivo KNXnet/IP, o endereço IP, endereço de multicast, quantas mensagens foram enviadas com sucesso para uma rede KNX ou rede IP, etc.

3.5 Tunnelling

O *tunnelling* descreve uma ligação de dados ponto-a-ponto numa rede IP e serve para configurar ou para trocar informações sobre os dispositivos de uma rede KNX. Um cliente KNXnet/IP que implementa o *tunnelling*, envia uma mensagem cEMI através do túnel para um servidor que implemente o *tunnelling* e este passará a mensagem cEMI à rede KNX.

É de notar que o KNXnet/IP não prevê atrasos nas entregas das tramas causadas pelo tráfego na rede IP e consequentemente o tempo de transferência terá de ser inferior ao tempo de *timeout* do protocolo KNX para transferência de tramas.

O *tunnelling* utiliza o *endpoint* de controlo para estabelecer um canal de comunicação como foi descrito no sub capítulo do núcleo e o *endpoint* de dados para a comunicação.

Existem três serviços diferentes para o *tunnelling*:

- *Tunnelling* na camada de *data link* do KNX: Cada dispositivo KNXnet/IP tem de suportar este modo obrigatoriamente. Depois de estabelecida uma conexão com o CONNECT_REQUEST, o servidor KNXnet/IP atribui um endereço de KNX individual para ele próprio e passa-o ao cliente na trama de CONNECT_RESPONSE. Este endereço pode ser configurado e armazenado na propriedade PID_ADDITIONAL_ADDRESSES.

Depois de criado o túnel, o servidor KNXnet/IP envia todas as mensagens KNX ou IP que receber, com destino algum dos seus endereços de grupo, num TUNNELLING_REQUEST para o cliente. Envia também mensagens KNX ou IP que receba para o seu próprio endereço. Por exemplo: quando o servidor KNXnet/IP recebe uma mensagem pela rede KNX onde o endereço de destino é um endereço de grupo KNX a que ele pertença, este envia a mesma mensagem para a rede IP. Se o servidor KNXnet/IP receber uma mensagem pela rede IP com o endereço IP de destino igual ao dele, retira da mensagem IP a mensagem KNX e envia-a para a rede KNX.

Para mais do que uma conexão *tunnelling*, são necessários endereços KNX individuais adicionais que também serão armazenados na propriedade PID_ADDITIONAL_ADDRESSES. O servidor gera igualmente tramas IACK para estes endereços adicionais.

Se o cliente KNXnet/IP envia uma trama cEMI com o endereço KNX de origem em 0 decimal, o servidor substitui o endereço pelo seu endereço individual e passa a mensagem para a rede KNX. Se o endereço de origem na trama cEMI é diferente de 0 decimal, o servidor envia a trama sem a modificar.

- *Tunnelling* sobre um cEMI puro: A implementação deste serviço é opcional. Neste modo, o servidor passa qualquer mensagem KNX recebida directamente para o cliente conectado. Além disso, não gera mensagens de IACK.

- *Tunnelling* sobre KNX *busmonitor*: A implementação é opcional. Activando o modo KNX busmonitor poderá desactivar outro serviço KNXnet/IP para uma sub rede KNX. Este modo não deve ser suportado por um dispositivo que implemente o *routing*.

Para este projecto, apenas será implementado o serviço de *tunnelling* na camada de *data link* do KNX.

3.6 *Routing*

O *routing* é utilizado para encaminhar mensagens entre dispositivos de redes KNX sobre a rede IP de alta velocidade. Um *router* KNXnet/IP envia uma mensagem UDP/IP multicast para que outros *routers* KNXnet/IP na mesma rede a possam receber. Os *routers* podem substituir as tradicionais linhas KNX e os acopladores de *backbone*. É necessário ter cuidado quando se atribui o endereço KNX individual a *routers* para garantir um encaminhamento correcto das tramas KNX e o software ETS foi equipado com mecanismos de garantir as atribuições de endereços correctamente.

Um *router* KNXnet/IP não necessita de estabelecer um canal de comunicação porque o envio das mensagens para a rede IP utiliza endereços *multicast*. Este módulo utiliza somente o *endpoint* de dados de *routing*. Por defeito, todos os *routers* KNXnet/IP são configurados com o endereço *multicast* de *setup* de sistema referido no sub capítulo do módulo núcleo. O endereço *multicast* é o IP 224.0.23.12 e a porta IP é a 3671, registada para este propósito, na *Internet Authority for Number Assignment* (IANA). [16]

É de salientar que se existirem duas instalações KNX diferentes utilizam o mesmo *backbone* da rede IP, terão de utilizar diferentes endereços de *multicast* para os seus *routers* KNXnet/IP.

É possível também controlar por quantos *routers* as mensagens podem passar utilizando o TTL (*Time To Live*) com o valor desejado.

Podem ocorrer situações de *overflow* no *router* porque devido à velocidade de transmissão de dados numa rede IP ser superior à de uma rede KNX, o *router* não consegue enviar para a rede KNX as tramas todas à medida que vão chegando pela rede IP. Nesta situação, os *routers* KNXnet/IP devem enviar uma mensagem de ROUTING_LOST_MESSAGE para a rede IP evitando continuar a receber mais tramas sem ter entregue as outras já recebidas.

3.7 *Resumo dos serviços do KNXnet/IP*

Cada módulo contém um conjunto de serviços e estes serão resumidos nesta secção.

Serviços do núcleo:

- **Search request:** Serve para efectuar uma pesquisa de dispositivos KNXnet/IP na rede. É enviado por um cliente KNXnet/IP via multicast para os *endpoints* de *discovery* de todos os servidores KNXnet/IP à escuta. O cliente KNXnet/IP tem de incluir na mensagem de pedido o HPAI do seu *endpoint* de controlo.

- **Search response:** Serve para responder a uma pedido de *search request*. O servidor KNXnet/IP responde para o endereço unicast que recebeu do cliente KNXnet/IP. Este endereço unicast foi fornecido através do HPAI do *endpoint* de controlo do cliente KNXnet/IP. Nesta mensagem de resposta, o servidor KNXnet/IP inclui o HPAI do seu *endpoint* de controlo assim como uma descrição do hardware e dos serviços que suporta.

- **Description request:** É utilizado para pedir uma descrição das capacidades e serviços suportados por um servidor KNXnet/IP. Este serviço é utilizado pelo cliente KNXnet/IP que envia o pedido para o *endpoint* de controlo do servidor KNXnet/IP. A mensagem deve incluir o HPAI do *endpoint* de controlo do cliente KNXnet/IP.

- **Description response:** Serve para o servidor KNXnet/IP responder a um pedido de *description request*. A resposta é enviada para o endereço unicast contido no HPAI do *endpoint* de controlo do cliente KNXnet/IP. Esta resposta contém vários *Description Information Blocks (DIBs)*. O DIB fornece informação sobre o meio físico da rede KNX, o estado do dispositivo, o endereço físico e unicast, o número do projecto de instalação, o número de série, o endereço de multicast de *routing*, endereço MAC e o nome do servidor KNXnet/IP.

- **Connect request:** Serve para enviar um pedido de estabelecimento de um canal de comunicação. É enviado pelo cliente KNXnet/IP para o *endpoint* de controlo do servidor KNXnet/IP. Esta mensagem inclui o HPAI do *endpoint* de controlo e o

HPAI do *endpoint* de dados do cliente KNXnet/IP. Além disso, contém *Connection Request Information (CRI)* adicionais, que indica o tipo ligação que se pretende (*Tunneling*, gestão de dispositivos,...).

- **Connect response:** Este serviço é enviado pelo servidor KNXnet/IP como resposta a um pedido de ligação. A mensagem é enviada para o *endpoint* de controlo do cliente KNXnet/IP. Se o *endpoint* de dados estiver disponível o servidor KNXnet/IP responde com uma mensagem contendo o identificador do canal de comunicação, informação do estado, o HPAI do seu *endpoint* de dados e um *Connection Response Data Block (CRD)*. O CRD é igual a um CRI.

- **Connection-state request:** É enviado pelo cliente KNXnet/IP para o *endpoint* de controlo do servidor KNXnet/IP. Contém o identificador do canal de comunicação e o HPAI do *endpoint* de controlo. Serve para saber o estado da ligação.

- **Connection-state response:** O servidor KNXnet/IP responde ao pedido anterior com o identificador do canal de comunicação assim como o código de estado: sem erro, identificador não encontrado, ligação de dados com erro e ligação KNX com erros. A mensagem é enviada para o *endpoint* de controlo do cliente KNXnet/IP.

- **Disconnect request:** Este serviço é enviado pelo dispositivo KNXnet/IP, normalmente pelo cliente KNXnet/IP, e serve para terminar uma ligação. Esta mensagem é enviada para o *endpoint* de controlo do dispositivo que irá receber a mensagem. A mensagem inclui o HPAI do *endpoint* de controlo de quem envia a mensagem.

- **Disconnect response:** Este serviço é enviado pelo dispositivo KNXnet/IP, normalmente pelo servidor KNXnet/IP, e serve para confirmar um pedido de *disconnect request* enviando a mensagem para o *endpoint* de controlo de quem efectuou o *disconnect request*.

Serviços da gestão de dispositivos:

- **Device configuration request:** O cliente KNXnet/IP, efectua um pedido de leitura/escrita de um valor de variável de um dispositivo KNXnet/IP. A mensagem é enviada para o *endpoint* de dados do servidor KNXnet/IP, que este reservou para dados de gestão de dispositivos.

- **Device configuration acknowledge:** Este serviço é a resposta ao pedido anterior. É enviado por um dispositivo KNXnet/IP para confirmar a recepção do pedido. Esta mensagem é enviada para o *endpoint* de dados do destinatário.

Serviços do tunnelling:

- **Tunnelling request:** Utilizado para enviar e receber uma mensagem KNX entre um cliente KNXnet/IP e um servidor KNXnet/IP. A mensagem é enviada para o *endpoint* de dados do destinatário.

- **Tunnelling acknowledge:** Este serviço é a resposta ao pedido anterior. É enviado por um dispositivo KNXnet/IP para confirmar a recepção *tunnelling request*. A mensagem é enviada para o *endpoint* de dados do destinatário.

Serviços do routing:

- **Routing indication:** Utilizado para enviar mensagens KNX sobre uma rede IP. Este serviço não necessita de confirmação. A mensagem é enviada para o *endpoint* de dados de encaminhamento do(s) destinatário(s).

- **Routing lost message:** Utilizado para indicação de uma mensagem de encaminhamento KNXnet/IP perdida. Este serviço não necessita de confirmação. A mensagem é enviada para o *endpoint* de dados de encaminhamento do(s) destinatário(s).

3.8 Formato das tramas KNXnet/IP

Todas as tramas KNXnet/IP contém um cabeçalho que é comum onde o primeiro octeto tem o valor fixo de 6 em decimal e representa o tamanho do cabeçalho. O segundo octeto representa a versão do protocolo KNXnet/IP que neste momento é a versão 1.0. Os seguintes dois octetos descrevem o tipo de serviço e os últimos dois octetos do cabeçalho são o tamanho do mesmo, mais o tamanho da mensagem.

A figura 38 ilustra o formato das tramas KNXnet/IP e os serviços que o protocolo KNXnet/IP fornece.

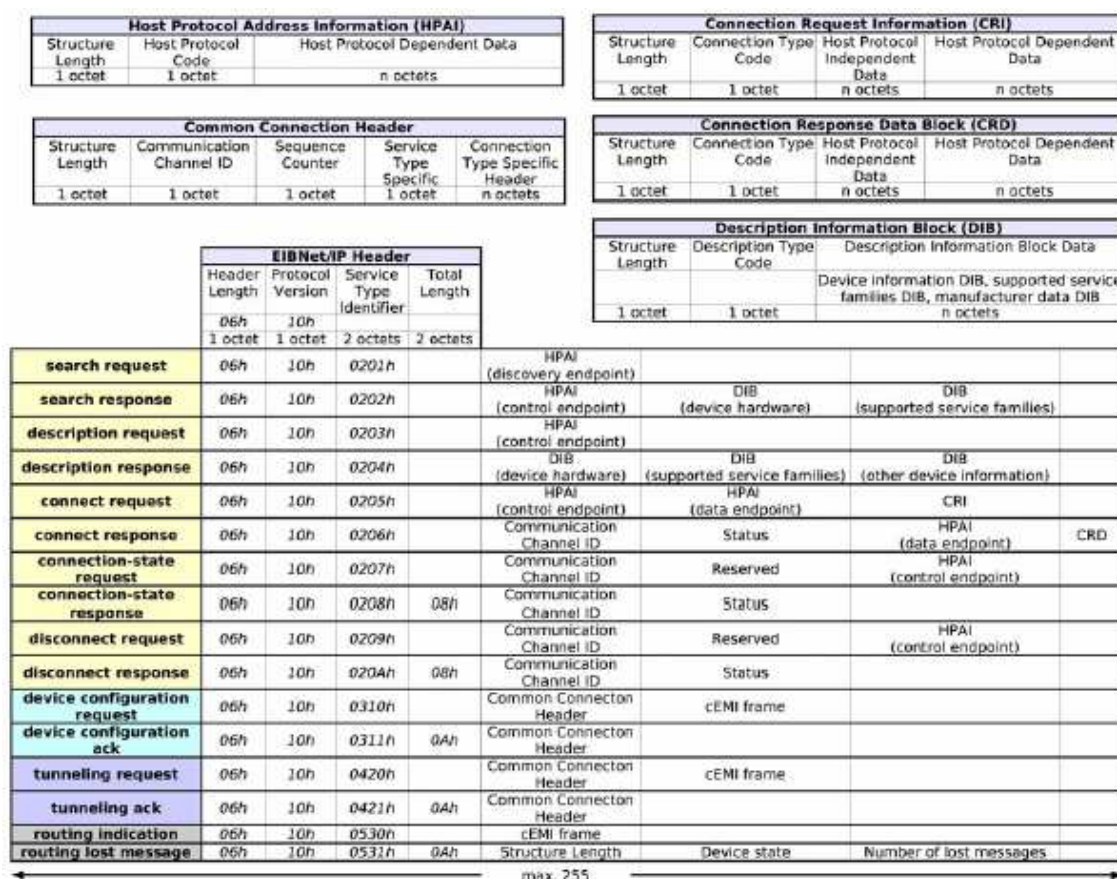


Figura 38 - Formato das tramas KNXnet/IP [8]

Esta figura tem também uma tabela com a estrutura do HPAI, CRI, CRD e DIB que são necessários em alguns serviços para alguns módulos.

4 *Análise da arquitectura*

No mercado, os dispositivos KNX/EIB possuem na sua maioria uma interface para redes KNX/EIB *twisted pair*. No entanto, os dispositivos KNX mais recentes permitem a utilização das redes TCP/IP, recorrendo a um *router* KNXnet/IP que para além de conter o protocolo KNXnet/IP, efectua a interface entre o meio físico TP e a *Ethernet*. Visto estes *routers* serem extremamente caros apenas são utilizados quando é pretendido ligar redes KNX/EIB mais distantes ou então quando é necessário poder controlar a rede KNX/EIB através da *Internet*. A utilização destes *routers* tem vindo a aumentar e o futuro do KNX/EIB é a utilização crescente das redes TCP/IP como meio físico. Os dispositivos KNX/EIB poderiam conter eles próprios a interface para rede TCP/IP, mas ficariam ainda mais caros, porque exigiriam mais poder de processamento para conter o protocolo KNXnet/IP e era necessário acrescentar a interface *Ethernet*. Tendo em conta isto, o estudo da solução foi feito para a utilização redes KNX/EIB sob o meio físico Ethernet, recorrendo ao protocolo de rede TCP/IP devido à sua grande utilização.

Um dos principais requisitos deste trabalho é desenvolver um sistema eficiente sem que isso se traduza num elevado preço do produto final. Dado que o preço final do produto é em grande parte influenciado pelo custo dos componentes usados na parte do hardware, é essencial no momento de desenvolvimento ter em conta a utilização de componentes *standards* e de fácil obtenção no mercado uma vez que estes se podem adquirir a preços mais competitivos. Só assim é possível competir com o produto no mercado, podendo assegurar qualidade e preços baixos.

A escolha da arquitectura é igualmente importante de forma a escolher a melhor configuração que necessita de menos recursos de hardware e menos dispendiosos.

Foram analisadas duas arquitecturas possíveis que serão descritas nos sub capítulos seguintes.

4.1 *Arquitectura Domoitech com KNXnet/IP*

Nesta arquitectura, aproveita-se a estrutura de hardware do projecto final de curso “Domoitech”, efectuando algumas alterações mínimas. É acrescentado ao software do “Domoitech”, a pilha do KNXnet/IP. O esquema da arquitectura é o seguinte:

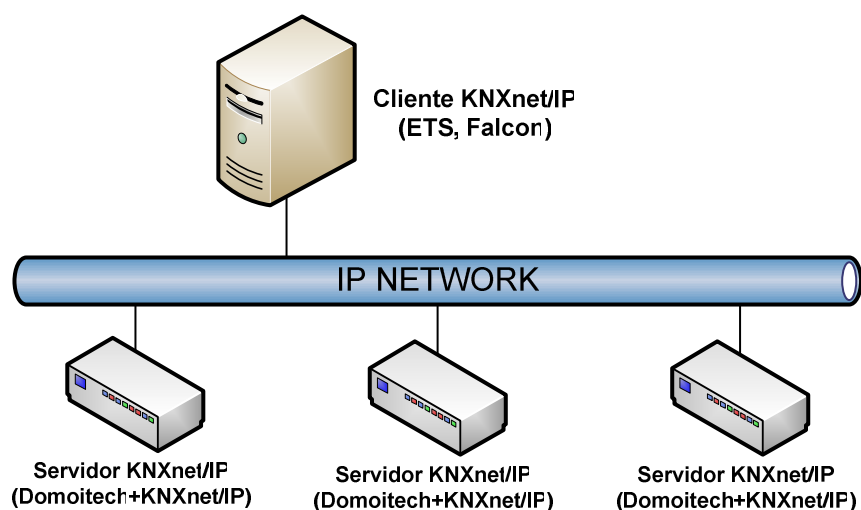


Figura 39 - Arquitectura Domoitech com KNXnet/IP

Esta arquitectura aproveita o projecto de hardware do “Domoitech”, fazendo apenas uns melhoramentos no hardware e acrescentar a pilha KNXnet/IP ao software. Com esta arquitectura é possível comunicar com outros dispositivos KNXnet/IP existentes no mercado.

4.1.1 *Projecto de Hardware*

O projecto de hardware, requer uma análise dos requisitos do software para que a escolha dos componentes seja optimizada.

Aproveita-se a arquitectura de hardware desenvolvido no projecto final de curso “Domoitech” e alteram-se apenas dois componentes essenciais. Como os requisitos do protocolo KNXnet/IP exigem alguns recursos que a XPORT [10] não possui, é necessário adquirir outra versão superior da XPORT com a denominação de XPORT AR [10]. Esta nova XPORT vem com mais duas portas séries tendo um total de três

portas séries: uma para configuração da XPORT através de RS232 e duas para utilizar como canais de comunicação para a rede IP. Para poder utilizar os dois canais de comunicação da XPORT AR, é necessário um microcontrolador com duas portas série. Analisando os microcontroladores PIC [11] com os recursos computacionais necessários, contendo duas portas séries e o preço, o escolhido foi o PIC18F6520 [11]. As grandes diferenças entre este PIC e o utilizado no projecto “Domoitech” são:

- No nível físico o PIC18F6520 é mais pequeno porque tem o encapsulamento TQFP;
- a memória de programa é igual mas a memória de dados é maior no PIC18F6520 sendo de 2048 bytes contra os 768 bytes do PIC18F4520 [11] de SRAM e 1024 bytes contra 256 bytes de EEPROM;
- o PIC18F6520 tem duas portas série e o PIC18F4520 tem apenas uma porta série;
- o PIC18F6520 tem 52 entradas/saídas e o PIC18F4520 tem 25 entradas/saídas.

Apesar do microcontrolador PIC18F6520 ter dimensões inferiores ao PIC18F4520, não vai diminuir o tamanho da placa que o alberga porque a XPORT AR é maior que a XPORT normal. Outra vantagem de utilizar o PIC18F6520 é que irá compensar a diferença de tamanho da XPORT AR para a XPORT de forma a não aumentar o tamanho da PCB.

Mantendo a arquitectura do projecto “Domoitech”, apenas se efectua alterações na interface entre o microcontrolador PIC18F6520 e a rede IP. A figura 40 ilustra o esquema genérico da ligação do microcontrolador à rede IP utilizando a XPORT AR.

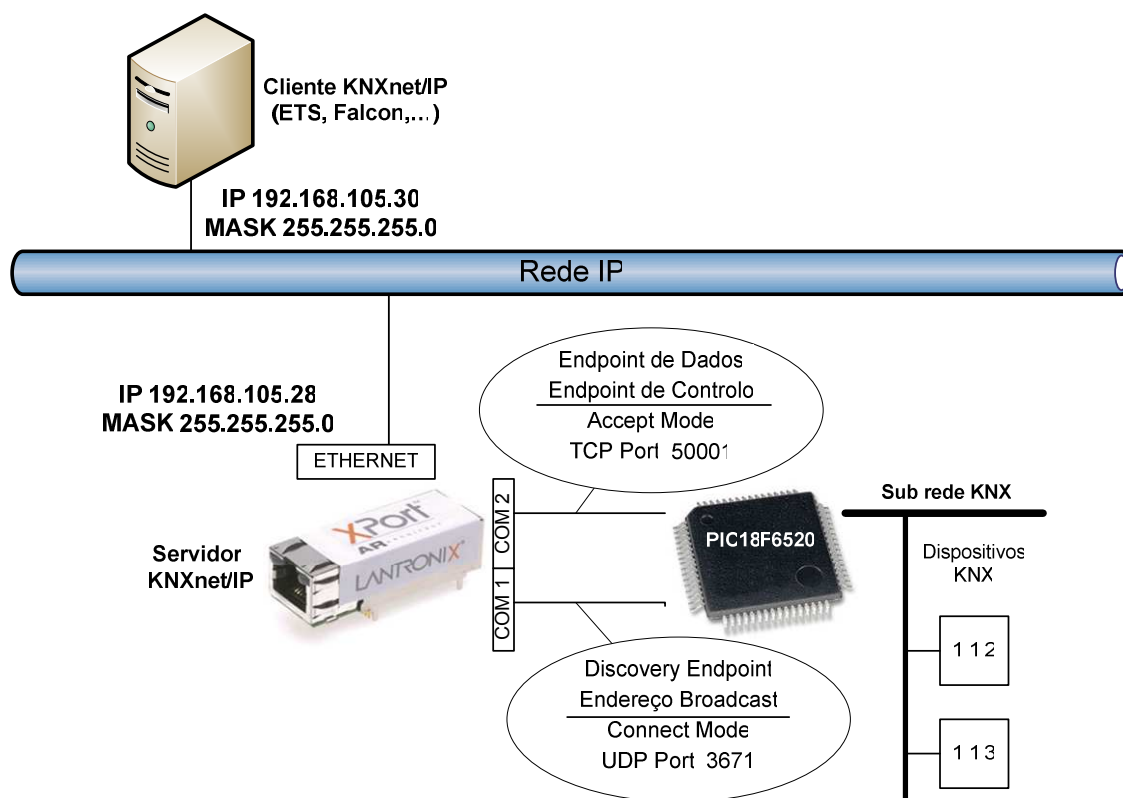


Figura 40 - Esquema do hardware evidenciando as ligações do PIC18F6520 e da XPORT AR

A figura anterior evidencia as ligações do microcontrolador PIC18F6520 e da XPORT AR. Interpretando a figura: o PIC faz a interface entre uma sub rede KNX e a XPORT AR. É de salientar que a sub rede KNX é virtual e o PIC liga directamente aos dispositivos sensores/actuadores.

O PIC não realiza a pilha TCP/IP por isso apenas codifica a trama KNXnet/IP até à camada de serviços do KNXnet/IP. Quem acrescenta à trama KNXnet/IP o cabeçalho de transporte e as restantes camadas inferiores do TCP/IP, é a XPORT AR. O microcontrolador PIC passa e recebe tramas de e para a XPORT AR através de uma das portas série utilizando o protocolo RS 232.

A porta série número 1 da XPORT AR é configurada no *Connect Mode* mas sem endereço IP, com o porto 3671, e UDP. Como não tem um endereço IP configurado, recebe tramas de qualquer dispositivo. Esta porta série tem a funcionalidade de receber tramas *broadcast* para o porto 3671 sobre o protocolo

UDP. Recebe e envia tramas de e para o *discovery endpoint*. É de referir que os *endpoints* são criados e geridos pelo microcontrolador PIC. A XPORT apenas direcciona as mensagens para os *endpoints* existentes no microcontrolador.

A porta série número 2 da XPORT AR é configurada no *Accept Mode* para utilizar ligações TCP. Não é necessário configurar nenhum endereço IP remoto visto que a XPORT AR funciona como servidor a aguardar pedidos de ligações. Apenas é necessário escolher a porta TCP/IP pelo qual irá receber todos os pedidos de ligação, bem como a criação e manutenção de uma ligação TCP. A porta IP escolhida foi a 50001 por se de fácil memorização e porque está excluída da lista de portas registadas na *Internet Authority for Number Assignment (IANA)*. [16] Esta porta série recebe e envia mensagens de e para o *endpoint* de controlo e os *endpoints* de dados. Surgiu a necessidade de utilizar estas duas portas séries porque eram necessários dois tipos de endereços IP: de *unicast* e *multicast*. Como a XPORT AR não suporta *multicast* (nem outro dispositivo da mesma empresa), utilizou-se a porta série 1 para o endereço de *broadcast*. A porta série 2 ficou reservada para receber tramas em *unicast*. Como só *discovery endpoint* utiliza o endereço *multicast*, só ele ficou atribuído à porta série 1. Os restantes *endpoints* utilizam endereços *unicast* por isso podem utilizar a mesma porta série 2.

A figura 41 ilustra a arquitectura de hardware que é praticamente igual à do projecto “Domoitech”, diferenciando apenas a ligação da XPORT AR ao microcontrolador que é feita através de duas portas série.

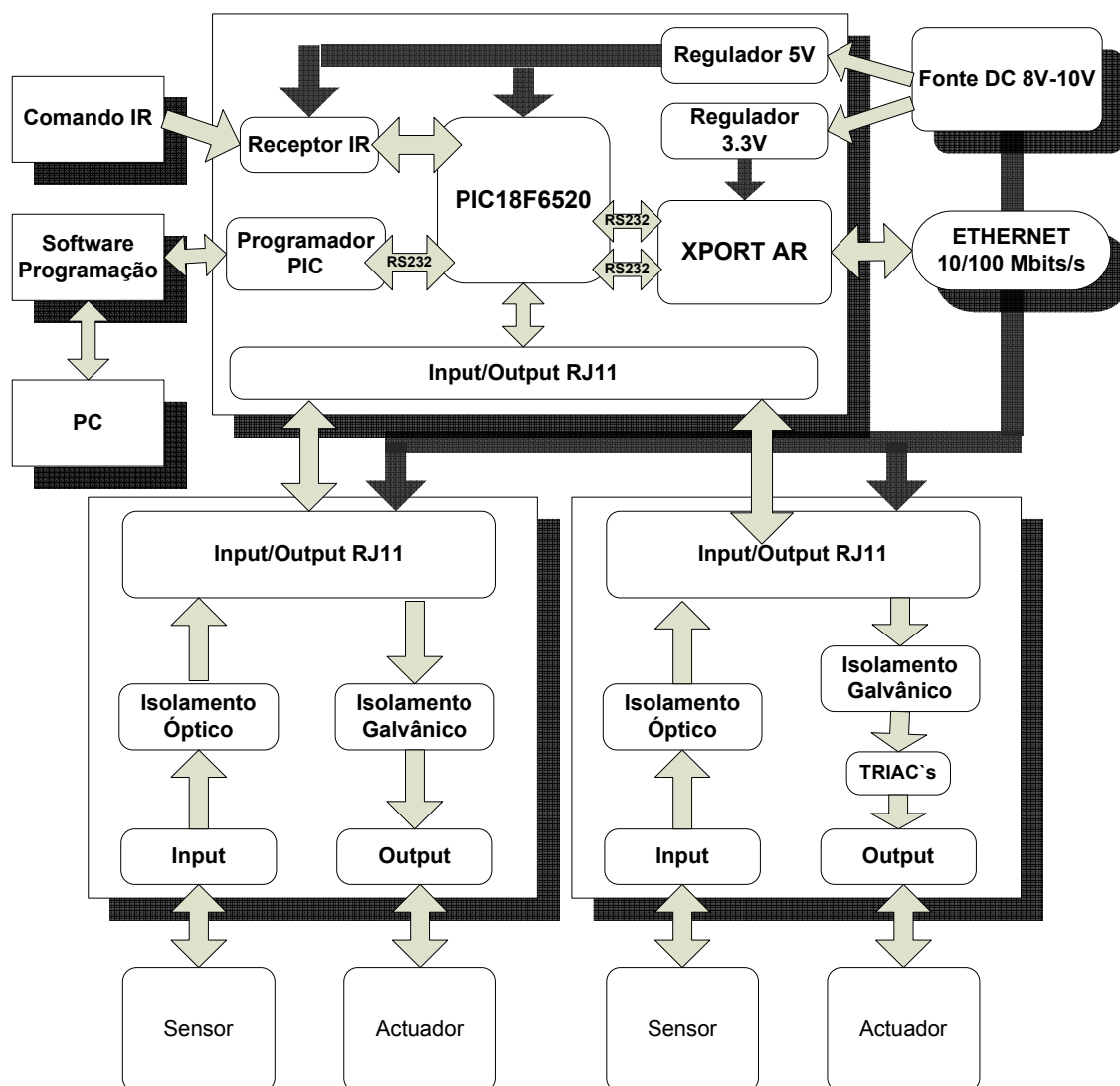


Figura 41 - Diagrama de blocos da estrutura de hardware

Nesta arquitectura existe uma placa principal, que contém todo o processamento necessário e os vários módulos de interface com os sistemas exteriores. Existem ainda dois tipos de placas secundárias, diferenciando-se pelas suas saídas, em que uma tem interface de saída utilizando relés e a outra tem interface de saída utilizando TRIAC's.

A programação do microcontrolador é feita *on-board*, recorrendo a um circuito de interface entre o microcontrolador e a porta série do PC. Aconselha-se a utilização de uma fonte DC com uma tensão entre 8V e 10V. Pode-se utilizar uma fonte DC de 12V, mas existirá mais energia dissipada nos reguladores de tensão e logo o consumo é maior. O microcontrolador tem um receptor de infravermelhos que

permite receber comandos por aparelhos externos de infravermelhos (ex: comando de tv). Os blocos representados por sensores e actuadores podem ser por exemplo: lâmpadas e motores eléctricos para os actuadores e os sensores podem ser interruptores.

Não se optou pela utilização de uma rede, como por exemplo o Modbus, para interligar a placa principal às placas secundárias, porque aumentaria significativamente o custo do projecto. Recorreu-se a uma extensão das portas de entrada/saída do microcontrolador para ligar as placas secundárias directamente ao microcontrolador. A alimentação das placas secundárias é feita pela placa principal, utilizando um dos pinos da ficha RJ11. Os restantes quatro pinos da ficha são reservados para duas entradas e duas saídas.

4.2 Arquitectura Híbrida

Esta arquitectura, permite comunicar com os dispositivos “Domoitech” sem efectuar qualquer alteração e permite comunicar com outros dispositivos KNXnet/IP existentes no mercado. A figura 42 ilustra um esquema desta arquitectura.

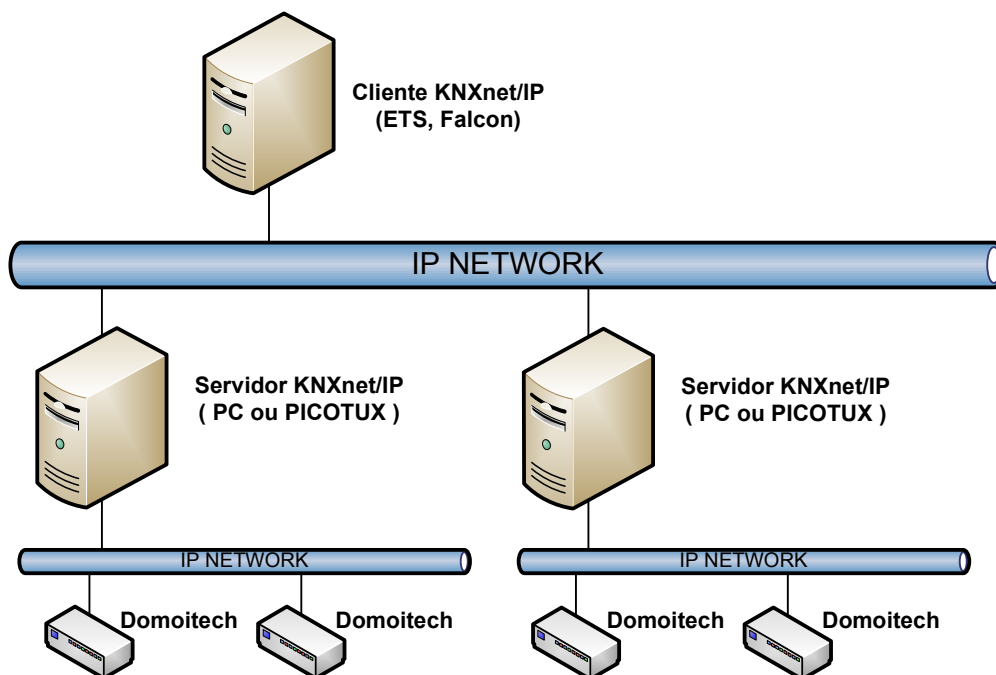


Figura 42 - Arquitectura híbrida

O servidor funciona como uma *bridge* entre as duas redes, “Domoitech” e KNXnet/IP.

Nesta arquitectura, o servidor KNXnet/IP é implementado sobre a tecnologia UNIX para computadores ou para sistemas UNIX embebido (PICOTUX). [14]

Na figura anterior, todos os barramentos da rede IP são um só, mas para uma melhor compreensão da estrutura, separou-se os barramentos para reforçar a ideia de que são redes diferentes, embora todas possam aceder a um meio físico partilhado. Para o servidor KNXnet/IP comunicar com uma rede que contém dispositivos “Domoitech” necessita de ter um mecanismo de software compatível com as tramas “Domoitech” porque estas são tramas KNX puras encapsuladas numa trama IP sem qualquer mecanismo referente na especificação do KNXnet/IP.

4.2.1 Projecto de Hardware

Com esta arquitectura, o servidor KNXnet/IP pode estar num computador ou num computador embebido como é o exemplo do PICOTUX. [14]

O PICOTUX é fisicamente semelhante a uma XPORT mas a correr o sistema operativo Linux para sistemas embebidos.

Utilizando um computador ou micro computador, o projecto de hardware é simples porque só é necessário o computador e a alimentação do mesmo. A figura 43 ilustra o diagrama de blocos da estrutura de hardware se for utilizado o PICOTUX para realizar o servidor KNXnet/IP.

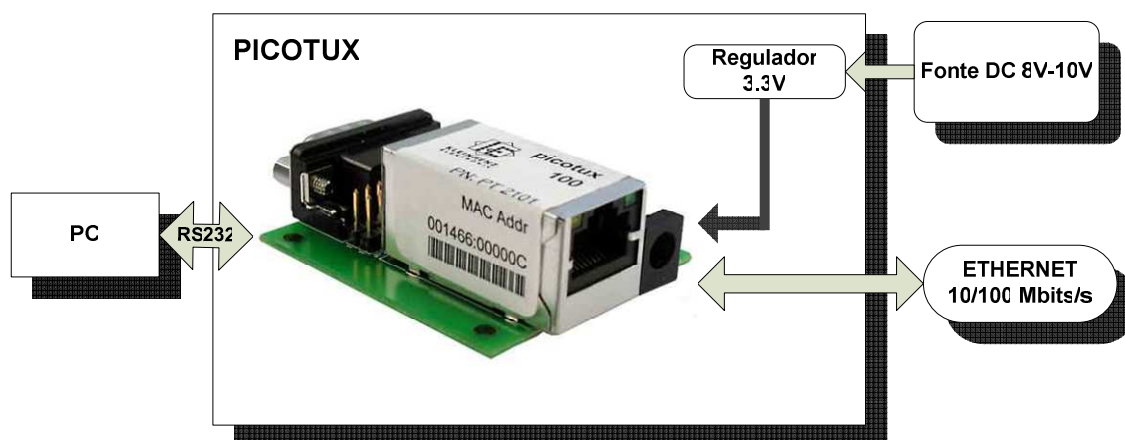


Figura 43 - Diagrama de blocos da estrutura de hardware

4.3 Comparação de arquitecturas

Para ambas as arquitecturas, o software a desenvolver é compatível e facilmente aplicado em sistemas embebidos sem sistema operativo (microcontrolador PIC) ou em sistemas operativos (Linux como tem o PICOTUX e Windows).

Cada arquitectura têm vantagens e desvantagens, mas em termos funcionais são idênticas.

Para conseguir identificar as vantagens e desvantagens de cada arquitectura para as poder comparar, é necessário saber o custo do hardware de cada arquitectura.

O preço final de um servidor KNXnet/IP da arquitectura Domoitech com KNXnet/IP, é semelhante ao projecto Domoitech, diferenciando apenas a diferença de preço dos componentes: XPORT AR e microcontrolador PIC18F6520. A tabela 3 resume o custo dos componentes de hardware do servidor KNXnet/IP.

Tabela 3 - Custo placa principal Domoitech com KNXnet/IP

Quantidade	Descrição	Preço unitário (€)	Valor (€)
1	Gc-Xport AR	44.93	44.93
1	Pic18f6520 TQFP	7.46	7.46
7	Socket vertical 6/6 pcb	0.70	4.90
7	Conector modular 6/6	0.49	3.43
1	Socket plcc 44 pinos	0.40	0.40
1	Cristal de 20Mhz	0.43	0.43
1	Pulsor	0.50	0.50
1	Lm7805 regulador de 5v	0.40	0.40
1	Ld1117av33 regulador 3.3v	0.92	0.92
-	Resistências	0.20	0.20
1	Placa PCB 7x7mm		
TOTAL			63.57 €

Ao valor final de 63,57 € é necessário acrescentar o custo de placas de interface com os sensores e actuadores. O custo médio dessas placas é de 8 €. A diferença de custo entre o sistema “Domoitech” com KNXnet/IP e o sistema Domoitech normal é de apenas 3,14€. [20]

É de notar que estes preços foram consultados na empresa Farnell [9], tendo em conta a compra de apenas uma unidade e por isso, torna o preço dos componentes, mais elevado. É possível portanto reduzir o custo do projecto se comprado em bastantes unidades. Existem outros valores que não foram tidos em conta, como os custos de transporte, o custo das placas de circuito impresso e o custo de engenharia.

O preço final de um servidor KNXnet/IP da arquitectura híbrida utilizando o PICOTUX é o custo do mesmo acrescido do valor do regulador de tensão e dos conectores (RS232 e alimentação). O preço do PICOTUX mais simples é de 100 € na versão base de 2MB de flash preparado para receber a imagem do LINUX. O custo final deverá ser de aproximadamente 122€ comprando a versão Picotux 112 com 2 MB Flash.

Vantagens da Arquitectura Domoitech com KNXnet/IP:

- Cada servidor KNXnet/IP tem uma rede KNX virtual e utiliza a arquitectura de hardware do projecto “Domoitech”;
- O custo em hardware do servidor KNXnet/IP desta arquitectura em relação ao da arquitectura híbrida, é mais baixo;
- Implementa uma rede KNX interna.

Desvantagem da Arquitectura Domoitech com KNXnet/IP:

- Necessitará de um *upgrade* na XPORT AR para implementar *multicast*.

Vantagens da Arquitectura Híbrida:

- Pode ser utilizado num PC ou num microPC como o PICOTUX, JACK PC [13], Calao's USB-9260 [15];
- Não necessita de efectuar alterações no projecto “Domoitech”;

- É compatível com dispositivos “Domoitech” e outros dispositivos KNXnet/IP.

Desvantagem da Arquitectura Híbrida:

- Solução mais cara que a arquitectura Domoitech com KNXnet/IP se utilizada com menos de 36 dispositivos “Domoitech”.

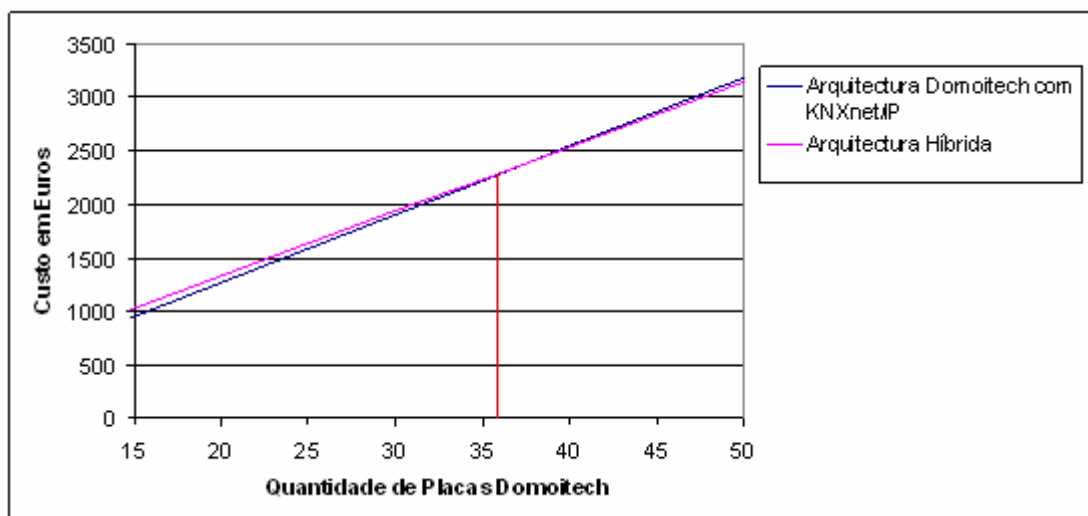


Figura 44 - Gráfico comparativo entre as arquitecturas



5 Implementação do Servidor KNXnet/IP

Depois de analisadas as arquitecturas físicas do sistema, foi necessário pensar qual delas iria implementar para posteriormente validar o que foi estudado e desenhado na teoria. Tendo em conta as semelhanças ao nível de *software* em ambas as arquitecturas, a opção foi de realizar um *software* multi-plataforma, para ser facilmente adaptado a qualquer uma das arquitecturas.

O *software* a implementar, seria um servidor KNXnet/IP e por isso, era necessário um estudo do protocolo para ajudar no desenho da estrutura do mesmo.

5.1 Arquitectura do Software

De forma a realizar o *software* multi-plataforma, foram estudadas arquitecturas de *software* diferentes e linguagens de programação indicadas para o propósito.

A linguagem de programação escolhida foi o ANSI C, e o compilador o GCC. O ANSI C é uma norma publicada pelo instituto de normas nacionais Americanas (*American National Standards Institute*). Esta linguagem de programação fornece aos programadores, a vantagem de criarem código que é facilmente transportado para outras plataformas. A opção de utilizar esta linguagem de programação é pela vantagem mencionada e também porque o compilador GCC, é uma ferramenta livre e existe para diversas plataformas; Windows, Linux, Windows CE, PALM, Mac OS, etc.

Depois de estudar o protocolo KNXnet/IP, foram identificados 5 módulos de *software* essenciais: núcleo, gestão de dispositivos, *tunneling*, *routing* e cEMI. Foi necessário a criação de um módulo denominado por knxnet/ip cuja função é de gerir os restantes cinco módulos e contém todas as definições de objectos KNXnet/IP.

Para além do *software* ser multi-plataforma, terá de ser funcionar em hardware diferente. Por exemplo, um microcontrolador PIC, contém diferentes interfaces de entradas/saídas em relação a um computador. De forma a contornar este obstáculo, foram criados três módulos que serão diferentes para um microcontrolador PIC e um computador. Os três módulos são: configuração de hardware, HAL(IP) e HAL(TP1).

O módulo de configuração de hardware, contém as rotinas de inicialização e configuração das interfaces de entradas/saídas (portas série, *ethernet*, ...).

Os módulos de HAL(IP) e HAL(TP1), têm funções semelhantes, servem para realizar a interface aos meios físicos; TCP/IP sob *Ethernet* e *Twisted-Pair 1* do KNX/EIB.

A figura 45 ilustra a arquitectura de software com todos os módulos e os sentidos de comunicação entre eles.

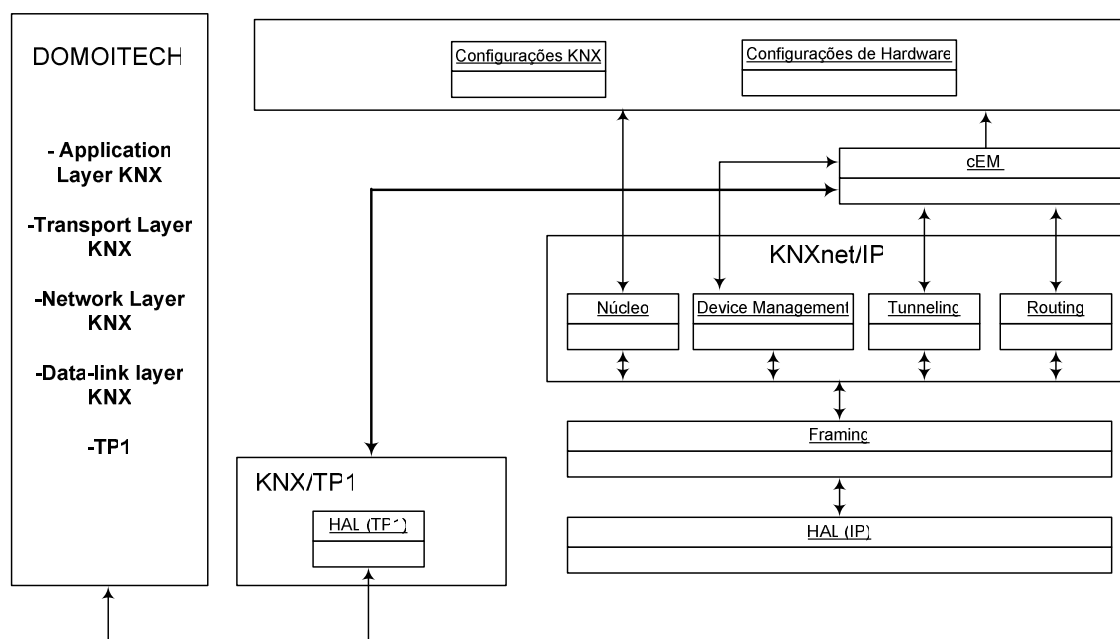


Figura 45 - Arquitectura de software

Na figura anterior, foi incluído um bloco que contém o software Domoitech e onde é visível a sua interligação ao servidor KNXnet/IP através do módulo HAL(TP1). Inclui também o módulo *framing* cuja funcionalidade é de tratar as tramas KNXnet/IP recebidas pela interface de *ethernet*. Para um melhor entendimento da função de cada módulo, será feito o resumo das funcionalidades de cada um.

O HAL(IP) efectua a interface à rede TCP/IP *ethernet* por onde envia e recebe tramas KNXnet/IP encapsuladas em tramas do protocolo TCP/IP.

O *framing* recebe as tramas KNXnet/IP e efectua uma análise primária da trama através da verificação do cabeçalho. Se for uma trama válida, passa-a para o módulo knxnet/ip. Quando recebe uma trama vinda do módulo knxnet/ip, acrescenta

o cabeçalho completando assim a trama KNXnet/IP e envia-a ao módulo HAL(IP), indicando também qual o HPAI de destino.

O módulo *knxnet/ip* contém as definições de objectos KNXnet/IP como por exemplo; a definição da estrutura de um HPAI ou da estrutura de uma trama KNXnet/IP. A sua principal função é distribuir as tramas KNXnet/IP para os módulos correctos.

O módulo do núcleo contém todos os serviços que estão definidos na norma KNXnet/IP que o núcleo implementa e que foram referidos anteriormente. Esses serviços são: o *search request*, *search response*, *description request*, *description response*, *connect request*, *connect response*, *connection-state request*, *connections state response*, *disconnect request* e *disconnect response*.

O módulo *devicemanagement* implementa todos os serviços da Gestão de Dispositivos que se encontra definida na norma KNXnet/IP. Os serviços são: *device configuration request* e *device configuration acknowledge*.

O módulo *tunneling* contém os serviços *tunnelling request* e *tunnelling acknowledge* definidos na norma.

O módulo de *routing* foi criado e preparado para a implementação dos dois serviços *routing indication* e *routing lost message*. A criação destes serviços foi deixada para o final pela inexistência de um equipamento que permitisse testar estas funções. Este módulo acabou por não ser preenchido devido ao curto prazo disponível para a implementação do *software*.

O módulo cEMI, contém todas as funções e definições para tratamento das tramas cEMI definidas na norma e que estão descritas no sub-sub-capítulo 2.5.2.

Os módulos de configurações KNX e de hardware, contém o que o próprio nome indica. O módulo configurações KNX contém a definição e funções para tratamento de tramas KNX, e as propriedades dos *device objects* que o servidor cEMI gere. Essas propriedades estão definidas na tabela 5 do apêndice 1.

O módulo HAL(TP1) tem a mesma função que o HAL(IP) só que a interface é feita com o barramento *Twisted Pair 1* do KNX. Este módulo recebe as tramas KNX do módulo cEMI e envia-as para o barramento TP1 KNX e se receber alguma trama KNX do barramento TP1, verifica se a trama é destinada a ele e envia para o módulo cEMI.

A figura 46 ilustra um diagrama UML em que os componentes são os ficheiros que contêm código.

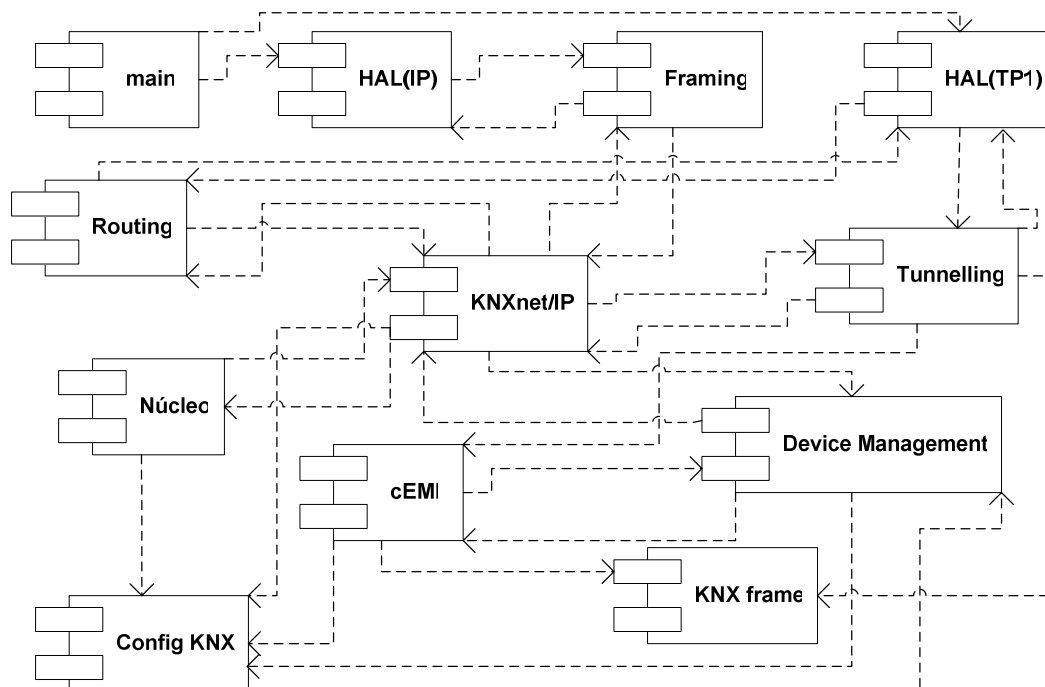


Figura 46 - Diagrama UML da estrutura de ficheiros do software

Em vez da criação de um ficheiro só albergando todo o código, dividiu-se o código em vários ficheiros de forma a facilitar a compreensão da estrutura do software apresentada anteriormente. Se observarmos o diagrama UML, reparamos que os ficheiros têm praticamente a mesma estrutura que a arquitectura do software. Desta forma é possível identificar facilmente os módulos entre o código o que facilita a compreensão das funções existentes em cada ficheiro. Os ficheiros Núcleo, Routing, Tunneling e *Device Management* contêm os respectivos módulos.

O diagrama UML ilustra também a dependência entre as funções existentes nos ficheiros. Aqui é notória a dependência bidireccional que quase todas as funções dos vários ficheiros têm com funções e variáveis do ficheiro KNXnet/IP. Além da semelhança entre a estrutura de ficheiros e a arquitectura de software representada na figura 45, as dependências também se assemelham aos sentidos de comunicação entre módulos no diagrama da arquitectura. A única diferença principal é que as funções do ficheiro cEMI, não comunicam directamente com o HAL(TP1) como acontece no diagrama da arquitectura, mas comunicam com as funções existentes no ficheiro *tunnelling* e estas por sua vez é que comunicam com

o HAL(TP1). O ficheiro *main* contém a função principal que serve para inicializar o servidor KNXnet/IP e coloca-lo no modo de “escuta” dos meios físicos.

O *software* é praticamente a transcrição da norma KNXnet/IP para algoritmia acrescentando uns módulos de manipulação do *hardware*. Para validar o *software*, é necessário efectuar testes a todos os serviços do KNXnet/IP. Só assim é possível verificar a solidez do mesmo e se todos os módulos estão a funcionar correctamente.

5.2 Validação e Resultados

Depois da conclusão do servidor KNXnet/IP, era necessário efectuar testes com equipamentos KNX/EIB. Como o acesso a este tipo de equipamentos era muito demorado e não havia muito tempo para efectuar os testes, recorreu-se a vários *softwares* aprovados, que pudessem comunicar com equipamentos KNX/EIB sob a rede TCP/IP *Ethernet*. Um desses *softwares* é o ETS que permite configurar redes KNX/EIB que comuniquem sob TCP/IP *Ethernet* recorrendo a um servidor KNXnet/IP que funciona como uma *gateway* entre o barramento TCP/IP *Ethernet* e o barramento TP1 KNX/EIB. Os outros *softwares* testados têm a mesma funcionalidade que o ETS, mas foram criados em projectos de investigação. Para além do ETS, foram utilizados os *softwares*; *Net'n Node* [6] que efectua testes e análises de redes KNX/EIB e uma aplicação de demonstração que o projecto *Calimero* [17] fornece. O *Calimero* é um cliente KNXnet/IP realizado em Java formado por um conjunto de API's que facilitam a criação de aplicações para controlo e acesso remoto a sistemas KNX/EIB.

Para validar todas as funcionalidades do servidor KNXnet/IP criado, era necessário utilizar adicionalmente uma rede KNX/EIB TP1 para verificar se o servidor conseguia cumprir o seu papel. Ainda não foi criado um *software* que pudesse simular uma rede KNX/EIB TP1 por isso era necessário obrigatoriamente um equipamento físico KNX/EIB. Surgiu a ideia de utilizar o projecto Domoitech que nunca tinha sido testado com outros equipamentos KNX/EIB aprovados e assim, além de validar o servidor KNXnet/IP, poderia ser possível comprovar se o Domoitech implementava correctamente o protocolo KNX/EIB sob meio físico TP1. Como foi explicado no sub capítulo do Domoitech, este utiliza o protocolo de rede

TCP/IP sob *Ethernet* para simular um barramento TP1. O meio físico que ele utiliza é uma ilusão visto que as tramas que circulam no barramento são tramas KNX/EIB TP1 encapsuladas dentro de uma trama IP e como foi explicado anteriormente, este servidor KNXnet/IP multiplataforma criado, tem a função de colmatar esta não conformidade com o protocolo KNX/EIB.

Para poder ligar o servidor KNXnet/IP à rede Domoitech, foi necessário alterar o módulo HAL(TP1) de forma a que em vez de enviar as tramas KNX/EIB para o barramento TP1, estas teriam de ser encapsuladas numa trama IP e enviadas pelo interface *Ethernet* que se liga ao barramento TCP/IP *Ethernet* partilhado com o Domoitech.

Para uma melhor compreensão da instalação dos equipamentos na rede de testes, a figura 47 ilustra o esquema dessa instalação.

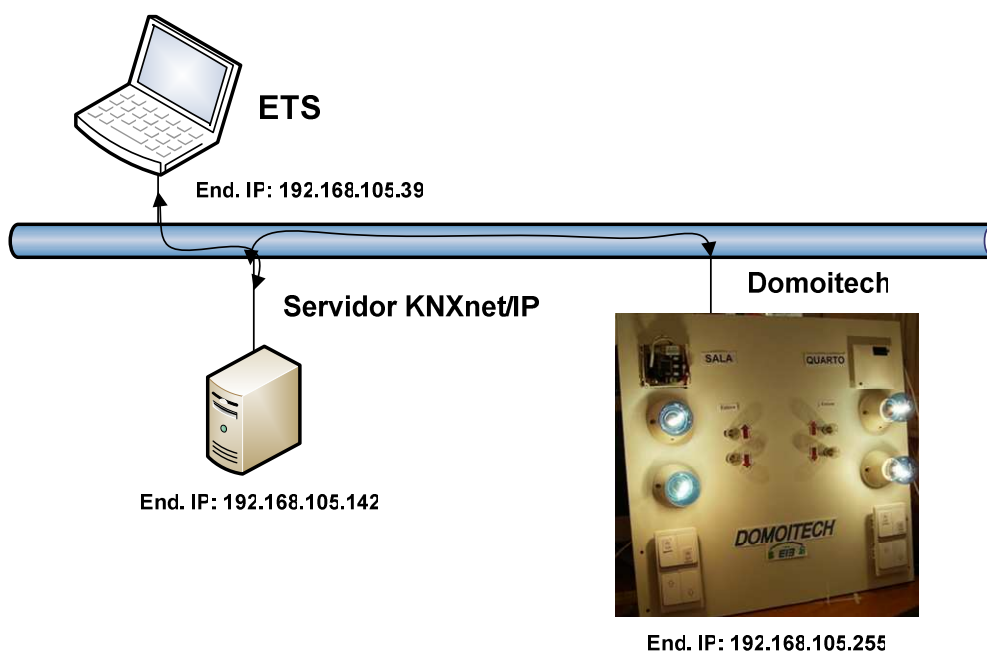


Figura 47 - Esquema da instalação da plataforma de testes

Apesar de o meio físico ser partilhado, nesta instalação temos dois tipos de redes lógicas diferentes. Numa circulam tramas do tipo KNXnet/IP e na outra rede circulam tramas KNX/EIB TP1 encapsuladas numa trama IP normal.

Depois da instalação e configuração dos endereços IP dos dispositivos, foi necessário instalar no computador que tinha os *software's*; ETS, *Net'n Node* e a aplicação *demo* do *Calimero*, um outro *software* que permitiria verificar as tramas que passavam no barramento TCP/IP *Ethernet*. Esse tipo de *software* é denominado

por *Sniffer* de rede, cuja funcionalidade é “escutar” continuamente o meio físico e captar todas as tramas que por lá passam. Existe uma variedade grande desse tipo de *software*, mas optou-se a utilização do *Analyzer* [4] que foi desenvolvido no Politécnico de Torino em Itália.

Como foi descrito anteriormente, a validação do servidor KNXnet/IP, terá de ser feita testando o funcionamento de todos os serviços do KNXnet/IP. Assim é garantido um teste a todos os módulos do *software* porque basta um falhar, que os serviços não funcionam.

Para testar o funcionamento dos serviços, recorre-se ao *software* ETS ou também a qualquer um dos outros que já foram mencionados. Para testar alguns dos serviços, como os do módulo de *Tunnelling* e o módulo cEMI, é também necessário utilizar o Domoitech para comprovar se as tramas KNX são bem construídas pelo servidor KNXnet/IP.

Os testes que serão apresentados foram feitos todos com o ETS, mas estes mesmos testes foram realizados utilizando os outros *software*'s e os resultados foram os mesmos.

5.2.1 Testes aos serviços do Núcleo

O primeiro serviço que o ETS utiliza é o *Search Request* para procurar servidores KNXnet/IP no barramento. A figura 48 ilustra a interface da aplicação *Analyzer*, onde é visível a troca de pacotes TCP/IP sob o protocolo UDP entre o servidor KNXnet/IP e o ETS. O servidor KNXnet/IP tem o endereço IP 192.168.105.142 e o computador que corre o ETS tem o IP 192.168.105.39.

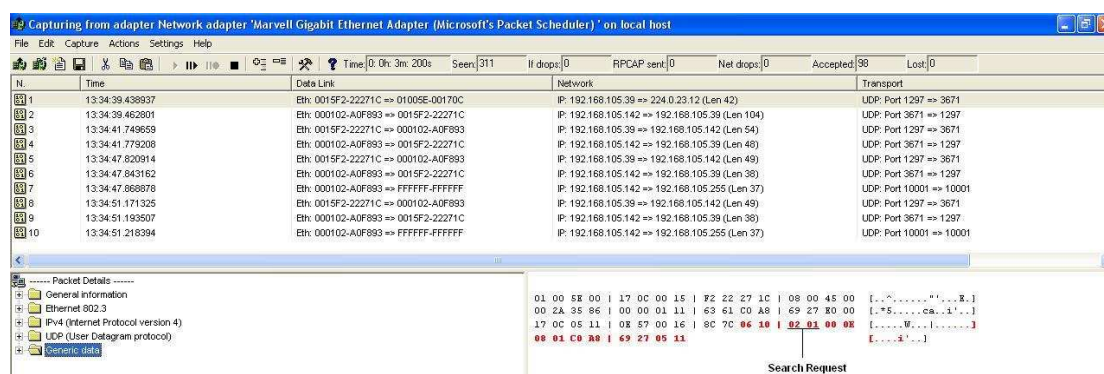


Figura 48 - Teste Search Request

É visível na figura 48 que o ETS enviou uma trama KNXnet/IP com o serviço *Search Request* para o endereço IP de multicast 224.0.23.12 e porta UDP número 3671. Como foi explicado no capítulo da descrição da norma KNXnet/IP, este serviço envia as tramas sempre para este endereço de multicast e porta UDP.

O servidor terá de responder a este serviço com um *Search Response* e como podemos ver na figura 49, o servidor KNXnet/IP, identificou o pedido do ETS e respondeu correctamente.

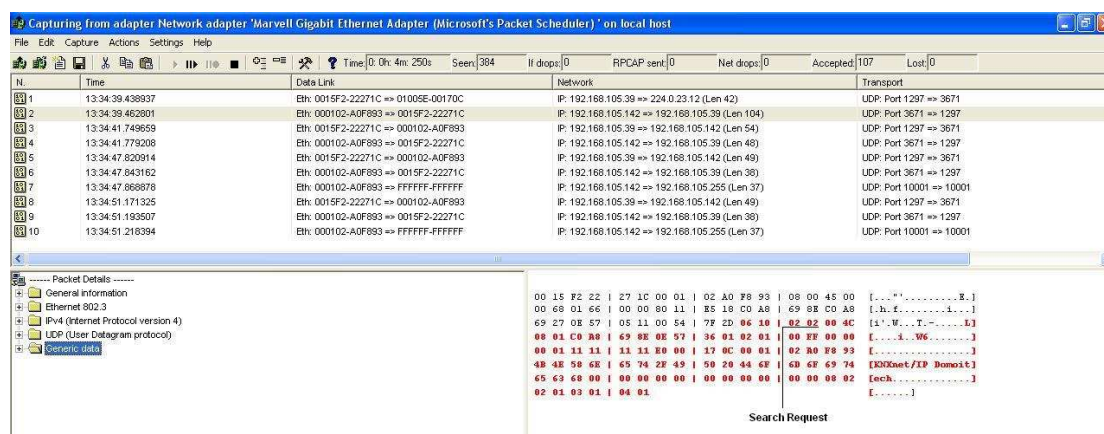


Figura 49 - Teste Search Response

Os serviços de procura e resposta de um servidor KNXnet/IP estão a funcionar correctamente como foi possível verificar na figura 48 em que recebeu o *Search Request* e na figura 49 envia a resposta para o ETS como o serviço *Search Response*.

Depois do servidor KNXnet/IP conectado ao ETS é possível utilizar os restantes serviços. Um dos serviços que permitem saber mais informações sobre o servidor KNXnet/IP é o *Description*. Na figura 50 é possível identificar a reposta ao pedido de *Description Request* enviado pelo ETS para o servidor KNXnet/IP.

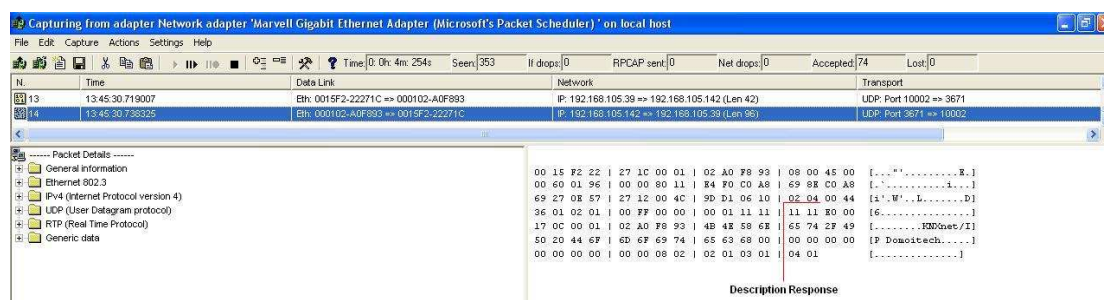


Figura 50 - Teste Description Response

Se observarmos com atenção a figura anterior, o pacote anterior à resposta enviada pelo servidor KNXnet/IP, é o pedido de *Description Request* ao qual o servidor responde correctamente.

Outro serviço importante do Núcleo, é o *Connect*. Existem dois tipos de conexões a criar, uma para o *Tunnelling* e outra para a Gestão de Dispositivos. Para ambos os pedidos de *Connect Request* feitos pelo ETS, o servidor KNXnet/IP, respondeu correctamente com o *Connect Response* indicando o número do canal de comunicação criado. Este servidor permite ter até 3 canais de comunicação activos. Esse valor é facilmente alterado acedendo ao ficheiro KNXnet/IP que contém a variável que define esse valor. A figura 51 ilustra o exemplo de uma resposta a um *Connect Request* para uma ligação *Tunnelling*:

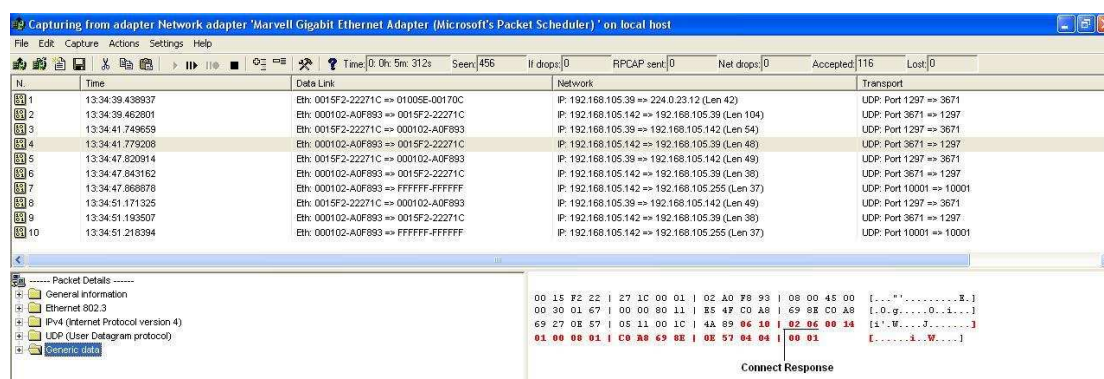


Figura 51- Teste Connect Response

Existe um serviço de teste da ligação que é utilizado pelo ETS para verificar se o canal de comunicações ainda se mantém activo ou se por algum problema existente (p.ex: quebra da ligação), foi desactivo. Esse serviço é o *Connection State Request* ao qual o servidor KNXnet/IP terá de responder com um *Connection State Response* caso o canal de comunicação esteja activo. A figura 52 ilustra um exemplo de envio da trama *Connection State Response* pela parte do servidor KNXnet/IP como resposta ao *Connection State Request* que recebeu anteriormente.

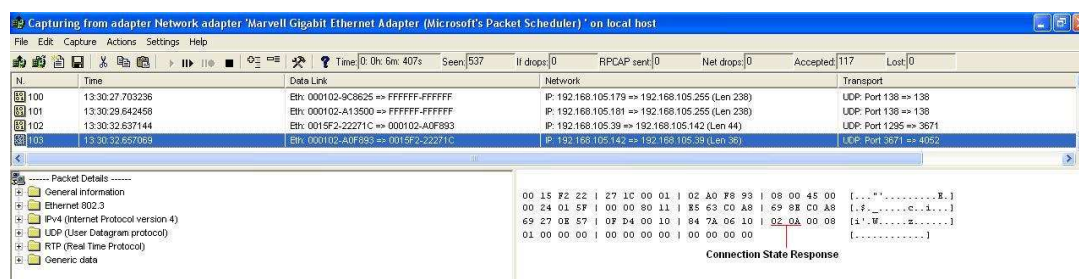


Figura 52 - Teste Connection State Response

Para terminar uma ligação criada através do serviço de *Connect*, utiliza-se o *Disconnect*. A figura 53 ilustra a trama de *Disconnect Response* enviada pelo servidor KNXnet/IP como resposta à trama de *Disconnect Request* que recebeu anteriormente.

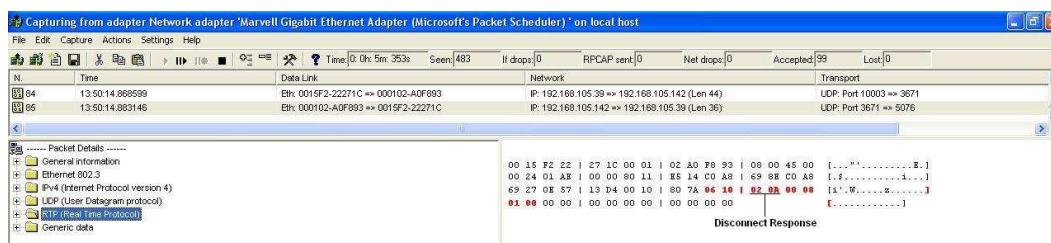


Figura 53 - Teste *Disconnect Response*

5.2.2 Testes aos serviços do Tunneling

Depois de iniciado o canal de comunicação de *tunneling*, dentro desse canal serão iniciadas trocas de tramas do tipo *Tunneling Request* e como resposta a esta trama, o destinatário terá de enviar tramas do tipo *Tunneling Acknowledge*.

Estes serviços servem para enviar tramas KNX/EIB pelo barramento TCP/IP *Ethernet*. O servidor KNXnet/IP quando recebe uma *Tunneling Request*, terá de retirar a trama KNX/EIB que veio encapsulada numa trama cEMI que por sua vez veio encapsulada numa trama KNXnet/IP. Depois envia a trama KNX/EIB directamente para o barramento KNX/EIB TP1 que neste caso é o barramento TCP/IP *Ethernet* que o Domoitech utiliza.

A figura 54 ilustra uma trama do tipo *Tunneling Request* que é enviada pelo ETS cuja trama KNX/EIB tem a função de mandar ligar a lâmpada com o endereço 1 da rede KNX/EIB Domoitech.

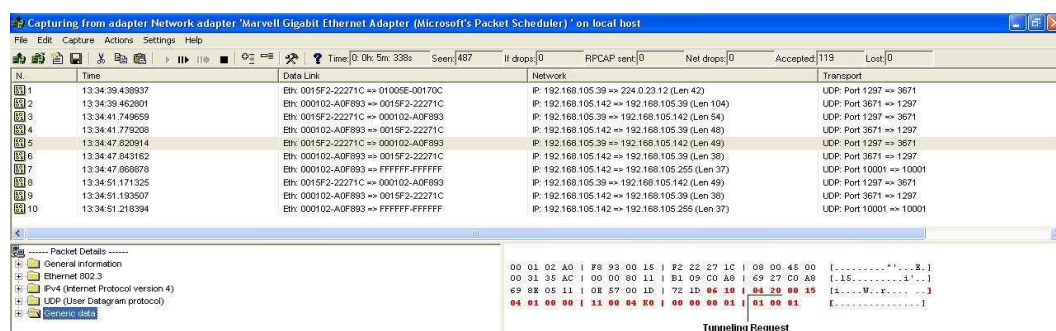


Figura 54 - Teste *Tunneling Request*

Como resposta à recepção da trama anterior, o servidor KNXnet/IP envia uma trama de *Tunneling Acknowledge* ao ETS, informando de que recebeu correctamente a trama anterior e que a irá enviar para a rede KNX/EIB. A figura 55 ilustra o envio da trama de *Tunneling Acknowledge*.

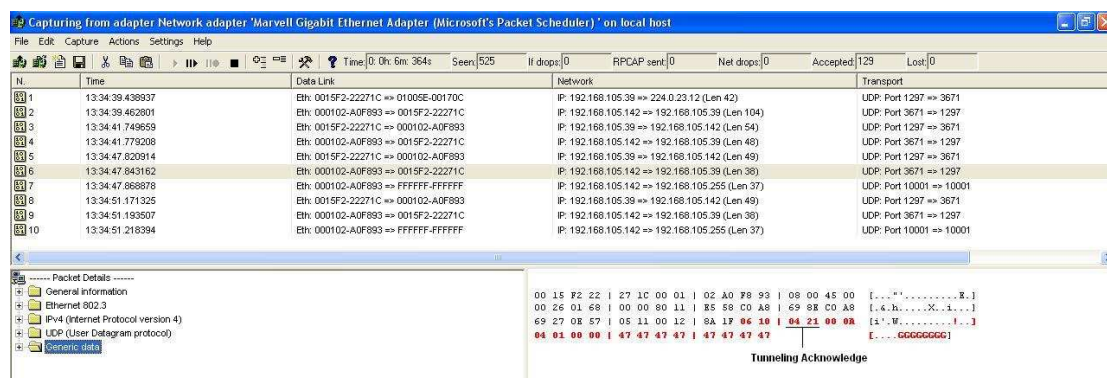


Figura 55 - Teste Tunneling Acknowledge

Depois de retirada a trama KNX/EIB, o servidor KNXnet/IP envia-a para a rede Domoitech. A figura 56 ilustra o envio dessa trama KNX/EIB para a rede Domoitech onde o endereço IP é o endereço de *broadcast* 192.168.105.255.

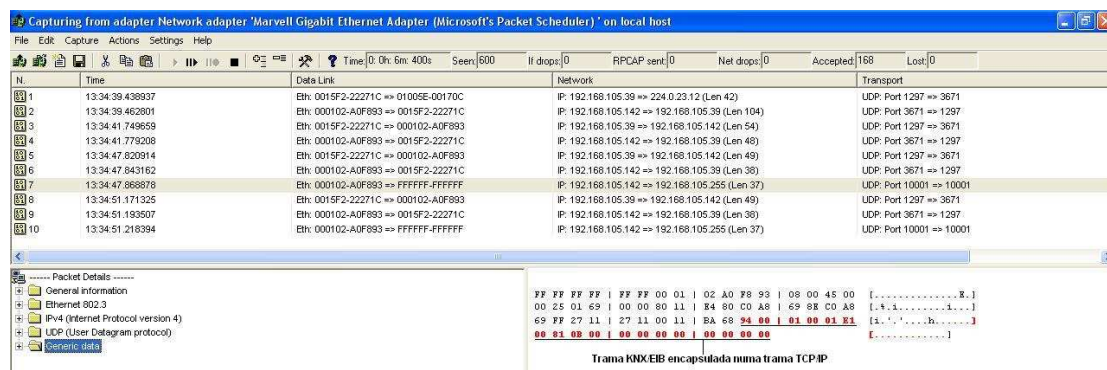


Figura 56 - Envio de trama KNX/EIB para o Domoitech

Após o envio da trama KNX/EIB para a rede Domoitech, verificou-se que a lâmpada com o endereço 1 acendeu como era previsto, caso ambos os *softwares* fossem verosímeis. Com este teste, comprova-se que os serviços do módulo *Tunneling* e cEMI do servidor encontram-se bem implementados e o que Domoitech implementa correctamente o protocolo KNX/EIB sob TP1.

5.2.3 Testes aos serviços da Gestão de Dispositivos

Tal como os anteriores serviços do *Tunneling*, depois de criado o canal de comunicações para a gestão de dispositivos, as tramas a circularem dentro desse canal são do tipo *Device Configuration Request* e *Device Configuration Acknowledge*.

No exemplo da figura 57, é enviada uma trama *Device Configuration Request* pelo ETS com destino o servidor KNXnet/IP que transporta uma mensagem cEMI do tipo *M_PropWrite* cuja função é alterar o valor de uma propriedade.

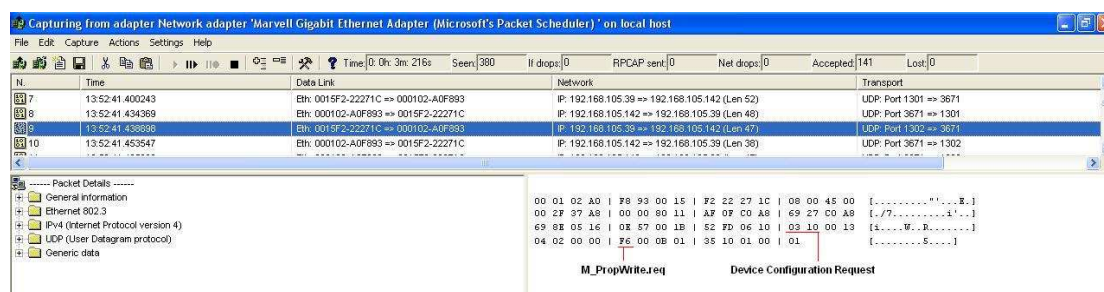


Figura 57 - Teste *Device Configuration Request*

Como resposta à trama anterior, o servidor KNXnet/IP enviou uma trama com o serviço *Device Configuration Acknowledge* para informar o ETS que recebeu correctamente o pedido. A trama de reposta encontra-se ilustrada na figura 58.

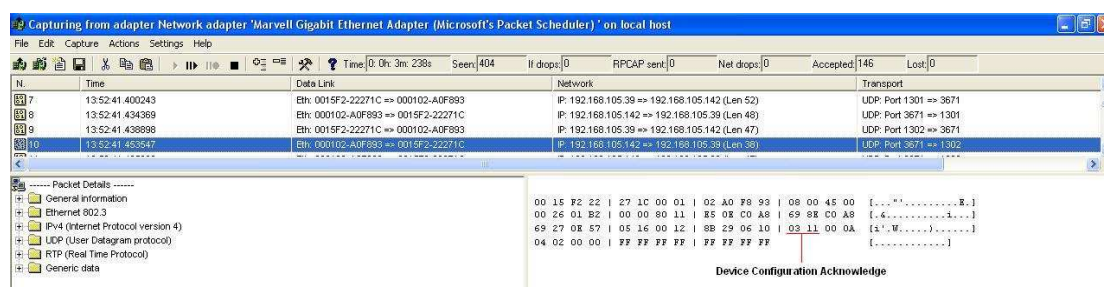


Figura 58 - Teste *Device Configuration Acknowledge*

Após a recepção da mensagem cEMI com pedido de alteração do valor de uma propriedade, o servidor KNXnet/IP actualizou de imediato a respectiva variável com o novo valor. Isto comprova que o módulo de gestão de dispositivos e o módulo cEMI de gestão, estão a funcionar correctamente.

6 Conclusão

Esta dissertação teve como objectivo aproveitar o projecto Domoitech para criar uma arquitectura KNX/EIB de baixo custo que pudesse utilizar uma rede IP sob *Ethernet*.

Devido à recente aceitação da Domótica no mercado Português, os preços das soluções existentes ainda são muito elevados. Têm surgido cada vez mais soluções proprietárias de Domótica de forma a apresentar produtos a preços competitivos. O problema dessas soluções é que o cliente ficará de certa forma “preso” ao fabricante por não existirem no mercado outros equipamentos de concorrência compatíveis com a solução que adquiriu. A opção de escolher um protocolo normalizado de Domótica serve para colmatar a incompatibilidade com outros equipamentos e permitir ao cliente a opção de escolha entre os equipamentos da concorrência.

O facto de utilizar um protocolo normalizado como o robusto KNX/EIB, aumenta significativamente o custo da solução. Contudo, é possível rivalizar com a concorrência no mercado, se a arquitectura for bem estudada ou apresentar soluções inovadoras e adequadas às necessidades das pessoas.

Depois de um estudo aprofundado do protocolo KNX/EIB e do KNXnet/IP, foram analisadas duas arquitecturas possíveis. A arquitectura Domoitech com KNXnet/IP, consiste em efectuar alterações de *hardware* e *software* ao Domoitech para acrescentar a norma KNXnet/IP. A arquitectura híbrida, consiste em manter intacto o Domoitech e criar uma *gateway* entre o Domoitech e redes KNXnet/IP. Nesta solução, em vez de colocar o servidor KNXnet/IP dentro do Domoitech, este é colocado num dispositivo exterior que pode ser um computador, PDA, etc. No confronto entre as arquitecturas, a mais económica é a arquitectura Domoitech com KNXnet/IP e só compensará a arquitectura híbrida para uma instalação que necessite de mais de 36 placas Domoitech.

Apresentadas estas duas arquitecturas, concluiu-se que ao nível do *software* foi necessário criar um servidor KNXnet/IP que pudesse ser multi-plataforma para ser facilmente adaptável a qualquer uma das arquitecturas. O servidor KNXnet/IP foi desenvolvido na linguagem de programação C não só pela razão de ser uma linguagem que permite multi-plataforma como pela razão de que os compiladores do microcontrolador PIC não fornecem outra linguagem para além do C e *Assembly*.

Após a implementação do servidor KNXnet/IP, foram realizados testes utilizando vários *softwares* como o ETS, Net'n Node e uma aplicação feita com o Calimero. Ao longo da implementação do servidor, foram efectuados testes a cada serviço e módulo implementado, e no final efectuaram-se novos testes a cada um dos serviços. Utilizando o *Analyser* pela agradável interface que fornece, foi possível verificar todo o tráfego no barramento de *Ethernet* onde estavam ligados o servidor KNXnet/IP, o Domoitech e um computador a correr o ETS. Foi possível observar o fluxo de tramas KNXnet/IP entre os dispositivos e verificar que os serviços do servidor estão a funcionar correctamente. Com este servidor, foi também possível comprovar que o Domoitech tem o protocolo EIB a funcionar correctamente. A configuração testada foi para o caso da arquitectura híbrida, contudo não foram efectuados testes para a arquitectura Domoitech com KNXnet/IP por falta de tempo.

Ao longo da realização deste trabalho o contacto com o mercado esteve sempre presente para uma constante actualização e atenção às novidades. Visitas a exposições como a Concreta em Outubro de 2007 e pequenas exposições de produtos de alguns fabricantes, permitiram uma constante actualização sobre o que existe no mercado. Foi notório que o futuro da Domótica passará pela utilização de redes TCP/IP sem fios. O KNX/EIB é um dos protocolos que suporta este tipo de rede e a solução apresentada está preparada para a utilização deste tipo de redes sem fios porque funciona sob o protocolo TCP/IP.

6.1 Desenvolvimentos futuros

Para ser possível colocar no mercado um produto, são necessários passos importantes. Não basta desenhar o produto, efectuar um estudo de mercado e calcular custos, serão necessárias mais etapas, tais como: a certificação do produto, efectuar testes de conformidade com normas europeias de segurança, etc. Quando se consegue lançar o produto no mercado, já este está ultrapassado, por isso a análise do mercado terá de ser feita com uma previsão futura para altura em que será lançado o produto.

Este trabalho que intitulei como FEUP KNX é o início da criação de uma solução de domótica de baixo custo que utiliza um protocolo caro como o KNX/EIB e que utilizará a rede TCP/IP sob *Ethernet* ou as redes Wi-Fi. O principal objectivo do



FEUP KNX é fornecer soluções de domótica KNX/EIB de fácil instalação, que não necessite de utilizar outra cablagem e bastante simples. Não serão exploradas todas as potencialidades do KNX/EIB, mas não é possível equilibrar o custo com a complexidade.

Pontos que não foram completados neste trabalho e que terão de ser implementados futuramente são:

- Implementar o código do módulo de *Routing*;
- Acrescentar o servidor KNXnet/IP ao *software* do Domoitech para ser possível testar a arquitectura Domoitech com KNXnet/IP;
- Implementar os restantes serviços do KNX/EIB no Domoitech;
- Melhorar o *hardware* do Domoitech;
- Realizar testes do sistema KNX/EIB do Domoitech a funcionar com dispositivos KNX/EIB de outros fabricantes e
- Certificar o produto junto da Konnex.

Referências

- [1] DOMOTICAPT, www.domotcapt.com/domotica.php, Fevereiro 2008
- [2] MONTEIRO, José, Montagem de um sistema didático utilizando a tecnologia Instabus/EIB da Siemens, FEUP, Dezembro 2004 .
- [3] DOMODESK, www.domodesk.com, Fevereiro 2008
- [4] ANALYZER, <http://analyzer.polito.it>, Fevereiro 2008
- [5] CASAINTELI, <http://acasainteligente.blog.com>, Fevereiro 2008
- [6] NETNNODE, www.netnnode.com, Fevereiro 2008
- [7] KNX, www.knx.org, Fevereiro 2008
- [8] KNX HANDBOOK, KNX Specification, Konnex Association, Bruxelas, Fevereiro de 2004
- [9] FARNELL, ww.farnell.com, Fevereiro 2008
- [10] LANTRONIX, www.Xport-RJ45.com, Fevereiro 2008
- [11] MICROCHIP, www.microchip.com, Fevereiro 2008
- [12] ABB, www.abb.pt, Fevereiro 2008
- [13] JADE, www.jadeintegration.com/jackpc.php, Fevereiro 2008
- [14] PICOTUX, www.picotux.com, Fevereiro 2008
- [15] CALAO'S, www.linuxdevices.com/news/NS6730529835.html, Fevereiro 2008
- [16] IANA, www.iana.org, Fevereiro 2008
- [17] CALIMERO, <http://calimero.sourceforge.net>, Fevereiro 2008
- [18] KNX LIVE CD, <https://www.auto.tuwien.ac.at/a-lab/knx-eib.html>, Fevereiro 2008
- [19] KNXCALIBUR, <https://www.auto.tuwien.ac.at/a-lab/knxcalibur.html>, Fevereiro 2008
- [20] DOMOITECH, Domoitech – Domótica com protocolo EIB, Duarte Pinto e Diana Palma, FEUP - DEEC, Julho 2007.

Apêndice 1

A tabela seguinte contém todos os códigos de mensagens cEMI. Quando um servidor cEMI recebe uma mensagem com um código que não existe nesta tabela, a mensagem é ignorada e portanto não é enviada uma mensagem de confirmação.

Tabela 4 - Códigos das mensagens cEMI

Serviço	Código da mensagem cEMI	Destino
L_Busmon.ind	2Bh	Camada de rede
L_Raw.ind	2Dh	Camada de rede
L_Raw.req	10h	Camada ligação lógica
L_Raw.con	2Fh	Camada de rede
L_Data.req	11h	Camada ligação lógica
L_Data.con	2Eh	Camada de rede
L_Data.ind	29h	Camada de rede
L_Poll_Data.req	13h	Camada ligação lógica
L_Poll_Data.con	25h	Camada de rede
M_PropRead.req	FC h	Servidor cEMI de gestão
M_PropRead.con	FB h	Cliente cEMI de gestão
M_PropWrite.req	F6 h	Servidor cEMI de gestão
M_PropWrite.con	F5 h	Cliente cEMI de gestão
M_PropInfo.ind	F7 h	Cliente cEMI de gestão
M_Reset.req	F1 h	Servidor cEMI de gestão
M_Reset.ind	F0 h	Cliente cEMI de gestão

Os serviços cEMI de configuração e gestão, têm a funcionalidade de alterar ou ler valores de propriedades de objectos. Existem 14 tipos de objectos no KNX/EIB, mas com estes serviços apenas é possível alterar propriedades do *Device Object*. A tabela seguinte contém as várias propriedades deste tipo de objecto.

Tabela 5 - Propriedades do device object

PID	Nome da Propriedade	Tipo	Leitura/ Escrita
51	PID_PROJECT_INSTALLATION_ID	Unsigned Int	L/E
52	PID_KNX_INDIVIDUAL_ADDRESS	Unsigned Int	L/E
53	PID_ADDITIONAL_INDIVIDUAL_ADDRESSES	Unsigned Int[]	L/E
54	PID_CURRENT_IP_ASSIGNMENT_METHOD	Unsigned Char	L/-
55	PID_IP_ASSIGNMENT_METHOD	Unsigned Char	L/E
56	PID_IP_CAPABILITIES	Unsigned Char	L/-
57	PID_CURRENT_IP_ADDRESS	Unsigned Long	L/-
58	PID_CURRENT_SUBNET_MASK	Unsigned Long	L/-
59	PID_CURRENT_DEFAULT_GATEWAY	Unsigned Long	L/-
60	PID_IP_ADDRESS	Unsigned Long	L/E
61	PID_SUBNET_MASK	Unsigned Long	L/E
62	PID_DEFAULT_GATEWAY	Unsigned Long	L/E
63	PID_DHCP_BOOTP_SERVER	Unsigned Long	L/-
64	PID_MAC_ADDRESS	Generic 6 hex	L/-
65	PID_SYSTEM_SETUP_MULTICAST_ADDRESSES	Unsigned Long	L/-
66	PID_ROUTING_MULTICAST_ADDRESS	Unsigned Long	L/E
67	PID_TTL	Unsigned Char	L/E
68	PID_EIBNETIP_DEVICE_CAPABILITIES	Unsigned Int	L/-
69	PID_EIBNETIP_DEVICE_STATE	Unsigned Char	L/-



70	PID_EIBNETIP_ROUTING_CAPABILITIES	Unsigned Char	L/-
71	PID_PRIORITY_FIFO_ENABLED	Unsigned Char	L/-
72	PID_QUEUE_OVERFLOW_TO_IP	Unsigned Int	L/-
73	PID_QUEUE_OVERFLOW_TO_KNX	Unsigned Int	L/-
74	PID_MSG_TRANSMIT_TO_IP	Unsigned Long	L/-
75	PID_MSG_TRANSMIT_TO_KNX	Unsigned Long	L/-
76	PID_FRIENDLY_NAME	Unsigned Char [30]	L/E