

**Edifícios Inteligentes – Soluções para gestão de
climatização em instalação de Domótica KNX
Estudo de um Caso**

Tiago Manuel Brás de Abreu

Relatório Final do Trabalho de Projeto apresentado à
Escola Superior de Tecnologia e Gestão
Instituto Politécnico de Bragança
Para obtenção do grau de Mestre em
Engenharia Industrial

Trabalho realizado sob a orientação de
Professor Doutor José Augusto Carvalho

Outubro de 2013

Edifícios Inteligentes – Soluções para gestão de climatização em instalação de Domótica KNX

Estudo de um Caso

Tiago Manuel Brás de Abreu

Relatório Final do Trabalho de Projeto apresentado à

Escola Superior de Tecnologia e Gestão

Instituto Politécnico de Bragança

Para obtenção do grau de Mestre em

Engenharia Industrial

Trabalho realizado sob a orientação de

Professor Doutor José Augusto Carvalho

2012/2013

Agradecimentos

A realização desta dissertação, inserida no plano curricular do Mestrado de Engenharia Industrial, só foi possível com o apoio de algumas pessoas a quem quero manifestar o meu agradecimento.

Agradeço em especial ao meu orientador Professor Doutor José Augusto Carvalho, pela sua excelente supervisão, disponibilidade e empenho demonstrado.

Agradeço aos meus colegas por todos os bons momentos passados, camaradagem e amizade que sempre demonstraram e que nunca esquecerei.

Por último, mas não menos importante, gostaria de agradecer aos meus familiares e amigos, pelo apoio e encorajamento neste meu percurso académico. E um especial obrigado à minha namorada, Alexandra, pela ajuda, palavras de incentivo e confiança.

Resumo

Ao longo dos anos tem-se verificado um desenvolvimento muito significativo no domínio da domótica. Actualmente, ela é uma realidade presente nos edifícios, pois uma instalação de domótica permite aos utilizadores um maior conforto, segurança e eficiência energética.

Esta dissertação propõe uma utilização racional do aquecimento nos quartos de uma residência do Instituto Politécnico de Bragança, com o objetivo de combater o desperdício energético. Para esse efeito será utilizado um sistema domótico EIB/KNX.

Pretende-se desenvolver uma aplicação suportada por vários protocolos de comunicação. A um nível superior, será implementada uma interface gráfica, para um computador, que irá comunicar com um dispositivo intermédio, com um controlador KNX-IP, utilizando para este efeito o protocolo de comunicação Modbus-TCP/IP. O controlador actua como supervisão da instalação de domótica, que é responsável pelo aquecimento dos quartos. Assim, e através da interface gráfica, o utilizador é capaz de definir perfis de aquecimento, assim como fornecer ao controlador a informação relativa à ocupação dos quartos, permitindo uma gestão racional do aquecimento da residência de estudantes.

Palavras-chave: Domótica, EIB/KNX, Modbus.

Abstract

Over the years there has been a very significant development in the field of home automation. Currently, it is a reality in buildings, as an installation of home automation allows users greater comfort, safety and energy efficiency.

This paper offers a rational use of heating in the rooms of a residence at the Polytechnic Institute of Bragança, in order to reduce energy waste. For this purpose, EIB/KNX will be the home automation system used.

It is intended to develop an application supported by various communication protocols. At a higher level, it will be implemented a graphical user interface for a computer, which will communicate with a controller KNX-IP, using for this purpose the Modbus-TCP/IP communication protocol. The controller acts as a supervision of the home automation installation, which is responsible for heating the rooms. Thus, through the GUI, the user is able to set heating profiles, as well as provide the driver information on the occupation of the rooms, allowing a rational heating of the residence.

Keywords: Home Automation, EIB/KNX, Modbus

Conteúdo

1	Introdução.....	1
2	Estado de Arte	4
2.1	Tecnologia de domótica.....	4
2.2	EIB/KNX.....	6
2.2.1	Introdução	6
2.2.2	Topologia da Rede EIB/KNX.....	7
2.2.3	Modos de configuração.....	9
2.2.4	Comunicação.....	10
2.3	Eficiência Energética em edifícios com KNX.....	11
2.4	Definição do problema	16
3	Caso de estudo	18
3.1	Arquitectura da solução	19
3.2	Controlador.....	20
3.3	Gateway.....	20
3.4	Modbus	21
3.4.1	Estrutura das mensagens	21
3.5	KNXnet/IP	25
3.5.1	Módulos do KNXnet/IP	26
3.6	Interface gráfica.....	28
3.7	Software ETS.....	29
3.8	Software CoDeSys.....	30

4	Implementação Prática	32
4.1	Domótica	32
4.2	Controlador KNX-IP	36
4.3	Interface gráfica em Labview	39
5	Conclusão	45
A	Anexo A	1

Lista de Tabelas

Tabela 2.1: Camadas do modelo OSI	10
Tabela 3.1: Estrutura típica de uma mensagem de Modbus-TCP/IP	22
Tabela 3.2: Exemplo das funções mais comuns do protocolo Modbus	23
Tabela 4.1: Tabela o perfil de cada quarto em função da variável “conforto_quarto”	35
Tabela 4.2: Mensagem Modbus-TCP/IP “1234 0000 0006 FF06 0301 00F1”	38
Tabela 4.3: Endereços de memória do controlador e a sua função.....	39

Lista de Figuras

Figura 2.1: Rede EIB/KNX. [1]	6
Figura 2.2: Topologia de uma rede EIB/KNX. [1]	7
Figura 2.3: Instalação de uma rede EIB/KNX. [1]	8
Figura 2.4: Estrutura de uma rede EIB/KNX [5]	8
Figura 2.5: Modelo do Sistema KNX [6]	9
Figura 2.6: Alguns exemplos onde o sistema EIB/KNX foi implementado [9]	12
Figura 2.7: Poupança potencial de energia com controlo automático de estores [8]	13
Figura 2.8: Poupança potencial de energia na iluminação controlada por presença [8]	13
Figura 2.9: Salas de aula do ZIMT, 1º andar. [11]	14
Figura 2.10: Comparação do consumo energético em aquecimento [11]	15
Figura 3.1: Arquitectura da solução	19
Figura 3.2: Exemplo de duas redes distintas ligadas por uma <i>gateway</i>	20
Figura 3.3: Gama das funções do protocolo. [12]	24
Figura 3.4: Envio de mensagem sem erro de recepção. [10]	24
Figura 3.5: Envio de mensagem com erro de recepção. [10]	25
Figura 3.6: Modelo em camadas do protocolo KNX e KNXnet/IP [6]	26
Figura 3.7: Exemplo de configuração de rede KNXnet/IP implementando tunnelling [6]	27
Figura 3.8: Exemplo de configuração de rede KNXnet/IP implementando routing. [6]	28
Figura 3.9: Exemplo de um VI em Labview	29
Figura 3.10: Interface gráfica do ETS 4	30
Figura 4.1: Unidade KNX de controlo de temperatura para quarto	33

Figura 4.2: Actuador de aquecimento à esquerda e accionamento termoeléctrico à direita.	33
Figura 4.3: Group Addresses do sistema.	34
Figura 4.4: <i>Front Panel</i> da aplicação em Labview.....	39
Figura 4.5: Exemplo de uma aplicação cliente-servidor, em Labview, usando Modbus. .	40
Figura 4.6: Programa, em Labview, que envia quais os quartos que estão ocupados.	41
Figura 4.7: Programa, em Labview, que envia a “hora em que o aquecimento se irá ligar, perfil de conforto” e a “temperatura desejada no perfil de redução nocturna”.....	42
Figura 4.8: Monitorização do sistema.....	43

Lista de Abreviações e Acrónimos

Bus	Barramento.
EIB	European Installation Bus.
ETS	Engineering Tool Software.
FBD	Blocos lógicos.
IL	Lista de Instruções
IP	Internet Protocol.
IR	Infravermelhos.
LabVIEW	Laboratory Virtual Instruments Engineering Workbench.
LAN	Local Area Network.
LD	Diagrama de Blocos
OSI	Open Systems Interconnection.
PID	Proportional Integral Derivative
PL	Powerline.
PLC	Programmable logic controller.
RF	Radiofrequência.
SFC	Função gráfica sequencial.
ST	Texto Estruturado.
TP	Twisted Pair (cabo de cobre de par entrançado).
VI	Virtual Instrument.
WLAN	Wireless Local Area Network.

Capítulo 1

1 Introdução

Com o propósito de melhorar a qualidade de vida e aumentar o bem-estar, foi criada a Domótica. O termo surge da junção da palavra “Domus”, que significa residência, com a palavra ”Telemática”, que significa electrónica e informática. A Domótica é conhecida como uma ciência moderna de engenharia das instalações em edifícios inteligentes e é uma tecnologia que engloba quatro vetores fundamentais: eficiência energética, segurança, comunicação e conforto.

Para alcançar tudo isto, é necessário um conjunto de dispositivos, que são distribuídos num edifício em função das necessidades dos seus proprietários. Estes dispositivos devem estar ligados entre si e podem ser sensores, actuadores, controladores ou interfaces, podendo-se a isto chamar de rede de domótica.

Inicialmente, a maioria das redes de domótica criadas consistia em produtos proprietários, e outros que ao longo dos tempos se tornaram normas. O EIB/KNX é actualmente uma norma europeia com bastante sucesso.

Este projeto tem por objetivo em gerir, de forma eficiente, o aquecimento das residências de estudantes do Instituto Politécnico de Bragança, utilizando a domótica para esse efeito. Recorrendo à tecnologia EIB/KNX, pretendem-se criar vários perfis de aquecimento para os diversos quartos, de modo a garantir o conforto no interior dos mesmos.

Durante os dias mais frios o aquecimento dos quartos das residências é efectuado 24 horas por dia, o que resulta num elevado consumo energético. O objetivo desta tese é

corrigir este gasto. Para isso, e sabendo sempre se cada quarto está ocupado ou livre, o utilizador é capaz de programar, através de uma aplicação, os perfis de aquecimento para cada quarto. Pretende-se ainda adicionar um sensor nas janelas dos quartos, para que seja sempre detectada a abertura das janelas e, se for o caso, desligar o aquecimento do quarto onde estas estejam abertas.

Para além deste capítulo de introdução o documento está estruturado da seguinte forma:

O capítulo 2 descreve o estado de arte da domótica. Será efectuado um levantamento bibliográfico sobre as várias tecnologias de domótica existentes no mercado e também exposto e definido o problema tratado ao longo desta tese.

No capítulo 3 é descrito o estudo das tecnologias adoptadas para suportar este trabalho, nomeadamente o Modbus-TCP/IP e KNXnet/IP. Também irá ser introduzido o controlador KNX-IP.

Quanto ao capítulo 4, será apresentada a implementação prática e também os pormenores mais importantes relativos à parte de domótica, do controlador KNX-IP e da interface gráfica.

Por fim, no capítulo 5, são apresentadas as principais conclusões desta dissertação.

Capítulo 2

2 Estado de Arte

Neste capítulo serão discutidas algumas das tecnologias de domótica mais importantes no mercado. Também vai ser explicado, com algum pormenor, o funcionamento dos sistemas KNX, visto que foi a tecnologia escolhida para a realização desta tese.

2.1 Tecnologia de domótica

A domótica tem vindo a expandir-se ao longo dos anos devido ao grande conforto que traz aos seus utilizadores. É de referir que se tem verificado um aumento do número de empresas especializadas neste tipo de tecnologia. Esta expansão deve-se, em grande parte, ao crescimento das novas tecnologias, como por exemplo a evolução da computação móvel.

Inicialmente, os sistemas domóticos foram dominados por produtores proprietários (produtos de uma determinada entidade/marca), não existindo normas (*standards*) para a criação de sistemas compostos por produtos de vários fabricantes. Com o decorrer dos anos, foram surgindo movimentos de uniformização, com o objetivo de permitir a compatibilidade entre produtos de diferentes fabricantes, promover reduções de custos, e consequentemente permitir uma expansão mais rápida destes sistemas. [1]

A escolha de produtos proprietários acarreta alguns riscos, pois a empresa pode abandonar o negócio. Se ocorrer alguma avaria, não será possível substituir o produto,

devido à não comercialização deste, o que representa um risco dos investimentos económicos. Além disso, a tecnologia pode ficar obsoleta. Em contrapartida, os produtos normalizados são fabricados por várias empresas, onde os equipamentos são compatíveis, independentemente da marca.

Alguns sistemas de domótica que dominam o mercado são o X10, CEBus e o EIB/KNX, sendo este último o escolhido para a realização desta tese, uma vez que é a tecnologia mais madura, normalizada internacionalmente e com uma grande implantação no mercado.

A tecnologia X10 foi desenvolvida nos anos 70, sendo o seu protocolo normalizado em 1997. O seu baixo preço e grande variedade de equipamentos tornam o X10 o protocolo de domótica mais divulgado no mundo. No mercado norte-americano é até possível adquirir equipamentos num supermercado. [3]

O X10 usa a rede de distribuição de energia eléctrica como meio de comunicação entre dispositivos, sendo esta a sua maior vantagem, pois dispensa o uso de cabos adicionais. Uma desvantagem prende-se com o facto de a comunicação ser relativamente lenta, não permitindo activação simultânea de diversos dispositivos e tendo também um número limitado de funções. [2]

Relativamente ao CEBus, este é um protocolo muito poderoso e complexo que foi criado em 1984 e normalizado em 1995. Este protocolo utiliza as camadas Física, Ligação de Dados, Rede e Aplicação do modelo OSI. Apresenta ainda algumas semelhanças com outras tecnologias nomeadamente o EIB/KNX. [2]

Apesar de ser capaz de suportar vários meios de comunicação, como rede eléctrica, cabo coaxial, infravermelhos, fibra óptica, cabo de cobre de par entrançado e radiofrequência, a tecnologia CEBus apresenta custos elevados e possui uma cota de mercado bastante baixa, devendo-se isso, em parte, à concorrência com o X10, que possui preços bastante mais acessíveis e tem uma larga divulgação.

A tecnologia KONNEX (KNX) foi criada a 14 de Abril de 1999 através da junção dos sistemas EIB, Batibus e EHS. Inicialmente esta tecnologia tinha como principal objetivo a criação de uma única norma europeia para a automação de edifícios. [4]

O KNX apresenta um elevado custo de hardware e software. Contudo, esta é uma tecnologia totalmente aberta e garante um funcionamento com produtos de diferentes marcas e modelos desde que sejam certificados e identificados pelo logótipo KNX.

2.2 EIB/KNX

2.2.1 Introdução

O sistema EIB/KNX foi desenhado de forma a poder ser instalado tanto em edifícios de grandes dimensões, como edifícios de escritórios, escolas, hospitais e fábricas, como em residências ou edifícios de habitação. A sua finalidade é monitorizar e controlar sistemas de iluminação, aquecimento, ventilação, ar condicionado, balanceamento de carga ou sistemas de vigilância. [1]

A figura seguinte mostra alguns exemplos de dispositivos comuns onde o sistema EIB/KNX pode ser implementado.

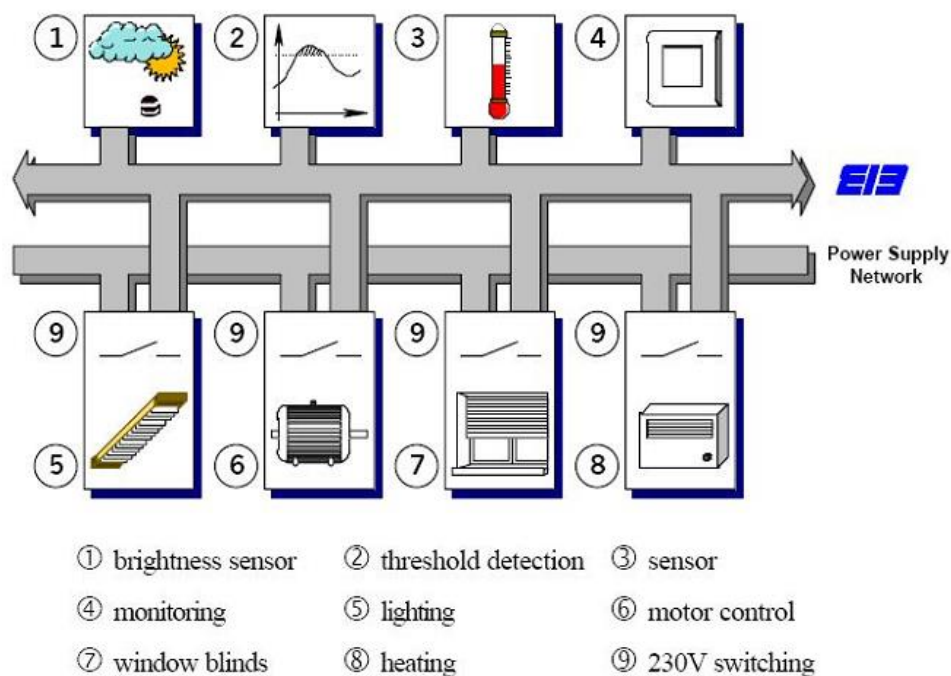


Figura 2.1: Rede EIB/KNX. [1]

O EIB/KNX possui uma arquitectura descentralizada, ou seja, os dispositivos podem ser emissores ou receptores que comunicam directamente entre si, sem

necessidade de hierarquia ou supervisão da rede. Por outro lado, pode ser ligado à rede uma aplicação de controlo para gestão do sistema, como por exemplo um computador, que pode ser ligado à rede a partir de qualquer localização, permitindo assim uma gestão centralizada do sistema.

2.2.2 Topologia da Rede EIB/KNX

O sistema EIB/KNX suporta diferentes meios físicos de comunicação, como *Twisted Pair* (cabo de cobre de par entrançado), radiofrequência, infravermelhos, entre outros. É ainda possível instalar *gateways* para outras redes, como *Ethernet* ou *WLAN*.

As redes de par entrançado são o meio de transmissão de dados mais comum num sistema EIB/KNX. Os segmentos da rede podem possuir uma topologia arbitrária (linear, em estrela, em árvore, em anel, ou combinações destas, como é ilustrado na figura 2.2) constituída por secções individuais desde que os requisitos eléctricos (comprimento dos segmentos) não sejam excedidos [1].

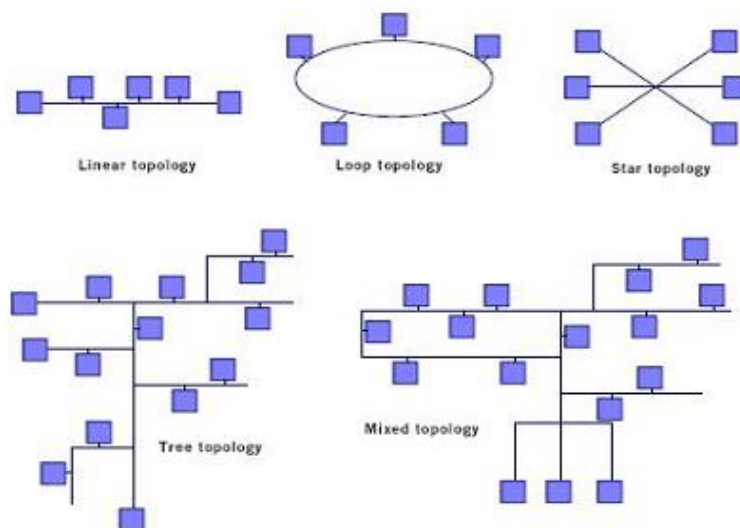


Figura 2.2: Topologia de uma rede EIB/KNX. [1]

O meio físico de comunicação radiofrequência ou infravermelhos, não é necessário estas topologias acima referidas, visto que é uma comunicação sem fios.

A rede EIB/KNX necessita de pelo menos um par de cabos para operar. Um dos cabos é usado para transmissão de informação, sendo o segundo necessário para fornecer energia complementar aos dispositivos. A instalação de uma rede EIB/KNX é semelhante à instalação de uma rede eléctrica (ver figura 2.3).

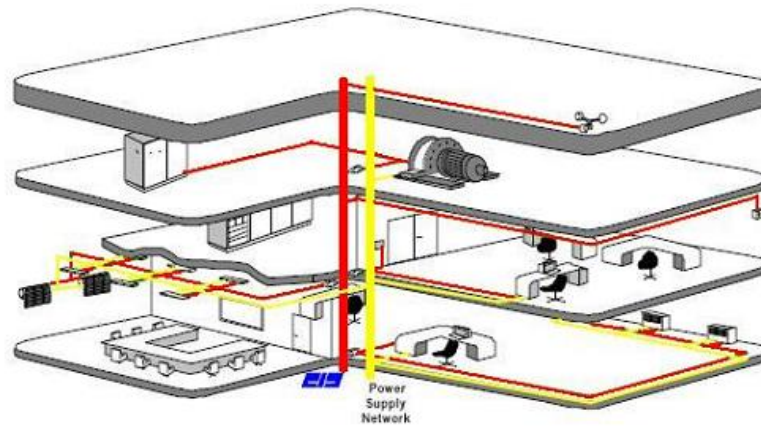


Figura 2.3: Instalação de uma rede EIB/KNX. [1]

Um sistema KNX pode possuir 16 áreas de controlo e pode ter até 16 linhas, permitindo ligar até 65536 (2^{16}) dispositivos. Na Figura 2.4 pode ser visualizada a topologia desta tecnologia. [4][5]

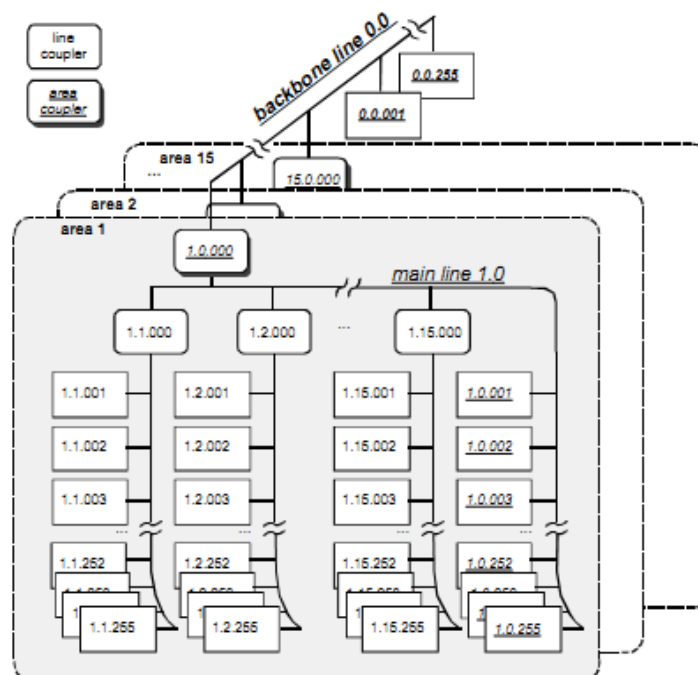


Figura 2.4: Estrutura de uma rede EIB/KNX [5]

2.2.3 Modos de configuração

Existem três modos de configuração: S-mode , E-mode e A-mode. A figura 2.5 ilustra o sistema de base do EIB/KNX. Para todos os modos de configuração, existe um *software* criado pela KONNEX para configurar os dispositivos EIB/KNX denominado por ETS (Engineering Tool Software).

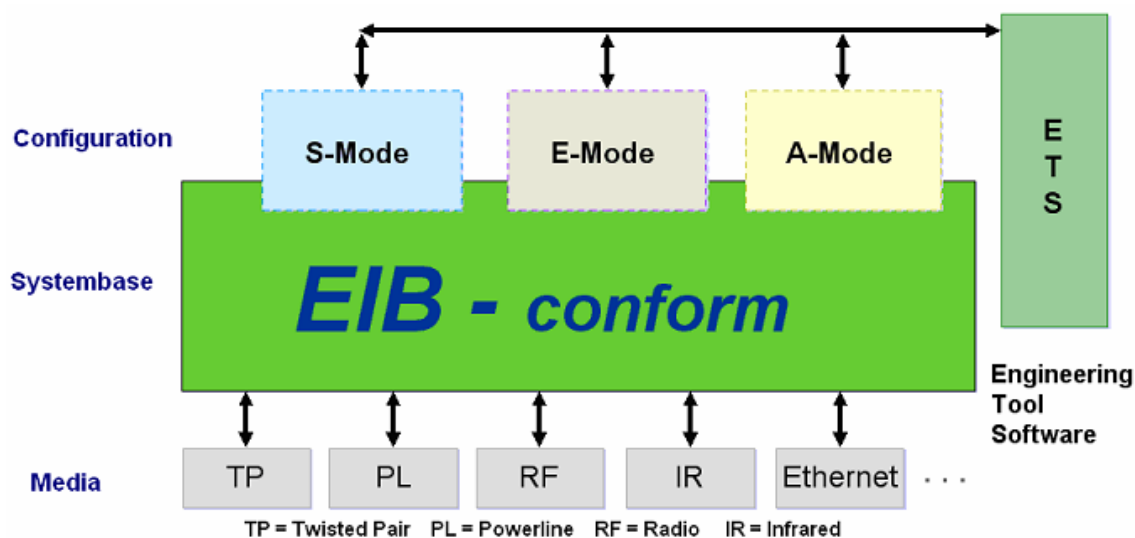


Figura 2.5: Modelo do Sistema KNX [6]

A imagem ilustra também diferentes meios físicos que o sistema EIB/KNX pode utilizar, como por exemplo: TP (twisted pair), PL (powerline), RF (radiofrequencia), IR (infravermelhos), *Ethernet*, entre outros.

O S-mode ou System mode é o mais versátil e mais usual de encontrar, uma vez que apresenta uma implementação simples. A complexidade de configuração deixa de estar no dispositivo EIB/KNX, passando para um software de configuração como o ETS. Este tipo de configuração exige que o ETS tenha uma base de dados dos dispositivos EIB/KNX de cada fabricante, com as respectivas funcionalidades. Os fabricantes são responsáveis por criarem os *templates* dos seus dispositivos e por os atualizarem. Normalmente, na compra de um dispositivo EIB/KNX, o fabricante fornece ao cliente o *template* do mesmo para inserir no ETS e assim poder configurar o dispositivo [6].

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

Em comparação com o System mode, o E-mode, ou Easy mode, apresenta um número de funções mais reduzido. Os dispositivos vêm pré-programados para executarem uma certa função e têm de ser configurados no local da instalação.

O A-mode, contrariamente aos outros, é destinado a instalações muito simples, podendo estas ser efectuadas por um utilizador inexperiente em sistemas EIB/KNX. Os dispositivos incluem mecanismos que, autonomamente, permitem localizar outros dispositivos. Têm também a capacidade de adquirir os seus próprios endereços físicos.

2.2.4 Comunicação

O sistema de comunicação EIB/KNX é constituído por uma pilha protocolar semelhante ao modelo OSI. Das sete camadas do modelo OSI, apenas cinco são implementadas para o correto funcionamento do sistema, sendo elas a camada Física, Ligação de Dados, Rede, Transporte e por último a de Aplicação (ver tabela 2.1)

Tabela 2.1: Camadas do modelo OSI

Camada	Modelo OSI	EIB/KNX
7	Camada de aplicação	Camada de aplicação
6	Camada de apresentação	
5	Camada de sessão	
4	Camada de transporte	Camada de transporte
3	Camada de rede	Camada de rede
2	Camada de ligação de dados	Camada de ligação de dados
1	Camada física	Camada física

A camada física define o tipo de meio físico do modelo, o sistema EIB/KNX possui um mecanismo que permite ser independente do meio físico. Como já referido anteriormente, este modelo suporta TP, PL, RF, IR, *Ethernet*, entre outros.

A camada de ligação de dados garante a transmissão e recepção de dados e controlo de fluxo entre dispositivos EIB/KNX. A ligação de dados também é responsável por corrigir possíveis erros que possam existir no nível físico.

A camada de rede faz o elo de ligação entre a camada de ligação de dados e a camada de transporte e é responsável pelo endereçamento dos tramas, determinando qual o percurso que eles devem tomar.

A camada de transporte efectua a interligação entre a camada de rede e a camada de aplicação. Implementa vários modos de comunicação como por exemplo:

- Um-para-muitos – não orientado à conexão (*multicast*);
- Um-para-todos – não orientado à conexão (*broadcast*);
- Um-para-um – não orientado à conexão;
- Um-para-um – orientado à conexão.

A camada de aplicação corresponde à última camada superior do modelo OSI. Permite efetuar uma grande variedade de serviços de configuração e controlo de equipamentos EIB/KNX, consoante os quatro modos de comunicação da camada de transporte anteriormente mencionados.

2.3 Eficiência Energética em edifícios com KNX

Actualmente cerca de 40% da energia consumida nos países industrializados deve-se à utilização de aquecimento, ar condicionado e iluminação nos edifícios residenciais e escritórios. Uma boa utilização do controlo de edifícios inteligentes contribui para a optimização da eficiência energética. [8]

Em comparação com as instalações elétricas tradicionais, os sistemas de controlo de edifícios inteligentes oferecem vantagens significativas. Com eles é possível controlar a poupança energética do aquecimento, ar condicionado e ventilação, assim como efetuar

uma regulação constante da iluminação, permitindo aproveitar a luz natural e ativar a iluminação apenas na percentagem necessária.

Os sistemas KNX das habitações têm a capacidade de reagir de forma rápida e inteligente em qualquer situação e a qualquer momento, independentemente de haver, ou não, pessoas no interior. É capaz de prevenir perigos contra a integridade física, tais como a detecção de intrusos, incêndios, fumo, ou efetuar uma simulação de presença controlando, por exemplo, os estores de x em x tempo.

Devido à flexibilidade do sistema KNX, esta tecnologia pode ser implementada em vários tipos de edifícios, como por exemplo: escritórios, aeroportos, escolas, residências, entre outros:

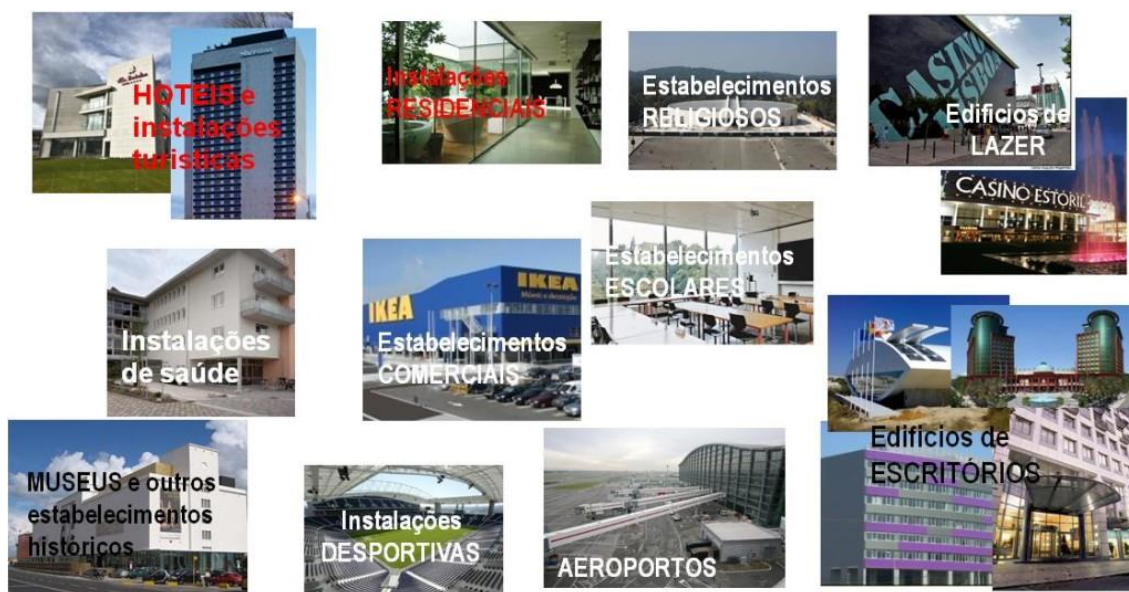


Figura 2.6: Alguns exemplos onde o sistema EIB/KNX foi implementado [9]

Em 2008, a Universidade de Ciências Aplicadas de Biberach efectuou o estudo “Poupanças energéticas e eficiência potencial através da utilização de tecnologia de bus assim como a automatização de habitações e edifícios”. Neste estudo foi utilizado um sistema KNX da marca ABB. O projeto de investigação foi realizado na Alemanha, num escritório de um edifício clássico, tendo como objetivo colocar em prática a diretiva CE 2002/91/CE (Diretiva de Eficiência Energética dos Edifícios). A principal exigência desta diretiva é a emissão de um certificado energético, onde se detalha tanto o consumo energético do edifício como a análise das poupanças possíveis. [8]

Os resultados deste estudo, ilustrados na figura 2.7, mostram que é possível poupar de 20% a 30% de energia com o controlo automático de estores.

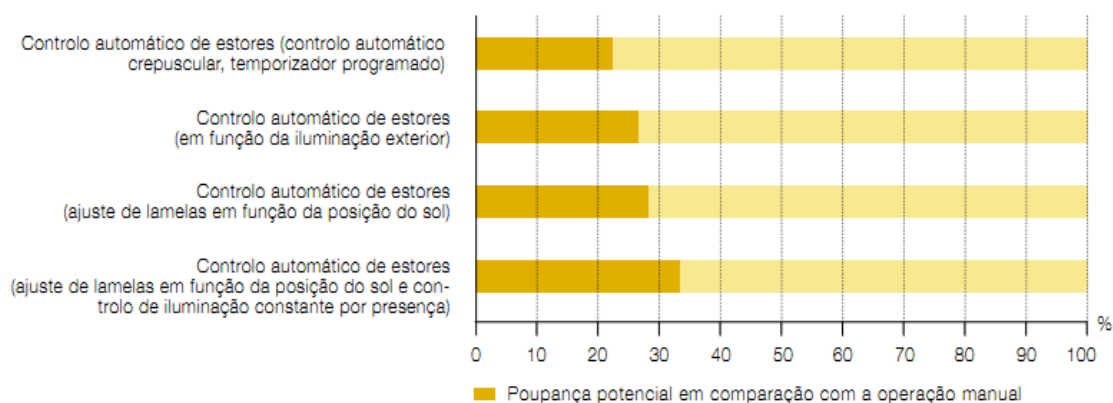


Figura 2.7: Poupança potencial de energia com controlo automático de estores [8]

Pode-se concluir que, dos quatro controlos automáticos de estores apresentados, o que apresenta maior potencial de poupança é “ajuste de lamelas em função da posição do sol e controlo de iluminação constante por presença”.

Ainda no mesmo escritório, foi testada a eficiência da iluminação controlada por presença. Os resultados podem ser observados na figura seguinte.

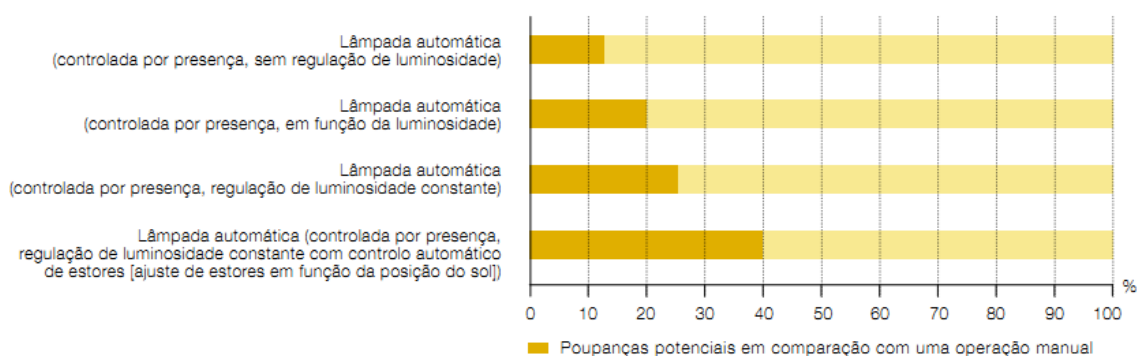


Figura 2.8: Poupança potencial de energia na iluminação controlada por presença [8]

Verifica-se que a poupança energética das lâmpadas controladas por presença, sem regulação de luminosidade, é de cerca de 10%. Já as lâmpadas com regulação de luminosidade constante, com controlo automático de estores (ajuste de estores em função da posição do sol), atinge uma poupança de 40%.

Conclui-se que é possível atingir um potencial de poupança energético significativo através da utilização de tecnologia bus, o KNX, neste caso. O grau de poupança potencial depende da função específica de cada automação e da correcta combinação de cada função.

Outro estudo realizado pela Universidade de Ciências Aplicadas de Bremen, sobre a poupança energética com o KNX, teve como objetivo controlar o aquecimento nas salas de aulas da própria Universidade, no centro ZIMT (Information and Media Technology), situado no primeiro andar da Universidade. [10]

No seguimento das experiências foram escolhidas duas salas de aulas idênticas, uma delas equipada com termostatos normais para aquecimento (sala 123, na figura 2.9) e outra com um sistema KNX (sala 122, na figura 2.9):

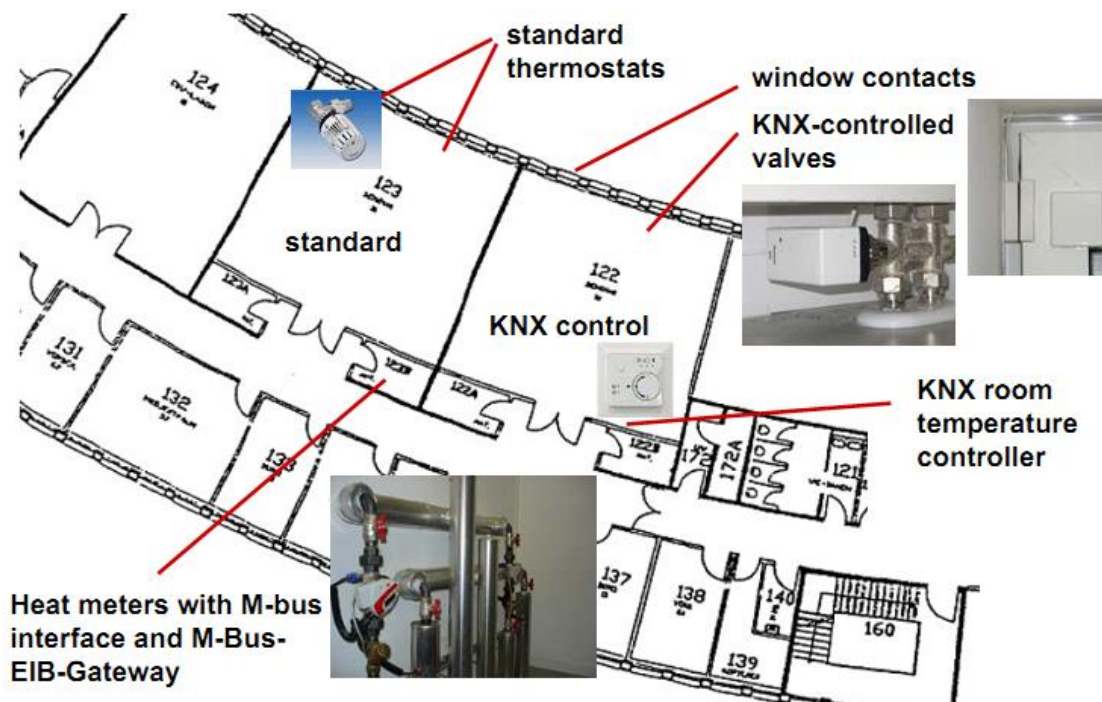


Figura 2.9: Salas de aula do ZIMT, 1º andar. [11]

Na sala 122 foram instaladas válvulas para o aquecimento, interruptores para as janelas, bem como um sistema para o controlo de temperatura e de aquecimento, que é também capaz de efetuar medições do consumo energético do aquecimento.

Os resultados obtidos foram satisfatórios. Na figura 2.10 é apresentado o consumo energético do aquecimento das duas salas ao longo do tempo. É de salientar que as salas só começaram a ser utilizadas em pleno a partir do segundo semestre de 2004.

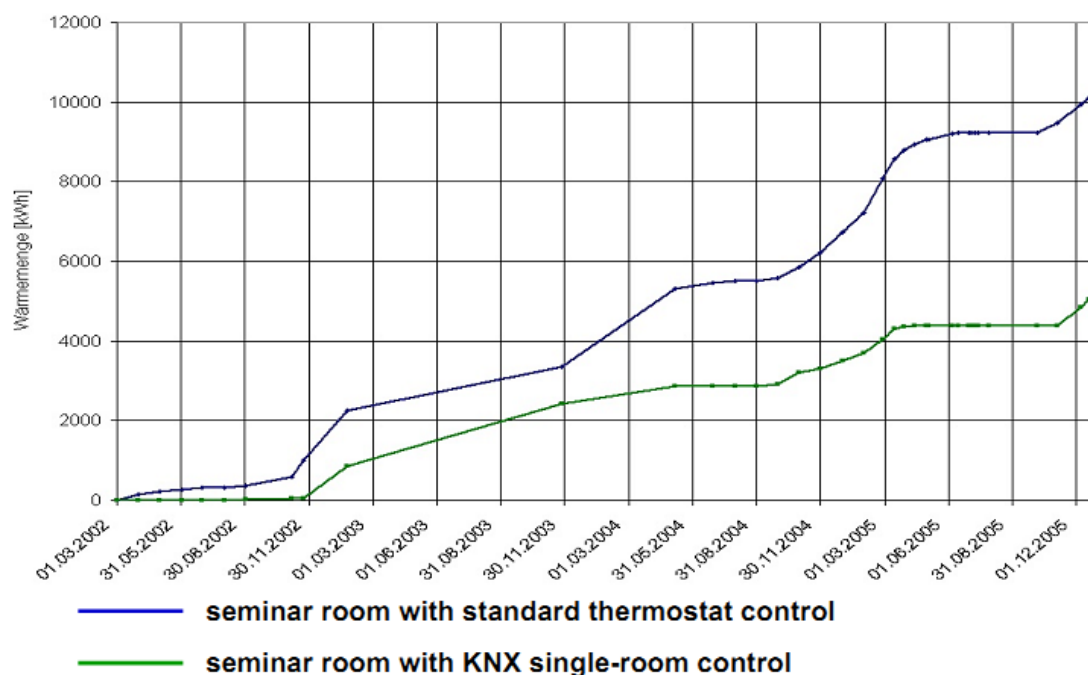


Figura 2.10: Comparação do consumo energético em aquecimento [11]

Verifica-se que a sala onde foi implementado o sistema KNX atinge um rendimento próximo dos 50%. Este rendimento elevado deve-se ao controlo racional da energia, pois, estando as janelas abertas, as válvulas desligam automaticamente, não existindo assim o aquecimento desnecessário que se verifica na outra sala.

Apesar de um sistema KNX ser um investimento caro, ele pode atingir níveis de rendimento bastantes satisfatórios, reduzindo assim os custos energéticos e, por conseguinte, obtendo um retorno a médio prazo.

Analisando as várias tecnologias de domóticas apresentadas, o KNX demonstrou ser a rede de domótica que melhor se ajusta aos objetivos propostos para esta dissertação, pois nos estudos apresentados verificou-se que esta tecnologia consegue atingir uma elevada eficiência energética em aplicações de aquecimento. Por outro lado, trata-se do standard europeu para instalações deste tipo e, desta forma, beneficia de todas as vantagens associadas a um produto normalizado, suportado por um elevado número de fabricantes.

2.4 Definição do problema

Durante os dias mais frios, nas residências escolares do Instituto Politécnico de Bragança, o aquecimento é efectuado 24h por dia em todos os quartos dos vários andares. Desta forma verifica-se que há um elevado consumo de energia neste edifício.

Pretende-se com este projeto melhorar a eficiência energética desta residência e utilizar a energia de uma forma racional, sem gastos desnecessários. Um sistema KNX, tal como já foi constatado anteriormente, pode resolver este problema.

Pretende-se instalar um sistema KNX nesta residência, que fará o controlo do aquecimento em todos os quartos. Também é necessário o desenvolvimento de uma interface gráfica que irá interagir com a rede KNX. Esta interface gráfica deverá ser intuitiva e simples de trabalhar, de modo que um utilizador comum possa efetuar o controlo da rede.

O sistema deverá ser capaz de efetuar um controlo individual de cada quarto, com base na informação da ocupação do quarto, nomeadamente com informação se os alunos foram ou não de fim-de-semana, e com isto configurar vários parâmetros, como por exemplo as temperaturas desejadas, e escolher vários perfis de aquecimentos adequados a cada caso.

Capítulo 3

3 Caso de estudo

Neste capítulo será descrito o caso de estudo usado nesta dissertação. Será também discutido o problema e as tecnologias de suporte para desenvolvimento da aplicação de controlo.

O caso de estudo desta tese consiste em gerir o aquecimento de uma residência do Instituto Politécnico de Bragança, dotada de cerca de 150 quartos, com vários andares/blocos. Nesta residência o aquecimento é efectuado com radiadores de água quente.

O cenário de ocupação da residência é bastante heterogénea, embora seja possível definir alguns perfis de utilização. As aulas não começam nem acabam ao mesmo tempo para todos os alunos, no entanto, a maioria deles dirige-se para a escola entre as 8.30h e as 10.30h, voltado pelas 14h, 16h ou 18h. O período entre as 18 e 23 horas é o preferido dos alunos para o estudo, ocupando estes a sala de estudo da residência.

No fim-de-semana verifica-se uma menor afluência na residência, visto que a grande parte dos alunos a abandona na sexta-feira, voltando ao domingo.

Para controlar o sistema de aquecimento desta residência propõe-se a solução apresentada no subcapítulo 3.1.

3.1 Arquitectura da solução

Como já foi referido anteriormente, pretende-se efetuar um controlo do aquecimento de uma residência do IPB. O sistema deverá incluir uma interface gráfica, um controlador e a parte de domótica (KNX).

A interface gráfica deverá ser uma aplicação, para um computador, que irá comunicar com o controlador através do protocolo Modbus-TCP/IP. Mais à frente irá ser explicado como este protocolo funciona.

O controlador tem como objetivo estabelecer a comunicação entre a interface gráfica e os dispositivos KNX. Para isso é necessário criar uma *gateway* e utilizar os protocolos Modbus e KNXnet/IP.

Quanto à parte de domótica, ela é que irá efetuar o controlo do aquecimento nos radiadores nos quartos na residência.

Na figura 3.1 pode ser visualizado, de uma forma muito resumida, como este projeto funciona.

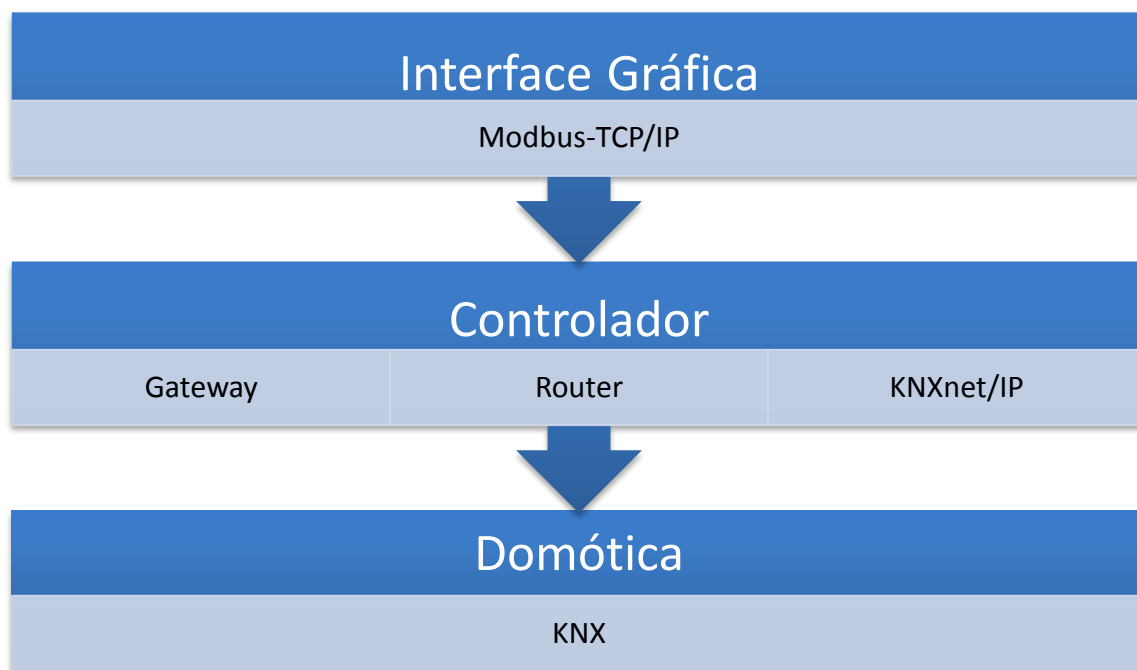


Figura 3.1: Arquitectura da solução.

3.2 Controlador

Para este trabalho, o controlador deverá ter suporte para KNX-IP, ou seja, o controlador deve possuir uma interface *ethernet* e um *router* interno, capaz de comunicar com os dispositivos de domótica (KNX) através de uma *gateway* e utilizando o protocolo KNX-IP.

O controlador está equipado com um servidor Modbus no qual estão mapeados em memória um conjunto de registos de escrita e registos de leitura.

Através desses registos é possível a troca de informação entre a interface gráfica e o controlador e, por conseguinte, com os dispositivos de domótica.

O controlador *KNX-IP Wago 750-849* é semelhante a um PLC (*Programmable logic controller*) bastante potente e é o controlador escolhido para este trabalho, uma vez que satisfaz todos os requisitos acima referidos.

3.3 Gateway

Uma *gateway* garante o caminho para duas ou mais redes distintas conseguirem comunicar entre si, tal como é ilustrado na figura 3.2:

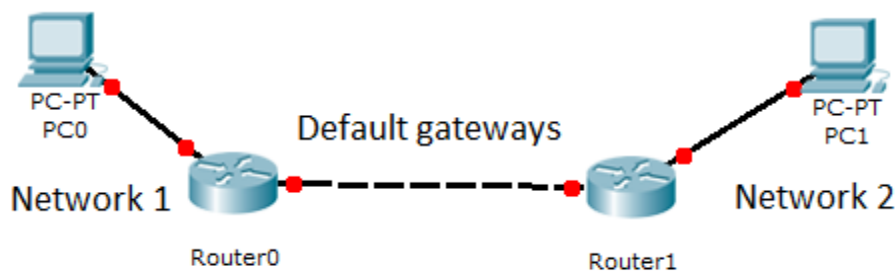


Figura 3.2: Exemplo de duas redes distintas ligadas por uma *gateway*.

Esta configuração, em que existe a comunicação entre duas redes distintas, é semelhante ao que se verifica neste projeto, isto é, o controlador deve possuir um *router* que, através de uma *gateway*, irá comunicar com o *router* do sistema KNX.

Para fazer chegar a informação da interface gráfica ao controlo do sistema é necessário escolher um protocolo de comunicação entre a interface gráfica e o controlador. Na secção seguinte irá ser explicado o modbus, que foi o protocolo de comunicação escolhido para este projeto.

3.4 Modbus

O Modbus foi pela primeira vez implementado para aplicações de redes de autómatos programáveis em 1979, desenvolvido pela Modicon. Actualmente, este é um protocolo aberto e bastante utilizado na automação industrial.

Posteriormente, em 1999, foi adicionado a norma Modbus-TCP/IP ao protocolo original e adaptando-o às necessidades actuais.

Este protocolo situa-se na camada Aplicação do modelo OSI. A porta 502 do TCP/IP está reservada para protocolo Modbus.

3.4.1 Estrutura das mensagens

O Modbus define uma arquitectura cliente/servidor em que o cliente é responsável pela organização da rede, pelo fluxo das mensagens e também pelo envio de mensagens aos servidores. Um servidor numa rede Modbus é, usualmente, um módulo de entradas e saídas.

Tendo em vista os objetivos deste projeto, apenas será explicada a estrutura de uma mensagem de Modbus por TCP/IP.

Uma mensagem Modbus-TCP/IP consiste no encapsulamento de pacote em segmentos TCP. Estas mensagens são compostas por um cabeçalho para indicar parâmetros de transmissão, um campo para indicar o código da função (ou function code) utilizada e um campo que inclui os dados que serão transmitidos. [13]

A tabela 3.1 mostra os campos existentes em pacotes Modbus-TCP/IP, sendo o cabeçalho composto pelos quatro primeiros campos.

Tabela 3.1: Estrutura típica de uma mensagem de Modbus-TCP/IP.

Cabeçalho				Função	Dados
Identificador de Transacção	Identificador de Protocolo	Comprimento	Identificador de Unidade		
2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	n Bytes

Os campos do Modbus-TCP/IP podem ser definidos como:

- **Identificador de Transacção** - este campo, com dois bytes, identifica a comunicação entre o servidor e o cliente Modbus. Cada requisição Modbus tem a sua resposta com o mesmo identificador de transacção.
- **Identificador de Protocolo** - este campo, com dois bytes, é utilizado para identificar o protocolo. Para utilizar o Modbus este campo deve conter o valor 0 (zero).
- **Comprimento** - este campo, com dois bytes, contém a quantidade de bytes dos campos seguintes a ele (Identificador de Unidade, Função e Dados).
- **Identificador de Unidade** - este campo, com um byte, é utilizado para identificar os dispositivos Modbus que estejam ligados na rede através de um *gateway*.
- **Função** - este campo representa o objetivo da mensagem.
- **Dados** - este campo é responsável pelos dados que serão enviados/recebidos.

O campo da função não é mais do que um código de um byte que identifica uma função que o destinatário deverá reconhecer. Na tabela seguinte pode-se visualizar uma lista com apenas algumas das funções disponíveis, bem como o respectivo código em hexadecimal, para este protocolo.

Tabela 3.2: Exemplo das funções mais comuns do protocolo Modbus

Descrição	Função (em hexadecimal)
Ler saída	0x01
Ler entrada discreta	0x02
Ler registo “holding”	0x03
Ler registo de entrada	0x04
Escrever saída simples	0x05
Escrever registo simples	0x06

Por exemplo, o código “0x01” especifica a função para ler o estado de uma saída do servidor. Já o código “0x05” serve para escrever numa saída.

Existem 3 tipos diferentes de funções sendo elas funções públicas, funções definidas pelo utilizador e funções reservadas.

- **Funções públicas** - São funções bem definidas pela norma e que têm de cumprir necessariamente a sua estrutura.
- **Funções definidas pelo utilizador** - São funções que, como o próprio nome indica, são implementadas pelo próprio utilizador, sem que se garanta a compatibilidade entre outras podendo mesmo existir funções diferentes com o mesmo código.
- **Funções reservados** - Funções actualmente utilizadas por algumas empresas e que detêm o direito exclusivo do seu uso. Como tal, não podem ser utilizadas pelo público em geral.

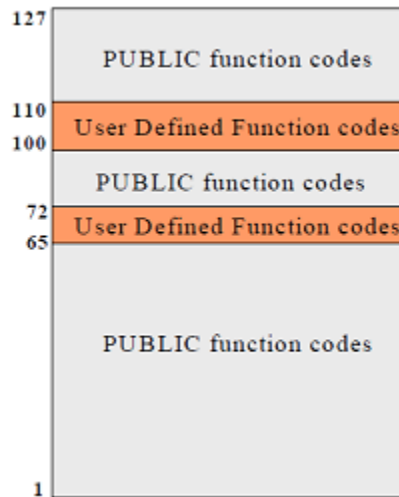


Figura 3.3: Gama das funções do protocolo. [12]

É importante referir que, para além destas 127 funções representadas na figura 3.3, existem ainda outras 127 funções que correspondem a diferentes incoerências presentes na mensagem recebida. Estas são denominadas por exceções onde o “function code” dessas mensagens é igual ao “function code” original, sem erro.

Isto significa que, em caso de erro na recepção do lado do servidor, uma função como a “0x01”, enviada pelo cliente, teria como resposta “0x81”. [10]

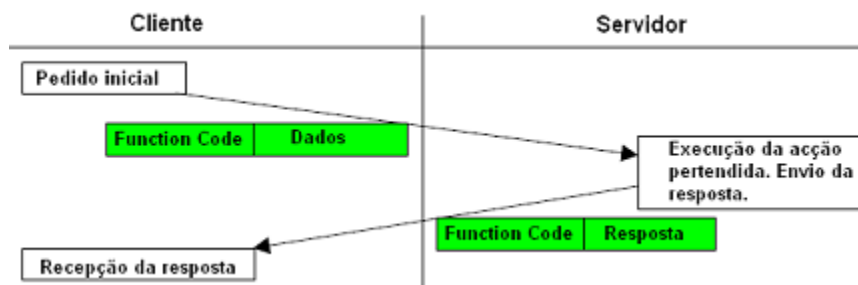


Figura 3.4: Envio de mensagem sem erro de recepção. [10]

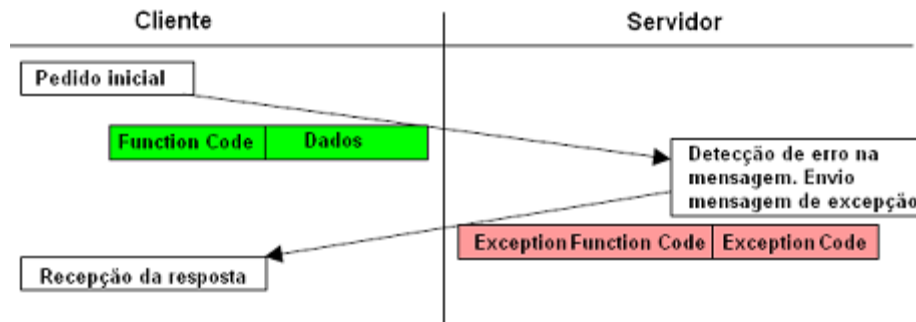


Figura 3.5: Envio de mensagem com erro de recepção. [10]

Na Figura 3.4 e na Figura 3.5 pode-se visualizar o esquema de troca de mensagens entre um cliente/servidor Modbus, sem e com erro de recepção, respectivamente.

Uma mensagem de Modbus, em traços gerais, é composta por dois campos. O primeiro, o “function code” é responsável por chamar uma função específica do lado do servidor. [10] O segundo campo é reservado a dados adicionais relativos à respectiva função. Este campo pode ser utilizado para enviar dados específicos a cada função ou para reportar erros de leitura na mensagem recebida por parte do servidor. Na Figura 3.5 é apresentado o esquema de uma transacção, com as verificações necessárias, que o servidor tem que percorrer, de modo a ser garantida a integridade da mensagem recebida e, posteriormente, o envio de uma resposta válida.

3.5 KNXnet/IP

Actualmente é bastante usual um edifício incluir uma rede IP com ligação a internet ou mesmo uma LAN. Como estas Tecnologias de Informação e Comunicação são bastante baratas, a Konnex (KNX) criou o protocolo KNXnet/IP que permite utilizar uma rede IP para efetuar o controlo de uma rede KNX.

Este protocolo situa-se na camada Física do modelo em camadas do KNX e define a integração do protocolo KNX sobre o protocolo de rede IP como se pode visualizar na figura seguinte. [6]

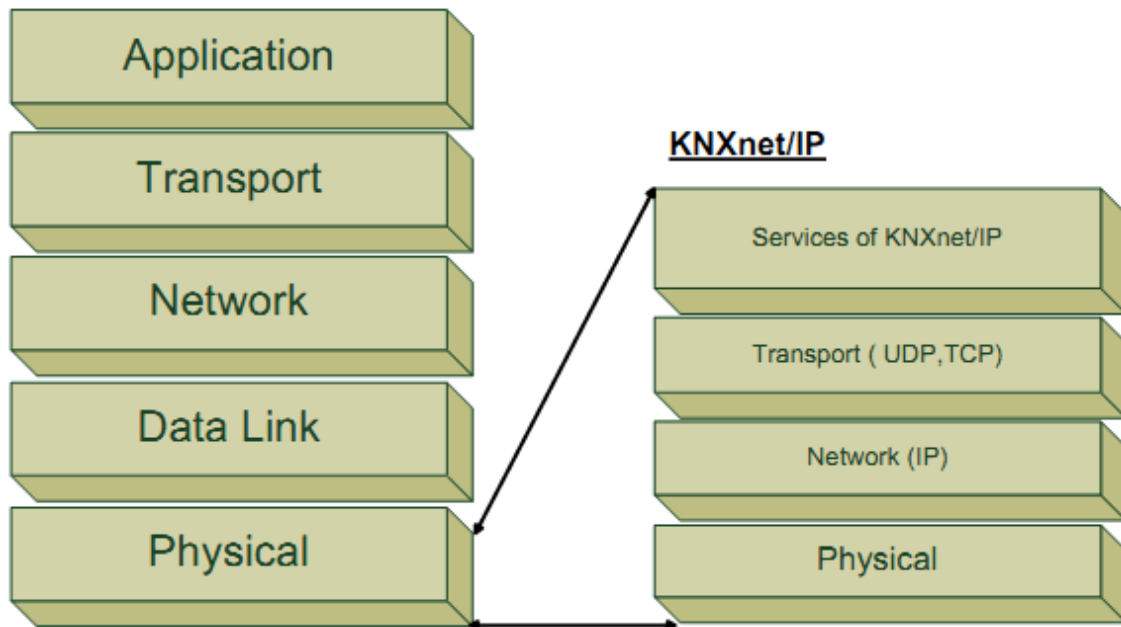


Figura 3.6: Modelo em camadas do protocolo KNX e KNXnet/IP [6]

O KNXnet/IP apresenta várias vantagens como por exemplo:

- Permite a supervisão e controlo dos dispositivos remotamente.
- Não provoca atrasos nas comunicações, pois uma rede IP possui velocidades elevadas em comparação com a velocidade num *backbone* de uma rede KNX

3.5.1 Módulos do KNXnet/IP

O KNXnet/IP está organizado por módulos, o que permite ao projectista de produtos KNX implementar apenas o que lhe interessa, diminuindo a complexidade e consequentemente o custo. Esses módulos são os seguintes: núcleo, gestão de dispositivos, Tunnelling, Routing.

- **Núcleo:** é um módulo obrigatório, visto que é responsável pelas especificações das tramas KNXnet/IP.
- **Gestão de dispositivos:** define os serviços para efetuar configurações e gestão de servidores KNXnet/IP.

- **Tunneling:** 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. A figura 3.7 ilustra uma configuração *tunneling*.

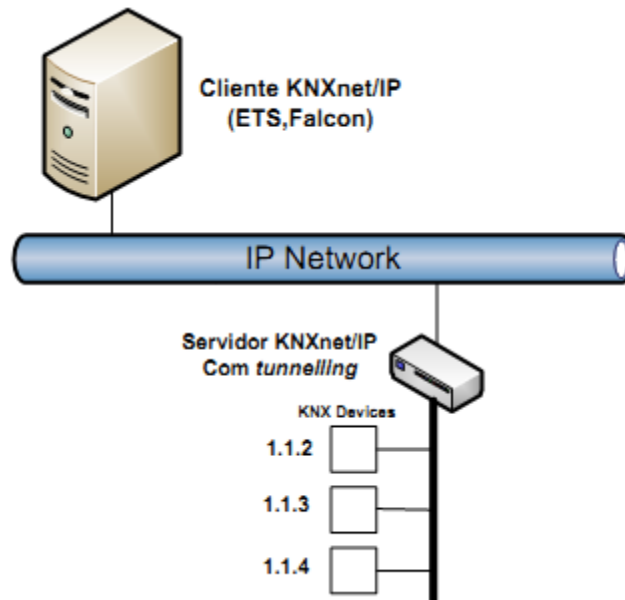


Figura 3.7: Exemplo de configuração de rede KNXnet/IP implementando tunnelling [6]

- **Routing:** é utilizado para efetuar 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 tanto do próprio servidor, como da rede KNX ligada a esse servidor. [6]

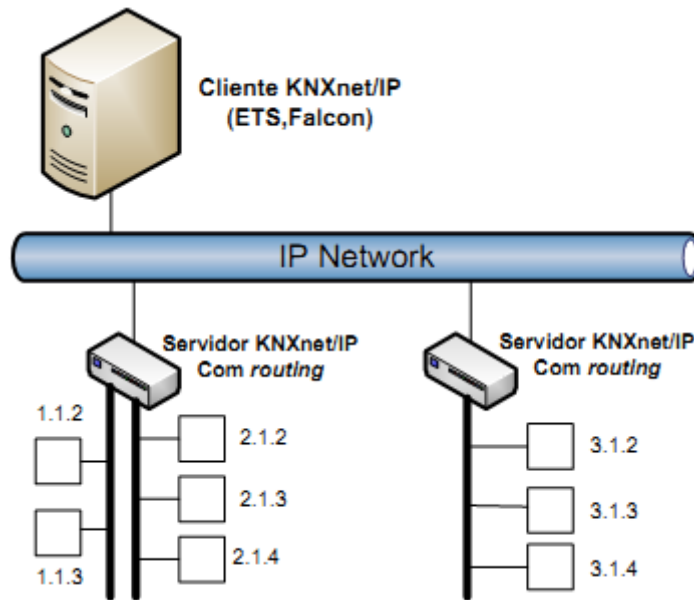


Figura 3.8: Exemplo de configuração de rede KNXnet/IP implementando routing. [6]

Com o routing, podem-se interligar várias sub-redes KNX através de uma mesma rede IP, que fica a funcionar como backbone da rede KNX, tal como aparece na figura 3.8, e o dispositivo 1.1.2 não só consegue comunicar com o dispositivo 3.1.3 como também com todos os restantes dispositivos, incluindo o cliente KNXnet/IP.

3.6 Interface gráfica

A interface gráfica para este projeto deve ser simples de utilizar, de modo que um utilizador comum, sem qualquer formação em programação, possa operar sem qualquer dificuldade.

Sendo assim, o Labview foi a plataforma escolhida. Fabricado pela National Instruments™, este software é bastante poderoso e flexível. É também dotado de um ambiente gráfico muito intuitivo, o que se enquadra perfeitamente nos objetivos definidos para este trabalho.

O Labview é uma linguagem gráfica direccionada para aplicações de aquisição de dados e controlo. Esta programação funciona segundo um modelo de fluxo de dados. Um programa desenvolvido em Labview é designado por VI ou Virtual Instrument.

O ambiente de trabalho do Labview está dividido sob duas plataformas, *Front Panel* e *Block Diagram*. A primeira permite interagir directamente com o utilizador, sendo possível monitorizar ou controlar determinado processo, enquanto que no *Block Diagram*, a segunda plataforma, é onde está o código fonte (linguagem G) que define a funcionalidade do VI.

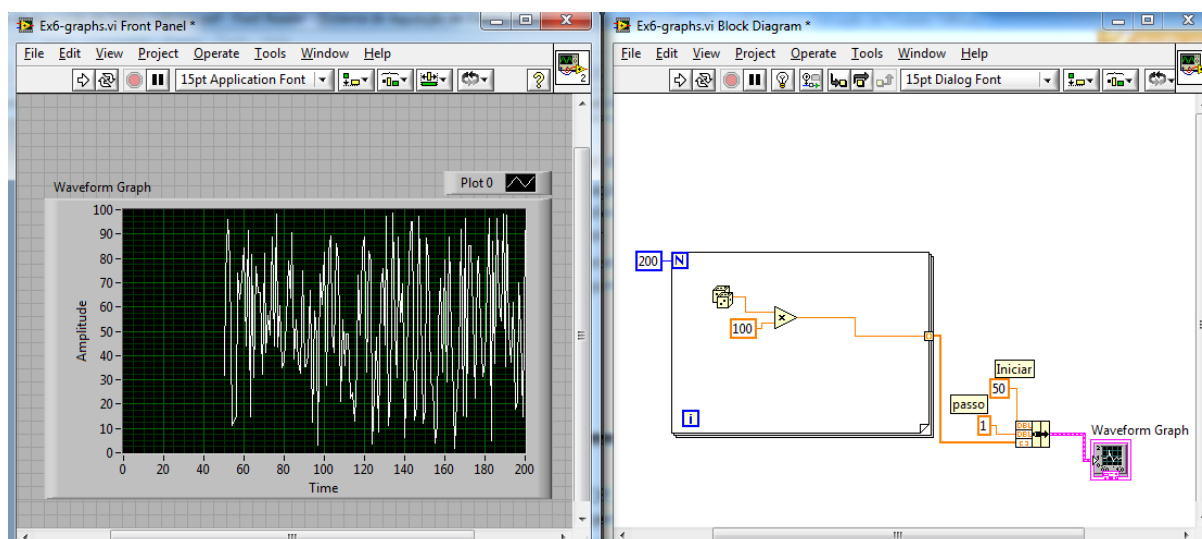


Figura 3.9: Exemplo de um VI em Labview

Na figura 3.9 está ilustrado o *Front Panel*, à direita, e o *Block Diagram*, à esquerda. O utilizador irá apenas trabalhar no *Front Panel*.

Como o protocolo TCP/IP está incluído nas bibliotecas do Labview, é possível utilizar o Modbus para comunicar com o controlador KNX-IP.

3.7 Software ETS

O ETS (Engineering Tool Software), como já foi referido anteriormente, é um software criado pela KONNEX e uma ferramenta para engenharia de sistemas EIB/KNX. Actualmente este software encontra-se na versão 4.1.6.

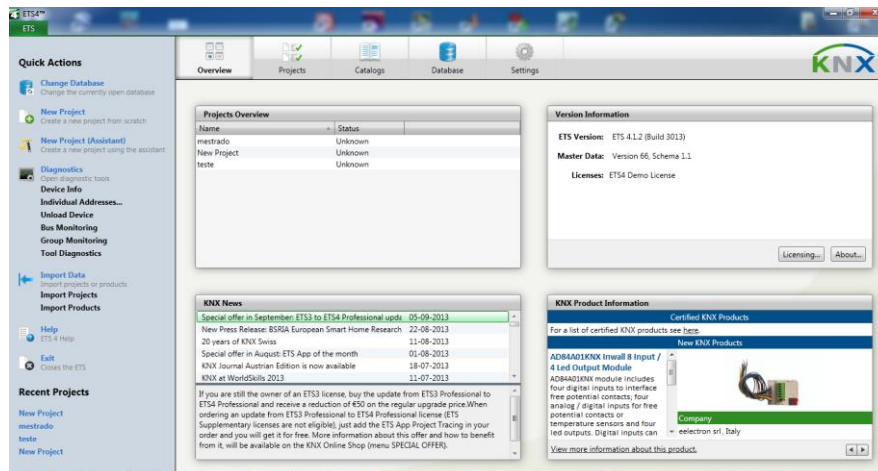


Figura 3.10: Interface gráfica do ETS 4

O ETS possui uma base de dados de todos os produtos que são fornecidos pelos fabricantes dos mesmos. A base de dados contém a informação necessária sobre cada dispositivo EIB/KNX, sendo assim possível saber quais as funções ou aplicações destes equipamentos.

Esta ferramenta apresenta várias funcionalidades, tais como desenhar e configurar um projeto EIB/KNX, ou fazer a gestão da rede EIB/KNX. Também é possível configurar toda a rede no modo offline e posteriormente carregar toda a configuração para os dispositivos.

3.8 Software CoDeSys

O CoDeSys é um software de programação para PLC e foi desenvolvido e comercializado pela empresa alemã 3S-Smart. Este software utiliza a norma industrial internacional IEC 61131-3.

Esta norma emprega várias linguagens de programação que está disponível no CoDeSys: IL (lista de instruções), ST (texto estruturado), LD (diagrama de blocos), FBD (blocos lógicos) e SFC (função gráfica sequencial).

Este programa possui várias vantagens, tais como a possibilidade de combinar múltiplas linguagens de programação, runtime, entre outras.

O controlador escolhido para este trabalho (KNX-IP Wago 750-849) é programado através do software CoDeSys.

Capítulo 4

4 Implementação Prática

Neste capítulo serão descritos os pormenores mais importantes de como foi implementado o controlo do aquecimento para o caso de estudo. Dar-se-á maior destaque à parte de domótica, ao controlador KNX-IP e à interface gráfica desenvolvida em Labview.

Neste projeto, apenas 8 quartos da residência foram considerados para efeitos de demonstração.

4.1 Domótica

A instalação de domótica vai ser estruturada e configurada através do software ETS 4. Em cada um dos quartos irá ser introduzido um controlador de temperatura (figura 4.1), que inclui sensor de temperatura, controlador PID e interface de comunicação para comunicar com os outros dispositivos KNX. Este controlador pode ser configurado num conjunto significativo de parâmetros, o que permite um elevado número de funcionalidades adaptáveis a um grande leque de aplicações. Nomeadamente é possível definir perfis de temperatura (ex. – conforto, extensão de conforto, standby, redução nocturna, protecção contra geada/calor).



Figura 4.1: Unidade KNX de controlo de temperatura para quarto.

Nesta tese foram utilizados os seguintes perfis:

- Conforto - horas de maior conforto, com temperaturas agradáveis, superiores à temperatura do perfil de redução nocturna. Este perfil é accionado no caso de o quarto estar ocupado.
- Redução nocturna - para quando o residente do quarto estiver a dormir. Proporciona uma temperatura mais baixa do que a do perfil do conforto, mas suficiente para manter a comodidade deste. Este perfil é accionado no caso de o quarto estar ocupado.
- Standby – Este perfil será apenas accionado no caso de o quarto estar desocupado ou nas horas em que não é necessário aquece-lo.

De modo a obter a temperatura desejada o controlador de temperatura comunica com o actuador de aquecimento, dando comando de posicionamento dos accionadores termoelectricos. Estes dois dispositivos estão ilustrados na figura seguinte, estando o primeiro à direita e o segunda a esquerda.



Figura 4.2: Actuador de aquecimento à esquerda e accionamento termoelectrico à direita.

O actuador de aquecimento tem 6 saídas independentes, cada uma capaz de suportar até 4 módulos de acionamento termelétrico. No caso de estudo cada quarto está equipado com um único radiador pelo que só é ligado um módulo de accionamento por saída. Assim um actuador pode ser utilizado para controlar de forma independente 6 quartos. Para além das 6 saídas independentes o actuador tem um conjunto de configurações que lhe permite adaptar-se a diferentes sistemas de aquecimento. Assim como estratégias de actuação que melhorem o rendimento da fonte de calor, bem como ações de manutenção da própria instalação.

Estes dispositivos são adicionados no software ETS 4 e é necessário coloca-los no “Group Addresses”, ou seja, estabelecer associações entre dispositivos. Para isso, é necessário atribuir endereços de rede às divisões da residência e adicionar os dispositivos necessários para cada divisão. Estes endereços estão divididos em 3 campos. O primeiro identifica o andar da residência, o segundo identifica o número do quarto e, por fim, o terceiro identifica o perfil de temperatura. Daí este tipo de endereços serem considerados endereços de sistema.

Na figura seguinte podem-se visualizar os endereços do sistema. O endereço 0-1-1 corresponde ao andar número 0, ao quarto número 1 e o último campo é responsável pelo perfil de Conforto usado para aquele quarto. O endereço 0-1-2 corresponde ao mesmo andar e quarto, mas para o perfil de Redução nocturna. O mesmo acontece para o endereço 0-2-1 e 0-2-2. Contudo, estes endereços correspondem ao quarto número 2.

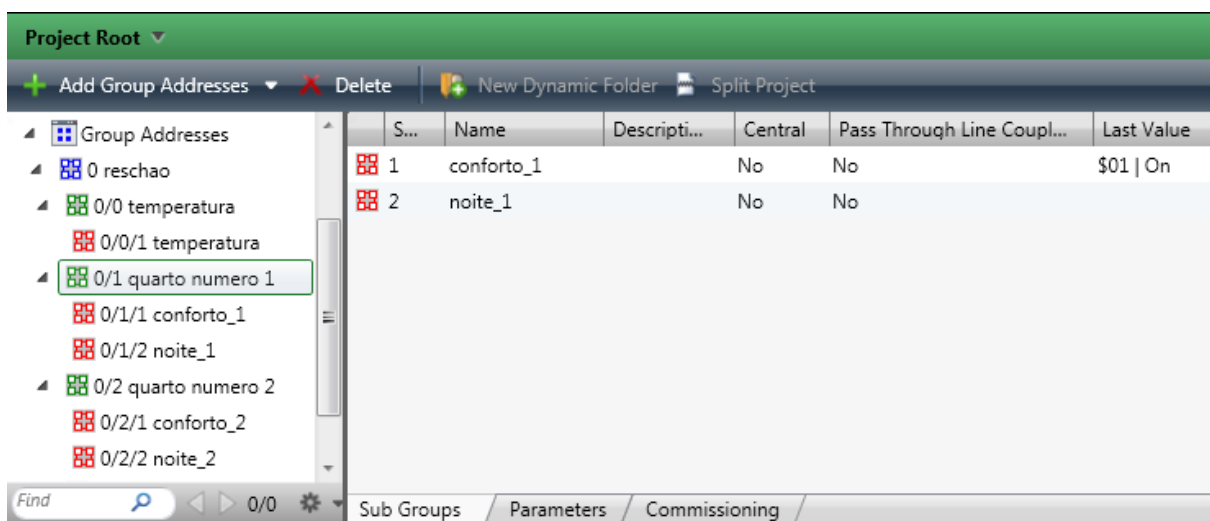


Figura 4.3: Group Addresses do sistema.

Cada um destes endereços engloba o controlador KNX-IP, a unidade KNX de controlo de temperatura para quarto e o actuador de aquecimento. Estes dois últimos dependem de uma variável do controlador KNX-IP, do tipo ON/OFF. Quando esta variável estiver com o valor lógico 1, é activado o perfil de conforto. Esta variável designa-se por “conforto_quarto” e será explicada ao pormenor mais a frente neste capítulo. Contudo, ela depende do quarto estar ou não ocupado e da hora em que os perfis devem ficar activos.

Na tabela seguinte, são apresentados os perfis de cada quarto (apenas para os quartos 1 e 2) em função do valor da variável enviada pelo controlador KNX-IP.

Tabela 4.1: Tabela o perfil de cada quarto em função da variável “conforto_quarto”

Quarto	Endereços	Variável do controlador KNX-IP (conforto_quarto)	Perfil
1	0/1/1	ON	Conforto
	0/1/2	OFF	
	0/1/1	OFF	Redução nocturna
	0/1/2	ON	
	0/1/1 e 0/1/2	OFF	Standby
2	0/2/1	ON	Conforto
	0/2/2	OFF	
	0/2/1	OFF	Redução nocturna
	0/2/2	ON	
	0/2/1 e 0/2/2	OFF	Standby

Para enviar a temperatura desejada para cada perfil, que corresponde ao endereço 0/0/1 do ETS 4, é necessário outra variável vinda do controlador KNX-IP, designada por

“Setpoint”. Esta variável difere da anterior na medida em que não é do tipo ON/OFF, possuindo o tamanho de 2 bytes.

4.2 Controlador KNX-IP

O controlador KNX-IP tem como função estabelecer a interligação entre a rede de domótica e a aplicação em Labview, através de uma *gateway*. Posto isto, o controlador KNX-IP irá transmitir os comandos enviados pelo Labview e transforma-los em comandos KNX, que posteriormente serão enviados pelo *router* KNX para a instalação de domótica.

Contudo, é necessário criar uma base de dados para que haja um controlo eficaz do sistema. Com esta base de dados o controlador do sistema de domótica fica com informação de quais os quartos que estão ocupados, a temperatura desejada e também as horas em que os radiadores irão ligar ou desligar. A base de dados está implementada na aplicação do Labview e é manuseada pelo operador. Sempre que esta seja alterada, é enviada a alteração para o controlador KNX-IP.

No CoDeSys, facilmente se consegue transformar qualquer comando vindo do Labview em comandos KNX, visto que este possui bibliotecas preparadas para este efeito.

O código a seguir apresentado, na linguagem ST do CoDeSys, mostra como duas funções podem controlar as variáveis do controlador, e enviá-las para a instalação de domótica. O código pode ser consultado na íntegra no anexo A.

```
conforto_quarto_1_4(  
    xInput_0:= switch_conf[1],  
    xInput_1:= switch_conf[2],  
    xInput_2:= switch_conf[3],  
    xInput_3:= switch_conf[4],  
    typKNX:=typKNX_IP_Controller ,  
    typBI_4:= typBI_4_dummy  
);
```

```
conforto_quarto_5_8(  
    xInput_0:= switch_conf[5],  
    xInput_1:= switch_conf[6],  
    xInput_2:= switch_conf[7],  
    xInput_3:= switch_conf[8],  
    typKNX:=typKNX_IP_Controller ,  
    typBI_4:= typBI_4_dummy  
);
```

Estas funções permitem ativar o perfil de conforto no respectivo quarto e serão adicionadas no software ETS 4 (consultar a tabela 4.1). Cada uma destas funções tem quatro saídas, do tipo ON/OFF, representando, cada uma delas, um quarto. É importante referir que estas funções estão ligadas ao sensor que detecta se a janela está aberta ou fechada, de modo a não deixar ligar a válvula de aquecimento enquanto a janela permanecer aberta.

Os dispositivos de domótica necessitam de saber quando devem actuar, ou seja, se o utilizador do sistema pretender que os radiadores estejam ligados a determinada hora, o controlador deve dar ordens para estes actuarem.

Existe uma função que desempenha este papel, comparando a hora actual (visto que o controlador possui um relógio interno) com a hora que o utilizador pretende ligar ou desligar o aquecimento. Se as horas coincidirem, o controlador envia um sinal aos dispositivos KNX para que estes actuem. Esta função é efectuada através do código a seguir transcrito:

```
sala_conf_mode(  
  bWEEK_DAY:= week_day ,  
  bInputHour:= UINT_TO_BYTE (ActualTimeEx.Hour),  
  bInputMinute:= UINT_TO_BYTE (ActualTimeEx.Minute),  
  bON_Hour:= sala_comfort_hour_on,  
  bON_Minute:= sala_comfort_minute_on,  
  bOFF_Hour:= sala_comfort_hour_off,  
  bOFF_Minute:= sala_comfort_minute_off,  
  xMonday_1:= bool_dummy,  
  xTuesday_2:= bool_dummy ,  
  xWednesday_3:= bool_dummy,  
  xThursday_4:= bool_dummy,  
  xFriday_5:= bool_dummy,  
  xSaturday_6:= bool_dummy,  
  xSunday_7:= bool_dummy,  
  xOutput=> );
```

O Labview utiliza o Modbus para comunicar com o controlador KNX-IP. As mensagens Modbus-TCP/IP devem seguir o formato apresentado na tabela 3.1, de modo que a comunicação seja efectuada com sucesso. Na tabela seguinte é exemplificado a mensagem Modbus-TCP/IP (em hexadecimal): “1234 0000 0006 FF06 0301 00F1”.

Tabela 4.2: Mensagem Modbus-TCP/IP “1234 0000 0006 FF06 0301 00F1”

Cabeçalho				Função	Dados
Identificador de Transacção	Identificador de Protocolo	Comprimento	Identificador de Unidade		
2 Bytes	2 Bytes	2 Bytes	1 Byte	1 Byte	4 Bytes
1234	0000	0006	FF	06	0301 00F1

Esta mensagem representa quais os quartos que estão ocupados na residência. Para identificar se um quarto está ocupado basta 1 bit (do tipo ON/OFF). Por exemplo, no caso de o bit estar a 1 (um), o quarto está ocupado. Na mensagem, o “F1” corresponde a 1 byte (8 bits), “11110001” em binário. Pode-se concluir que, no exemplo dado, apenas os quartos 1, 2, 3, 4 e 8 estão ocupados.

No subcapítulo 3.4.1 explicou-se que o campo “Identificador de Protocolo” deve conter o valor 0 (ou 0000 correspondendo a 2 bytes) para se utilizar o protocolo Modbus. Quanto ao campo do Comprimento, este possui o valor “0006” na mensagem, o que significa que, à frente deste campo, faltam ainda 6 (1+1+4) bytes até que a mensagem termine.

Existe um pormenor bastante importante, relativo ao campo da Função na mensagem Modbus-TCP/IP, que se deve ao facto de ele possuir o valor “06”. Como podemos verificar pela consulta da tabela 3.2, a Função com o código “06” corresponde a “escrever registo simples”, ou seja, a mensagem “1234 0000 0006 FF06 0301 00F1” irá escrever, no endereço de memória “0301”, o valor “00F1”.

É de notar que apenas o campo de Dados irá sofrer alterações, isto é, as mensagens devem iniciar sempre com “1234 0000 0006 FF06”, variando o endereço de memória e o que se pretende escrever nele. Na tabela seguinte são exemplificados os vários endereços utilizados e suas funções.

Tabela 4.3: Endereços de memória do controlador e a sua função.

Endereço (em Hexadecimal)	Função
0301	Quais os quartos ocupados
0302	Hora em que o aquecimento se irá ligar, perfil de conforto
0303	Hora em que o aquecimento se irá desligar, perfil de conforto
0304	Hora em que o aquecimento se irá ligar, perfil de redução nocturna
0305	Hora em que o aquecimento se irá desligar, perfil de redução nocturna
0306	Temperatura desejada no perfil de conforto
0307	Temperatura desejada no perfil de redução nocturna

4.3 Interface gráfica em Labview

A interface gráfica, desenvolvida em Labview, tem o objetivo de fornecer parâmetros e dados de ocupação ao controlador que permitam efetuar um controlo de supervisão da instalação de domótica. Esta é a parte do sistema que é visível para o utilizador e ele deve ser capaz, através de instruções simples de controlar toda a instalação de domótica, sendo-lhe assim ocultada toda complexidade do sistema.

A interface gráfica que será apresentada ao utilizador irá ter um aspecto semelhante ao da figura abaixo:

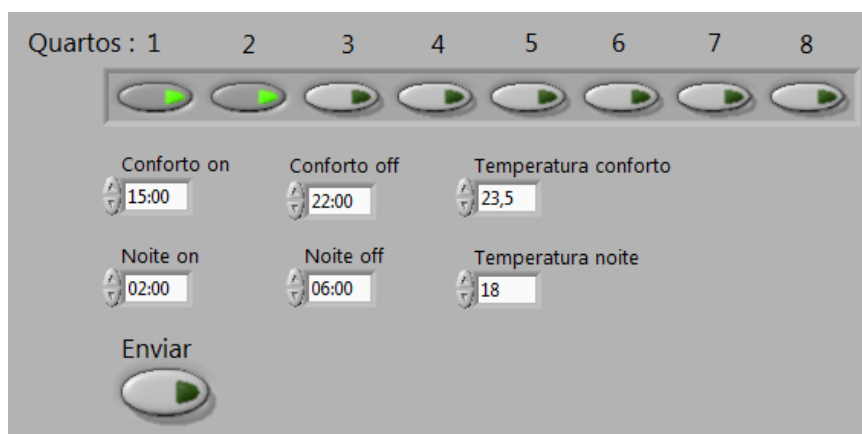


Figura 4.4: Front Panel da aplicação em Labview.

Verifica-se que, ao utilizador, são apresentados todos os campos previamente definidos, com a simplicidade que era exigida, de modo a que a aplicação seja bastante fácil de utilizar:

- Cada quarto é representado por um botão do tipo ON/OFF.
- Para as horas em que o aquecimento se irá ligar/desligar tanto para o perfil de conforto como o de redução nocturna é apresentado um indicador bastante simples de manusear.
- Quanto às temperaturas (em grau Celsius) desejadas para ambos os perfis, a aplicação não deixa que estes ultrapassem valores exagerados.
- Quando estiver tudo configurado o utilizador pode dar ordem para enviar num botão facilmente perceptível. De seguida, o sistema trabalha autonomamente.

Após estar apresentado o *Front Panel* da aplicação desenvolvida em Labview, segue-se a programação no *Block Diagram*.

O Labview possui nas suas livrarias o protocolo TCP/IP, sendo assim possível a utilização deste *software* para a realização desta tese.

O modo de utilização do protocolo Modbus no Labview é bastante simples. Basta criar uma aplicação de acordo com o modelo cliente-servidor suportada pelo protocolo TCP/IP, utilizando a porta 502, como se pode observar na figura 4.5.

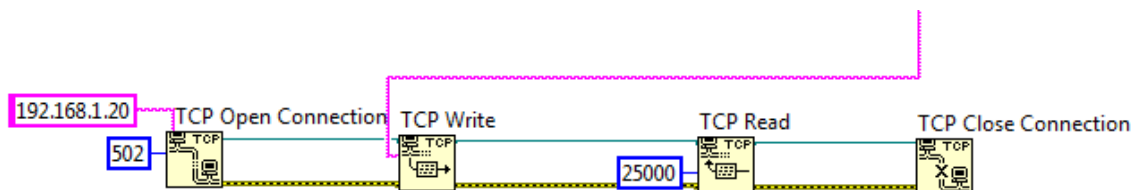


Figura 4.5: Exemplo de uma aplicação cliente-servidor, em Labview, usando Modbus.

Com esta aplicação cliente-servidor é possível comunicar com o controlador KNX-IP, visto que o IP deste é 192.168.1.20.

Um dos objetivos do programa desenvolvido em Labview é transmitir ao controlador quais os quartos que estão ocupados, a hora que devem ligar/desligar os radiadores e as temperaturas desejadas para cada quarto. Como já referido anteriormente,

deve-se enviar, segundo o protocolo Modbus, a mensagem “1234 0000 0006 FF06”, seguida dos respectivos dados, dependendo do objetivo da mensagem.

Um exemplo de como enviar ao controlador quantos quartos estão ocupados é mostrado na figura 4.6.

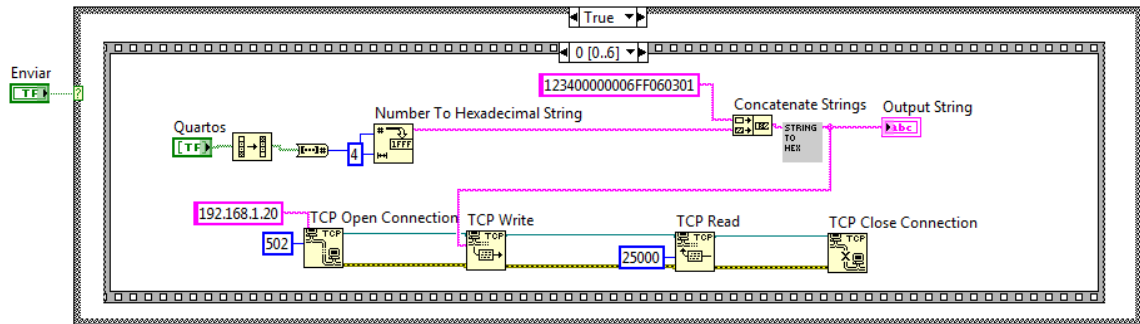


Figura 4.6: Programa, em Labview, que envia quais os quartos que estão ocupados.

A variável “Quartos” é um *array de boolean* (do tipo ON/OFF) que guarda quais os quartos que estão ocupados. À mensagem “1234 0000 0006 FF06 0301” será adicionado o valor da variável “Quartos”, em hexadecimal. Desta forma a mensagem está pronta a ser enviada.

Para enviar a hora a que devem ligar/desligar os radiadores, bem como as temperaturas desejadas, é executado algo semelhante. Na figura seguinte pode-se visualizar o mesmo programa, mas desta vez para enviar a “hora em que o aquecimento se irá ligar, perfil de conforto” e a “temperatura desejada no perfil de redução nocturna”.

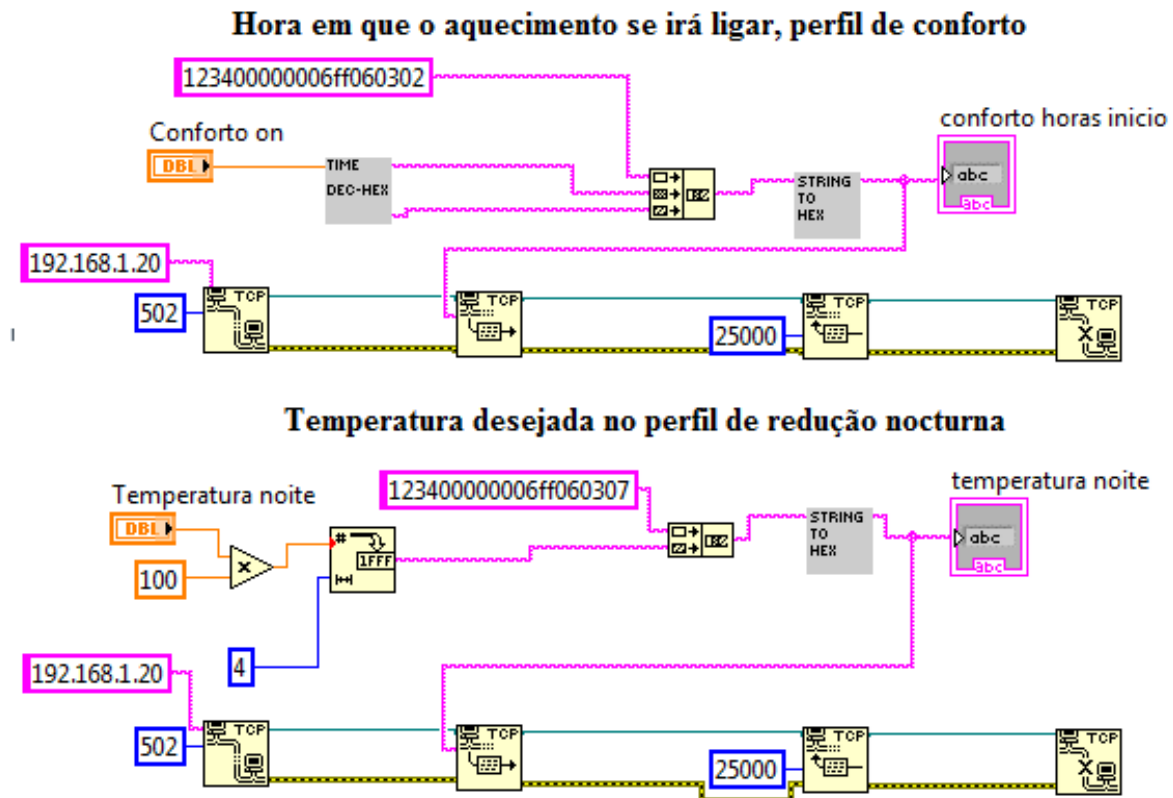


Figura 4.7: Programa, em Labview, que envia a “hora em que o aquecimento se irá ligar, perfil de conforto” e a “temperatura desejada no perfil de redução nocturna”.

Pode-se observar que apenas se altera o tratamento dos dados. A mensagem também é alterada, seguindo os endereços de memória da tabela 4X, sendo o endereço “0302” para a imagem de cima e “0307” para a de baixo.

O ETS 4 disponibiliza uma unidade de monitorização. Na figura que se segue é apresentado o ETS a efetuar a monitorização dos dispositivos KNX após terem sido dadas ordens a partir do Labview.

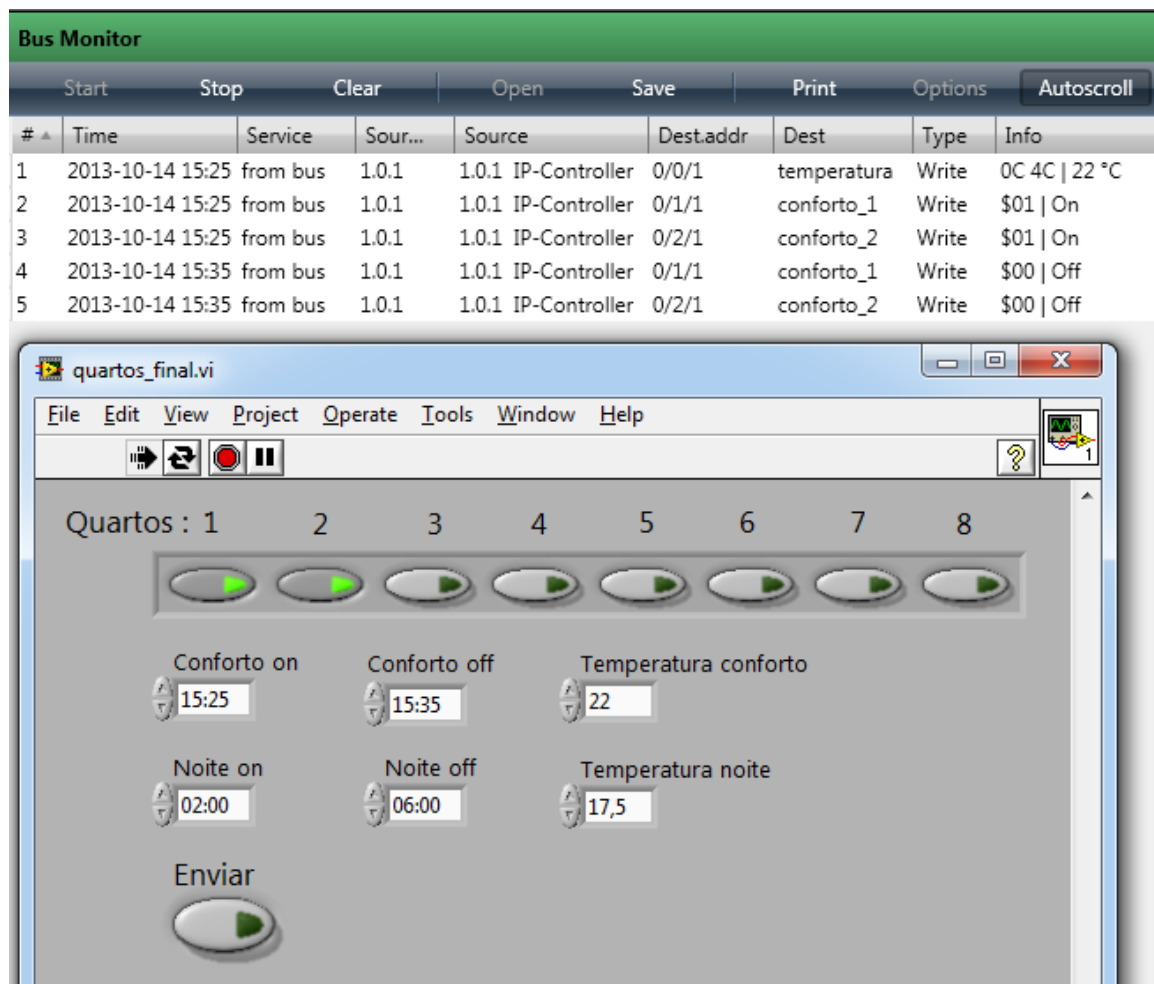


Figura 4.8: Monitorização do sistema.

Na interface gráfica, pode-se observar que os quartos 1 e 2 estão ocupados e era pretendido que o perfil de conforto fosse activado às 15:25 h e desligado às 15:35 h com uma temperatura de 22°C. Verifica-se ainda que o sistema está funcional, visto que os dispositivos KNX respondem às ordens dadas pelo utilizador.

Capítulo 5

5 Conclusão

O trabalho realizado ao longo desta dissertação resume-se ao controlo de uma rede de domótica KNX, destinada ao controlo do aquecimento dos quartos de uma residência, no Instituto Politécnico de Bragança, de uma forma eficiente e racional, sem desperdício de energia. Para efeitos de demonstração, apenas foram considerados 8 quartos da residência, uma vez que a rede de domótica não foi realmente implementada na residência.

Chegou-se à conclusão que a interface gráfica desenvolvida em Labview é funcional e bastante simples de operar. Um utilizador sem qualquer conhecimento das funções de domótica é capaz de controlar o aquecimento para cada quarto através de pedidos simples na aplicação.

O controlador KNX-IP estabeleceu de uma forma eficaz o ponto de ligação entre a aplicação desenvolvida em Labview e os dispositivos KNX, tal como foi planeado inicialmente.

O sensor que detecta se as janelas estão abertas também se encontra operacional. No caso do residente do quarto optar por abrir uma janela, o controlador KNX-IP dá automaticamente ordens aos dispositivos KNX, de modo a desligar as válvulas de aquecimento.

Referências bibliográficas

- [1] João Silva - Aplicação de Interface Com Sistema Domótico EIB
- [2] Renato Nunes - Análise Comparativa de Tecnologias para Domótica
- [3] EuroX10 - <http://www.eurox10.com/Content/x10information.htm> acessado em 10/08/13
- [4] Kevin Pinheiro de Castro -Domótica - Desenvolvimento de uma solução integradora
- [5] The EIB Association: The EIBA Handbook Series – Release 3.0, Brussels, 1998
- [6] Diana Palma – FEUP KNX Domótica KNX/EIB de Baixo Custo.
- [7] Tiago Mendes - Rede domótica KNX sobre rede física can.
- [8] ABB - Eficiência Energética em edifícios com KNX
- [9] <http://eurodomotica-knx.com.br/br/curso-knx/> acessado em 05/08/13
- [10] Adriano Ferreira - Desenvolvimento de Infra-Estrutura de Comando Multifunções EIB-KNX Para Smartphone
- [11] KNX Scientific Conference, Wien, 2006
- [12] Modbus, I.D.A -Modbus Application Protocol V1.1b
- [13] Modbus, I.D.A -Modbus messaging on tcp/ip implementation guide v1.0b

Anexo A

A Anexo A

Declaração de variáveis no *software* CoDeSys:

```
PROGRAM PROG_KNX
VAR
  teste_array :          ARRAY[1..8] OF BOOL;
  flag_janela_aberta:   ARRAY[1..8] OF BOOL;
  flag_quarto_ocupado:  ARRAY[1..8] OF BOOL;
  switch_conf:          ARRAY[1..8] OF BOOL;
  switch_noite:         ARRAY[1..8] OF BOOL;
  index_array : INT;

  MODBUS_IN_256 AT %QW256 : WORD;
  MODBUS_OUT_256 AT %IW256 : WORD;
  MODBUS_OUT_bit264_7 AT %IX264.7 : BOOL;
  MODBUS_OUT_bit256_0 AT %IX256.0 : BOOL;
  MODBUS_OUT_bit256_1 AT %IX256.1 : BOOL;
  MODBUS_OUT_bit256_2 AT %IX256.2 : BOOL;
  MODBUS_OUT_bit256_3 AT %IX256.3 : BOOL;
  MODBUS_IN_bit256_0 AT %QX256.0 : BOOL;
  MODBUS_IN_bit256_1 AT %QX256.1 : BOOL;
  MODBUS_IN_bit256_2 AT %QX256.2 : BOOL;
  MODBUS_IN_bit256_3 AT %QX256.3 : BOOL;

  (* Quartos *)
  MODBUS_IN_257 AT %QW257 : WORD;
  MODBUS_OUT_257 AT %IW257 : WORD;
  MODBUS_OUT_bit257_0 AT %IX257.0 : BOOL;
  MODBUS_OUT_bit257_1 AT %IX257.1 : BOOL;
  MODBUS_OUT_bit257_2 AT %IX257.2 : BOOL;
  MODBUS_OUT_bit257_3 AT %IX257.3 : BOOL;
  MODBUS_OUT_bit257_4 AT %IX257.4 : BOOL;
  MODBUS_OUT_bit257_5 AT %IX257.5 : BOOL;
  MODBUS_OUT_bit257_6 AT %IX257.6 : BOOL;
  MODBUS_OUT_bit257_7 AT %IX257.7 : BOOL;

  MODBUS_IN_bit257_0 AT %QX257.0 : BOOL;
  MODBUS_IN_bit257_1 AT %QX257.1 : BOOL;
  MODBUS_IN_bit257_2 AT %QX257.2 : BOOL;
```

```

MODBUS_IN_bit257_3 AT %QX257.3 : BOOL;
MODBUS_IN_bit257_4 AT %QX257.4 : BOOL;
MODBUS_IN_bit257_5 AT %QX257.5 : BOOL;
MODBUS_IN_bit257_6 AT %QX257.6 : BOOL;
MODBUS_IN_bit257_7 AT %QX257.7 : BOOL;

(*Horas conforto inicio *)
MODBUS_IN_258 AT %QW258 : WORD;
MODBUS_OUT_258 AT %IW258 : WORD;
(*Horas conforto fim *)
MODBUS_IN_259 AT %QW259 : WORD;
MODBUS_OUT_259 AT %IW259 : WORD;
(*Horas noite inicio *)
MODBUS_IN_260 AT %QW260 : WORD;
MODBUS_OUT_260 AT %IW260 : WORD;
(*Horas noite fim *)
MODBUS_IN_261 AT %QW261 : WORD;
MODBUS_OUT_261 AT %IW261 : WORD;
(*setpoint confonforto *)
MODBUS_IN_262 AT %QW262 : WORD;
MODBUS_OUT_262 AT %IW262 : WORD;

(*setpoint noite *)
MODBUS_IN_263 AT %QW263 : WORD;
MODBUS_OUT_263 AT %IW263 : WORD;

temperatura_conf: WORD;
temperatura_noite: WORD;
var_setpoint: WORD;

sala_confort_hour_on: BYTE;
sala_confort_minute_on: BYTE;
sala_confort_hour_off: BYTE;
sala_confort_minute_off: BYTE;
sala_night_hour_on: BYTE;
sala_night_minute_on: BYTE;
sala_night_hour_off: BYTE;
sala_night_minute_off: BYTE;

bool_dummy: BOOL:=TRUE;
week_day: BYTE;
KNX_IP_Master: FbKNX_Master_849; (* Instanz KNX Master /
Instance KNX Master *)
typKNX_IP_Controller: typKNX;
ProgMode: BOOL;
typDPT_Dummy: typDPT;
Status: EnumDeviceStatus;
(*setpoint_conf: FbDPT_Scaling;
setpoint_nig: FbDPT_Scaling;*)
sala_conf_mode: FbTimeSwitch;
sala_nig_mode: FbTimeSwitch;
ActualTime: SysTime64;
ActualTimeEx: SystemTimeDate;
RdCurTimeEx: CurTimeEx;

conforto_quarto_1_4: FbBinaryInput_Switch_4;
conforto_quarto_5_8: FbBinaryInput_Switch_4;
noite_quarto_1_4: FbBinaryInput_Switch_4;

```

```

        noite_quarto_5_8: FbBinaryInput_Switch_4;

        setpoint: FbDPT_Value_2_Ucount;
END_VAR

VAR RETAIN PERSISTENT
    typBI_4_dummy: typBI_4;
END_VAR

```

Código na linguagem ST do CoDeSys

```

KNX_IP_Master(typKNX:= typKNX_IP_Controller, enumDeviceStatus=> Status,
xProg_Mode=> ProgMode);

```

```

RdCurTimeEx(SystemTime:=ActualTime , TimeDate:= ActualTimeEx);

```

```

IF ActualTimeEx.DayOfWeek = 0 THEN
week_day := 7;
ELSE
week_day :=UINT_TO_BYTE( ActualTimeEx.DayOfWeek);
END_IF

```

```

sala_conf_mode(
    bWEEK_DAY:= week_day ,
    bInputHour:= UINT_TO_BYTE(ActualTimeEx.Hour),
    bInputMinute:=UINT_TO_BYTE (ActualTimeEx.Minute),
    bON_Hour:= sala_comfort_hour_on,
    bON_Minute:= sala_comfort_minute_on,
    bOFF_Hour:= sala_comfort_hour_off,
    bOFF_Minute:= sala_comfort_minute_off,
    xMonday_1:= bool_dummy,
    xTuesday_2:=bool_dummy ,
    xWednesday_3:= bool_dummy,
    xThursday_4:=bool_dummy ,
    xFriday_5:= bool_dummy,
    xSaturday_6:= bool_dummy,
    xSunday_7:= bool_dummy,
    xOutput=> );

```

```

sala_nig_mode(
    bWEEK_DAY:=week_day ,
    bInputHour:=UINT_TO_BYTE(ActualTimeEx.Hour) ,
    bInputMinute:= UINT_TO_BYTE (ActualTimeEx.Minute),
    bON_Hour:= sala_night_hour_on,
    bON_Minute:= sala_night_minute_on,
    bOFF_Hour:= sala_night_hour_off,
    bOFF_Minute:= sala_night_minute_off,
    xMonday_1:= bool_dummy,
    xTuesday_2:= bool_dummy,
    xWednesday_3:=bool_dummy ,
    xThursday_4:= bool_dummy,
    xFriday_5:=bool_dummy ,

```

```

        xSaturday_6:= bool_dummy,
        xSunday_7:= bool_dummy,
        xOutput=> );

(* flag quartos *)

flag_quarto_ocupado[1]:=MODBUS_OUT_bit257_7;
flag_quarto_ocupado[2]:=MODBUS_OUT_bit257_6;
flag_quarto_ocupado[3]:=MODBUS_OUT_bit257_5;
flag_quarto_ocupado[4]:=MODBUS_OUT_bit257_4;
flag_quarto_ocupado[5]:=MODBUS_OUT_bit257_3;
flag_quarto_ocupado[6]:=MODBUS_OUT_bit257_2;
flag_quarto_ocupado[7]:=MODBUS_OUT_bit257_1;
flag_quarto_ocupado[8]:=MODBUS_OUT_bit257_0;

(*Horas conforto inicio *)

sala_confort_hour_on:=WORD_TO_BYTE(SHR(MODBUS_OUT_258,8));
sala_confort_minute_on:=WORD_TO_BYTE(MODBUS_OUT_258);

(*Horas conforto fim *)

sala_confort_hour_off:=WORD_TO_BYTE(SHR(MODBUS_OUT_259,8));
sala_confort_minute_off:=WORD_TO_BYTE(MODBUS_OUT_259);

(*Horas noite inicio *)

sala_night_hour_on:=WORD_TO_BYTE(SHR(MODBUS_OUT_260,8));
sala_night_minute_on:=WORD_TO_BYTE(MODBUS_OUT_260);

(*Horas noite fim *)

sala_night_hour_off:=WORD_TO_BYTE(SHR(MODBUS_OUT_261,8));
sala_night_minute_off:=WORD_TO_BYTE(MODBUS_OUT_261);

(*Horas Temperatura conf/noite *)

temperatura_conf:=MODBUS_OUT_262;
temperatura_noite:=MODBUS_OUT_263;

(*Teste sensor janela*)
flag_janela_aberta[1] :=input_bit1;
flag_janela_aberta[2] :=input_bit2;
flag_janela_aberta[3]:= input_bit3;
flag_janela_aberta[4]:= input_bit4;

Output_bit1:=flag_janela_aberta[1];
Output_bit2:=flag_janela_aberta[2];
Output_bit3:=flag_janela_aberta[3];
Output_bit4:=flag_janela_aberta[4];

conforto_quarto_1_4(
    xInput_0:= switch_conf[1],
    xInput_1:= switch_conf[2],
    xInput_2:= switch_conf[3],
    xInput_3:= switch_conf[4],
    typKNX:=typKNX_IP_Controller ,
    typBI_4:= typBI_4_dummy
);

```

```

conforto_quarto_5_8(
    xInput_0:= switch_conf[5],
    xInput_1:= switch_conf[6],
    xInput_2:= switch_conf[7],
    xInput_3:= switch_conf[8],
    typKNX:=typKNX_IP_Controller ,
    typBI_4:= typBI_4_dummy
);

noite_quarto_1_4(
    xInput_0:= switch_noite[1],
    xInput_1:= switch_noite[2],
    xInput_2:= switch_noite[3],
    xInput_3:= switch_noite[4],
    typKNX:=typKNX_IP_Controller ,
    typBI_4:= typBI_4_dummy
);

noite_quarto_5_8(
    xInput_0:= switch_noite[5],
    xInput_1:= switch_noite[6],
    xInput_2:= switch_noite[7],
    xInput_3:= switch_noite[8],
    typKNX:=typKNX_IP_Controller ,
    typBI_4:= typBI_4_dummy
);

IF sala_conf_mode.xOutput = TRUE THEN
    var_setpoint:=temperatura_conf;
END_IF

IF sala_nig_mode.xOutput = TRUE THEN
    var_setpoint:=temperatura_noite;
END_IF

setpoint(
    wValue_IN:= var_setpoint,
    xUpdate_KNX:= ,
    bSendOnDelta:= ,
    tMinSendTime:= ,
    typDPT:= typDPT_Dummy,
    typKNX:= typKNX_IP_Controller,
    wValue_OUT=> ,
    xUpdate_PLC=> ,
    xTimeOut=> );

FOR index_array := 1 TO 8 DO

switch_conf[index_array]:=((sala_conf_mode.xOutput AND
flag_quarto_ocupado[index_array]) AND NOT
flag_janela_aberta[index_array]);
switch_noite[index_array]:=((sala_nig_mode.xOutput AND
flag_quarto_ocupado[index_array]) AND NOT
flag_janela_aberta[index_array]);
END_FOR

```