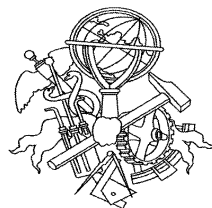


SISTEMA SEM FIOS PARA CONTROLO DE ILUMINAÇÃO DECORATIVA

Manuel Duarte Matos Lourenço



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Controlo

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

2011

Este relatório satisfaz, parcialmente, os requisitos que constam da Ficha da Unidade Curricular de Tese/Dissertação, do 2º ano, do Mestrado em Engenharia Electrotécnica e de Computadores

Candidato: Manuel Duarte Matos Lourenço, N° 1050381, 1050381@isep.ipp.pt

Orientação científica: Custódio João Pais Dias, cpd@isep.ipp.pt

Empresa:

Supervisão: Paulo Jorge Duarte Mendes, paulomendes@castros.com.pt



Mestrado em Engenharia Electrotécnica e de Computadores

Área de Especialização de Automação e Controlo

Departamento de Engenharia Electrotécnica

Instituto Superior de Engenharia do Porto

6 de Agosto de 2011

Dedico este trabalho a todas as pessoas que, de alguma forma, contribuíram para a realização deste Projecto de Final de Curso e me ajudaram a ultrapassar as dificuldades encontradas.

Resumo

Este projecto surgiu no contexto da solicitação de um dos clientes da empresa Castros Iluminações que pretendia um sistema de iluminação decorativa (baseada em LED) de dois conjuntos de 288 janelas, pertencendo cada conjunto a uma fachada diferente do edifício. Este sistema teria que prever a possibilidade de controlar a cor de cada janela individualmente, dando ao cliente a possibilidade de alterar o ambiente decorativo das fachadas do edifício.

A utilização de comunicação sem fios foi justificada pela necessidade de evitar a quantidade enorme de cabos que seria necessário passar utilizando os sistemas comerciais convencionais e a impossibilidade de os esconder. Esta solução foi pensada para ser implementada num edifício com 14 Andares, facto que por si só inviabiliza a passagem de cabos.

Para interligar todos os dispositivos de iluminação decorativa com um controlador portátil, e eliminar os problemas ligados à cablagem, foi utilizado o protocolo de comunicações sem fios ZigBee™. A escolha recaiu neste protocolo devido a factores como os seus baixos consumos, simplicidade do protocolo comparativamente com outras redes e o seu baixo custo. No desenvolvimento deste projecto foi utilizada a *stack* da Microchip®, versão gratuita, disponibilizada na internet e os seus *transceivers* de comunicação Zigbee™, MRF24j40MA.

Para fazer a interface de comunicação com o utilizador, foi desenvolvida uma aplicação de Software para correr em computadores com ambiente Windows™. Esta aplicação foi desenvolvida em Visual Studio™ utilizando a linguagem C#.

Foram efectuados alguns testes para se perceber a eficiência e robustez da comunicação ZigBee™ e apesar do número de dispositivos disponíveis para ensaios ser muito reduzido, foi possível verificar que, mesmo funcionando correctamente, o desempenho do sistema poderá ser melhorado, quer seja a nível da gestão das comunicações, quer a nível do *software* informático para controlo do ambiente decorativo das fachadas do edifício.

O sistema, no actual estado de desenvolvimento, permite controlar a cor dos vários dispositivos da rede através do computador, com uma resolução de 24bits. A aplicação desenvolvida em Visual Studio™ permite controlar de forma simples e intuitiva para o utilizador, a cor do material iluminativo dos vários dispositivos da rede.

Palavras-Chave

Controlo da Iluminação, ZigBee™, LED, RGB, USB, Comunicação sem fios, C#, Visual Studio™ 2008, PIC18LF4620, PIC18LF2550, Zena™, DMX, RS485, RGB, *Castros Iluminações*. ml-200486812

Abstract

This project arose in the context of a request from the company's customers Castros Illuminations which wanted a decorative lighting system (based on LED) of two sets of 288 windows, each set belonging to a different facade of the building. This system would have to provide that control the colour of each window individually, giving the customer the ability to change the environment of the decorative facades of the building.

The use of wireless communication was justified by the need to avoid the huge amount of cables that would need to spend using the conventional commercial systems and the inability to hide them. This solution was designed to be implemented in a building with 14 floors, which in itself undermines the darning cables.

To interconnect all the decorative lighting controller with a laptop and eliminate the problems beset the wiring, we used the protocol wireless ZigBee™. The choice fell on this protocol due to factors such as its low consumption, simplicity of the protocol compared with other systems and their low cost. In this project we used the stack from Microchip®, free version available on the Internet and its communication transceivers Zigbee™, MRF24J40MA.

To make the communication interface with the user, we developed an application software to run on Windows™ computers. This application was developed in Visual Studio™ using C #.

Some tests were made to realize the efficiency and robustness of communication ZigBee™ and despite the number of devices available for testing to be very low, we observed that even functioning properly, system performance can be improved, whether the management communications, both within the computer software for environmental monitoring of the decorative facades of the building.

The system, in the current state of development, allows you to control the colour of various network devices via a computer with a resolution of 24bits. The application developed in Visual Studio™ allows you to control a simple and intuitive to the user, the colour of the illuminating equipment of various network devices.

Keywords

Lighting Control, ZigBee™, LED, RGB, USB, Wireless Communication, C#, *Visual Studio*™ 2008, PIC18LF4620, PIC18LF2550, Zena™, DMX, RS485, RGB, Castros Iluminações.

Résumé

Ce projet est né dans le contexte d'une demande de clients de la société Castros Illuminations qui voulaient un système d'éclairage décoratif (à base de LED) de deux séries de 288 fenêtres, chaque ensemble appartenant à une façade différente de l'édifice. Ce système devra prévoir que le contrôle de la couleur de chaque fenêtre individuellement, offrant au client la possibilité de changer l'environnement des façades décoratives du bâtiment.

L'utilisation de la communication sans fil a été justifiée par la nécessité d'éviter l'énorme quantité de câbles qui auraient besoin de passer en utilisant les systèmes commerciaux conventionnels et l'incapacité de les cacher. Cette solution a été conçue pour être mise en œuvre dans un immeuble de 14 étages, ce qui en soi est une atteinte à reprendre les câbles.

Pour interconnecter tous les contrôleurs d'éclairage décoratif avec un ordinateur portable et d'éliminer les problèmes qui assaillent le câblage, nous avons utilisé le protocole sans fil ZigBee™. Le choix s'est porté sur ce protocole en raison de facteurs tels que sa faible consommation, la simplicité du protocole par rapport à d'autres systèmes et leur faible coût. Dans ce projet nous avons utilisé la pile de Microchip®, version gratuite disponible sur l'Internet et ses émetteurs-récepteurs de communication Zigbee™, MRF24J40MA.

Pour rendre l'interface de communication avec l'utilisateur, nous avons développé un logiciel d'application pour fonctionner sur des ordinateurs Windows™. Cette application a été développée en utilisant Visual Studio™ C #.

Certains tests ont été faits pour réaliser l'efficacité et la robustesse de communication ZigBee™ et malgré le nombre d'appareils disponibles pour des essais très bas, nous avons observé que même le bon fonctionnement, la performance du système peut être améliorée, si la gestion des communications, tant au sein du logiciel pour la surveillance de l'environnement des façades décoratives du bâtiment.

Le système, à l'état actuel de développement, nous permet de contrôler la couleur des différents périphériques du réseau avec un ordinateur ayant une résolution de 24 bits. Le

logiciel développée en Visual Studio™ nous permet de contrôler de manière simple et intuitive pour l'utilisateur, la couleur de l'équipement d'éclairage des différents périphériques réseau.

Mots-clés

Contrôle de l'Illumination, ZigBee™, LED, RGB, USB, Communication sans fil, C#, Visual Studio™ 2008, PIC18LF4620, PIC18LF2550, Zena™, DMX, RS485, RGB, Castros Iluminações.

Índice

RESUMO	III
ABSTRACT	V
RESUME	VII
ÍNDICE	IX
ÍNDICE DE FIGURAS	XI
ÍNDICE DE TABELAS	XIII
ACRÓNIMOS	XV
1. INTRODUÇÃO	1
1.1. CONTEXTUALIZAÇÃO	1
1.2. OBJECTIVOS.....	2
1.3. CALENDARIZAÇÃO	3
1.4. ORGANIZAÇÃO DO RELATÓRIO	4
2. ILUMINAÇÃO ARQUITECTURAL	7
2.1. SOLUÇÕES NO MERCADO PARA CONTROLO DE ILUMINAÇÃO DECORATIVA	7
2.2. UTILIZAÇÃO DO RGB PARA REPRESENTAÇÃO DE CORES	10
3. REDES SEM FIOS	13
3.1. TECNOLOGIAS NO MERCADO.....	13
3.2. COMPARAÇÃO ENTRE DIFERENTES TECNOLOGIAS SEM FIOS	14
3.3. TECNOLOGIA SEM FIOS A UTILIZAR NESTE PROJECTO - FUNDAMENTAÇÃO	15
4. O PROTOCOLO ZIGBEE™ –UM ESTUDO DETALHADO	19
4.1. PEQUENA PERSPECTIVA HISTÓRICA SOBRE O ZIGBEE™	19
4.2. ESPECIFICAÇÕES TÉCNICAS.....	20
4.3. A ARQUITECTURA PROTOCOLAR ZIGBEE™ / IEEE 802.15.4	21
4.4. DIFERENCIAÇÃO DE DISPOSITIVOS NA REDE.....	23
4.5. TOPOLOGIAS DE REDE.....	24
4.6. MODOS DE OPERAÇÃO DE UMA REDE ZIGBEE™	26
4.7. TERMINOLOGIA ZIGBEE™	27
4.8. ESTRUTURA DAS TRAMAS DO ZIGBEE™.....	29
5. DESENVOLVIMENTO DO PROJECTO	31
5.1. ESTRUTURA DE REDE ADOPTADA PARA O PROJECTO.....	31
5.2. DESCRIÇÃO DO HARDWARE.....	32
5.3. DESCRIÇÃO DO <i>FIRMWARE</i>	40

5.4.	DESCRIÇÃO DO SOFTWARE DA APLICAÇÃO.....	50
5.5.	ANÁLISE DOS RESULTADOS.....	54
6.	CONCLUSÕES	57
	REFERÊNCIAS DOCUMENTAIS.....	59
	ANEXO A. CONFIGURAÇÕES DO PIC18F2550 PARA COMUNICAÇÃO USB.....	61
	ANEXO B. ESQUEMÁTICOS DAS PCBS.....	63
	ANEXO C. LAYOUT DAS PLACAS DE CIRCUITO IMPRESSO.....	65

Índice de Figuras

Figura 1	Controlador para fitas de LED RGB - PrimeRGB Controller™[9]	8
Figura 2	Aplicação de Software da PIMETEC®[9]	9
Figura 3	Software <i>Sunlite Suite</i> ™ da NICOLAUDIE®[10]	9
FIGURA 4	Modelo de cores RGB, representação numérica de cores.	11
Figura 5	Arquitectura protocolar ZigBee™ [2]	21
Figura 6	Configuração de rede: Estrela (<i>Star</i>)	25
Figura 7	Configuração de rede: Árvore (<i>Cluster Tree</i>)	25
Figura 8	Configuração de rede: Malha (<i>Mesh</i>)	26
Figura 9	Arquitectura do <i>profile</i> do ZigBee™ [2]	28
Figura 10	Estrutura das tramas do protocolo ZigBee™	29
Figura 11	Configuração de rede adoptada para o projecto: Malha (<i>Mesh</i>)	32
Figura 12	a) Transceiver de Comunicação ZigBee™. b) Antena MRF24J40MA.	33
Figura 13	Interface de comunicação com o computador	35
Figura 14	Software de programação de PIC <i>microC</i> ™ da MikroElektronica®	36
Figura 15	HID Terminal – Ferramenta do <i>microC</i> ™ para apoio ao desenvolvimento para USB.	37
Figura 16	Interface de controlo das fitas de LED.	38
Figura 17	Forma de onda típica de um sistema PWM	39
Figura 18	Tramas de dados Trocadas entre o Software RGBCastros e o dispositivo ZigBee™.	44
Figura 19	Rotinas da PEN ZigBee™	47
Figura 20	Rotinas de um nó da rede – Controlo RGB,	49
Figura 21	RGBCastros, Aplicação de software para edição de cenários cromáticos.	51
Figura 22	Janela de criação de um novo projecto	52
Figura 23	Edição da cor da janela presente na Fachada 1 que está na Linha 1 e Coluna 1.	52
Figura 24	Constituição da trama de dados que é enviada para o dispositivo USB	53
Figura 25	Diagrama de configurações da fonte de <i>clock</i> utilizada na comunicação USB	62

Índice de Tabelas

Tabela 1	Calendarização do projecto	4
Tabela 2	Principais características do Wi-Fi™, Bluetooth™ e ZigBee™	15
Tabela 3	Especificações tecnológicas da comunicação ZigBee™	20
Tabela 4	Especificação IEEE 802.15.4: dispositivos lógicos e suas funções [2]	23
Tabela 5	Redes ZigBee™: dispositivos lógicos e suas funções [2]	23
Tabela 1	Ajuste dos bits de configuração do PIC18LF2550	61

Acrónimos

ACK	–	Acknowledgment
API	–	Application Programming Interface
APL	–	Application Layer
APO	–	Application Objects
APS	–	Application Support Sub Layer
ASCII	–	American Standard Code for Information Interchange
CSMA-CA	–	Carrier Sense Multiple Access with Collision Avoidance
DRC	–	Dimmer Remote Control
EEPROM	–	Electrically-Erasable Programmable Read-Only Memory
EUSART	–	Enhanced Universal Asynchronous Receiver Transmitter
FCS	–	Frame Check Sequence
FFD	–	Full Function Device
GTS	–	Guaranteed Time Slot
IEEE	–	Institute of Electrical and Electronics Engineers
MAC	–	The Medium Access Control Layer
OSI	–	Open Systems Interconnection
O-QPSK	–	Offset Quadrature Phase Shift Keying
PAN	–	Personal Area network

PHY	–	Physical
PWM	–	Pulse Width Modulation
RF	–	Radiofrequência
RFD	–	Reduced Function Device
SPI	–	Serial Peripheral Interface
WLAN	–	Wireless Local Area etwork
WMAN	–	Wireless Metropolitan Area etwork
WPAN	–	Wireless Personal Area etwork
WWAN	–	Wireless Wide Area etwork
ZCP	–	ZigBee Compliant Platform
ZDO	–	ZigBee Device Object

1. INTRODUÇÃO

Este documento, desenvolvido no contexto da unidade curricular Tese/Dissertação, do 2º ano do Mestrado em Engenharia Electrotécnica e Computadores (MEEC), do Departamento de Engenharia Electrotécnica (DEE), regista todos os desenvolvimentos e resultados obtidos, não só relativos ao trabalho realizado, mas também para futuros projectos a desenvolver neste campo. Nesta secção é apresentada uma contextualização do projecto que se pretende desenvolver, os objectivos do projecto, bem como uma apresentação da estrutura deste documento.

1.1. CONTEXTUALIZAÇÃO

Nos últimos anos tem-se assistido a uma enorme expansão de dispositivos de comunicação via rádio, que surgem nas mais diversas áreas, ocupando um espaço cada vez mais significativo no quotidiano de pessoas e empresas. A ausência de uma ligação por cabo e uma conseqüente versatilidade em termos de instalação ou posição dos dispositivos intervenientes na rede é a vantagem mais evidente das redes que se apoiam sobre este tipo de comunicação, a somar à possibilidade dos próprios emissores ou receptores poderem estar em movimento durante a comunicação.

A Castros Iluminações é uma empresa que oferece aos seus clientes as mais variadas soluções de iluminação decorativa para aplicação em edifícios, locais públicos e iluminações de Natal. Estas soluções têm que ser frequentemente personalizadas aos gostos do cliente e ajustadas às necessidades físicas dos locais onde são instaladas. Há frequentemente a necessidade de exercer acções de controlo sobre estes sistemas e a sua

instalação implica por vezes a utilização de grandes quantidades de cablagem que, por razões de estética ou protecção, tem que ser escondida. Isto pode tornar-se uma necessidade dispendiosa e incomportável quando se fala de instalações cuja longevidade é de uma época festiva, por exemplo.

Desta forma, revelar-se-ia de interesse a possibilidade de comunicação com estes dispositivos de iluminação sem recurso a cablagem de controlo, para que o impacto da introdução de um novo sistema seja diminuído, nomeadamente em termos de instalação. Não sendo desprezável a questão da mobilidade, torna-se relevante que a comunicação seja efectuada preferencialmente via rádio.

Este projecto surgiu no contexto da solicitação de um dos clientes da empresa Castros Iluminações que pretendia um sistema de iluminação decorativa (baseada em LED) de dois conjuntos de 288 janelas, pertencendo cada conjunto a uma fachada diferente do edifício. Este sistema teria que prever a possibilidade de controlar a cor de cada janela individualmente, dando ao cliente a possibilidade de alterar o ambiente decorativo das fachadas do edifício. Pretende-se que o sistema inclua também o software que permita ao utilizador programar os vários ambientes decorativos de forma simples e intuitiva.

É também essencial que o sistema a desenvolver apresente boa imunidade relativamente a interferências electromagnéticas, pois no local o ambiente estará consideravelmente poluído por outros sistemas de comunicação (com e sem fios).

1.2. OBJECTIVOS

O objectivo principal deste projecto é o desenvolvimento de um sistema sem fios para controlo de iluminação arquitectural e a sua prossecução foi focalizada nos seguintes pontos de desenvolvimento:

- Estudo das soluções de mercado mais utilizadas na iluminação arquitectural;
- Estudo das tecnologias de comunicação *Wireless* existentes no mercado e selecção da mais adequada a ser aplicada no projecto em desenvolvimento;
- O estudo da tecnologia de iluminação LED;

- Desenvolvimento da estrutura da rede de comunicação para controlo dos sistemas de LED;
- O desenvolvimento de um Software, com uma interface amigável para o utilizador, que permita controlar as características da iluminação;
- O aperfeiçoamento e simplificação de processos de desenvolvimento.

Com este projecto, pretende-se conseguir uma sólida abordagem ao problema, desde a identificação de características desejáveis ao levantamento das tecnologias e hardware disponíveis no mercado, na busca de soluções adequadas e implementáveis.

1.3. CALENDARIZAÇÃO

Este projecto foi desenvolvido ao longo de dois anos, sempre com a dificuldade de conciliar a actividade profissional desenvolvida na empresa Castros Iluminações, com o trabalho de investigação e desenvolvimento que este projecto implicou, o que foi fazendo com que as várias metas se prolongassem um pouco mais no tempo do que aquilo que seria desejável. Apesar disso, foi possível criar uma tabela de objectivos, Tabela 1, com a planificação dos respectivos tempos.

Esta tabela inclui um conjunto de tarefas, como por exemplo: o estudo das tecnologias e documentação de fabricantes; desenvolvimento de *Hardware* e *Software* para materialização do projecto; teste e validação das soluções, entre outras.

Tabela 1 Calendarização do projecto

Cronograma de Projecto																						
Ano Lectivo	2009/2010											2010/2011										
Tarefas	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Set	Out	Nov	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul
Planificação e estruturação dos objectivos	█	█																				
Estado da arte - Iluminação Arquitectural			█	█	█	█																
Estudo de Tecnologias sem fios			█	█	█	█	█															
Estudo pormenorizado do protocolo ZigBee							█	█	█	█	█	█	█									
Estudo de Stack ZigBee da Microchip - Desenvolvimento do Projecto										█	█	█	█	█	█							
Estudo da ferramenta de desenv. de aplicações em C# - Visual Studio Estudo															█							
Estudo da linguagem de programação "C#" - Desenvolvimento do Projecto															█	█	█					
Desenvolvimento de Hardware										█	█	█	█	█	█	█	█					
Controlo RGB Software e Hardware															█	█	█	█	█			
Fase de testes e optimização do sistema																	█	█	█	█	█	█
Elaboração do Relatório final		█	█	█	█	█									█	█	█	█	█	█	█	█

1.4. ORGANIZAÇÃO DO RELATÓRIO

Neste Capítulo foi feita uma pequena introdução a contextualizar o trabalho e apresentar os objectivos que o norteiam. É também neste capítulo que é apresentada a estrutura e calendarização de tarefas, Tabela 1. No capítulo seguinte, são apresentadas as principais redes sem fios existentes no mercado e destacam-se as diferenças entre elas. Pretende-se nesta fase do trabalho identificar qual a tecnologia que mais se adequa aos requisitos do projecto em desenvolvimento e justificar a opção pela tecnologia ZigBee™. No capítulo 4 faz-se uma abordagem mais aprofundada do protocolo ZigBee™ e topologias de rede que este permite, procurando identificar qual a mais adequada, no contexto da aplicação em desenvolvimento.

No capítulo 5 são apresentados todos os desenvolvimentos práticos feitos neste projecto e a forma como foram feitos, explicando-se as várias fases que o compõem: Comunicação sem fios, aplicação de software para interface com o utilizador, controlo de potência das fitas de LED RGB, comunicação USB, etc..

Por fim, no capítulo 6 são apresentadas as conclusões que foram sendo tiradas ao longo do projecto e as perspectivas de futuros desenvolvimentos.

2. ILUMINAÇÃO ARQUITECTURAL

Neste capítulo são apresentados, de uma forma resumida, os resultados de uma pesquisa do mercado relativamente a potenciais tecnologias que poderiam ser adquiridas para o desenvolvimento deste projecto e as técnicas que normalmente são utilizadas neste género de aplicações. Pretende-se com isto apresentar, de forma resumida, o estudo de mercado que foi feito, para tentar perceber o actual estado de desenvolvimento existente nesta área da iluminação decorativa e poder assim criar condições para desenvolver algo com alguma inovação e potencial acrescido relativamente ao que já existe, no contexto deste projecto.

2.1. SOLUÇÕES NO MERCADO PARA CONTROLO DE ILUMINAÇÃO DECORATIVA

Quando se fala em controlo de iluminação, na vertente do espectáculo, as soluções que são utilizadas, por excelência, têm na sua base o protocolo DMX. Este é também o protocolo de comunicação mais utilizado em iluminação arquitectural, sendo inúmeros os fabricantes com soluções baseadas neste protocolo de comunicação. Este protocolo utiliza o mesmo hardware/meio físico que o protocolo RS485.

São vários os fabricantes na área da iluminação arquitectural que possuem soluções interessantes, listando-se em seguida alguns exemplos:

O *PrimeRGB Controller*TM é um dispositivo electrónico da *PRIMETEC*[®] que permite a criação de programas de iluminação utilizando barras e fitas de LED.

Potencialidades:

- Pode ser utilizado com qualquer barra ou fita de iluminação LED que funcione a 12V DC;
- Através de um botão pode-se alterar o programa de memória activo;
- Cada programa pode conter uma cor ou uma sequência de cores com temporizações;
- Permite criar transições de cor suaves com alta resolução.



Figura 1 Controlador para fitas de LED RGB - PrimeRGB ControllerTM[9]

Em termos de programação possui uma aplicação de software muito simples para o utilizador, e que serviu de inspiração para a aplicação desenvolvida neste projecto (ver Figura 2).

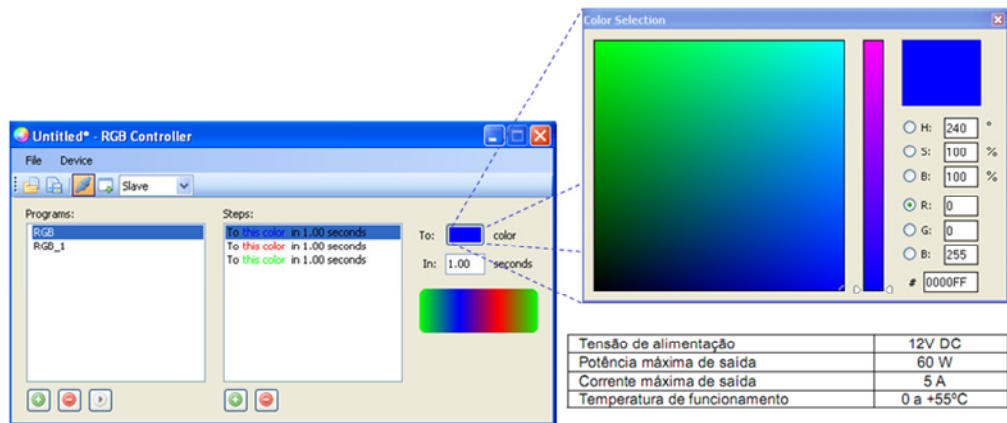


Figura 2 Aplicação de Software da PIMETEC®[9]

Uma ferramenta de Software mais completa, que utiliza o protocolo DMX na sua base de comunicação com os vários dispositivos de iluminação é o *Sunlite Suite*™, ver Figura 3.

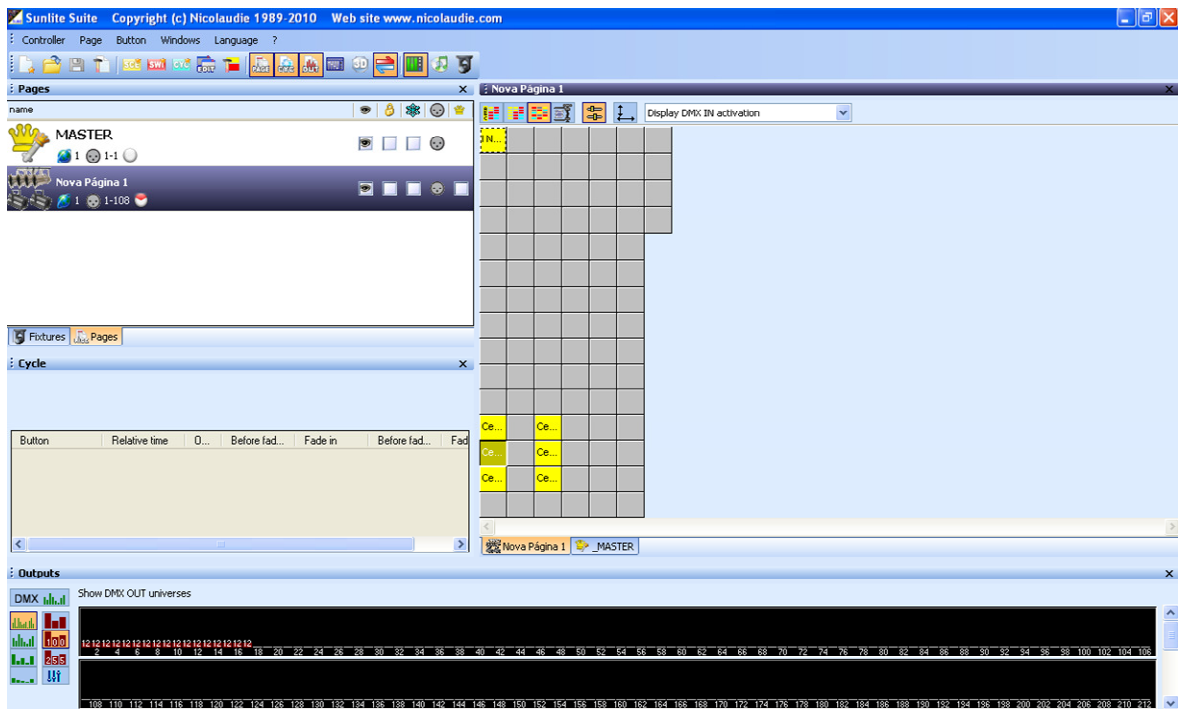


Figura 3 Software *Sunlite Suite*™ da NICOLAUDIE®[10]

Esta ferramenta de *Software* permite não só criar espectáculos de iluminação arquitectural, como também é largamente utilizado para criação de espectáculos de palco, como controlador dos projectores de luz, máquinas de fumo, *Gobos* com efeitos de luz, etc.. Qualquer dispositivo que utilize comunicação DMX pode ser controlado utilizando esta aplicação de *Software*. Possui também ferramentas mais avançadas para criação de espectáculos de luz sincronizada com som.

2.2. UTILIZAÇÃO DO RGB PARA REPRESENTAÇÃO DE CORES

O modelo de cores RGB é um modelo aditivo no qual as cores: vermelho, verde e azul claro são somados de várias formas para formar uma ampla gama de cores. O nome do modelo vem das iniciais das três cores primárias em inglês: *red*, *green*, *blue*. Este modelo, bem como muitos outros, permite fazer uma descrição inequívoca de qualquer cor, em termos da intensidade das cores primárias que a formam (vermelho, verde e azul).

Foi criado para a detecção, representação, e exibição de imagens em sistemas electrónicos, como televisores e computadores, embora tenha sido também utilizado em fotografia, sistemas de iluminação e projecção.

Tecnicamente, na iluminação RGB, é feito um agrupamento de três fontes luminosas, em que cada uma produz uma das três cores primárias. Estas fontes permitem o controlo da intensidade individualmente (e os três feixes de luz são somados), Figura 4c.

Para que a identificação de uma determinada cor seja inequívoca, é atribuído um valor numérico a cada uma das cores primárias que a constituem. A percentagem de cor presente é tão maior quanto maior for o valor numérico que a quantifica, dentro da escala considerada. A escolha da escala está relacionada com a resolução de cores que se pretende obter. É comum utilizar-se um byte (8 bits) de resolução para cada cor primária, o que permite variações de percentagem de cada cor entre 0 e 255. Por outras palavras, para cada cor primária, é possível obter 255 níveis diferentes de intensidade.

Sendo que cada cor que se pretende obter é composta pelas três cores primárias, e considerando 8 bits de resolução para cada uma, é possível obter uma resolução de 24 bits, o que equivale a mais de 16,7 milhões de cores diferentes. O conjunto de todas estas cores pode ser representado com recurso a um cubo, em que cada cor é um ponto da superfície ou dentro dele, Figura 4a.

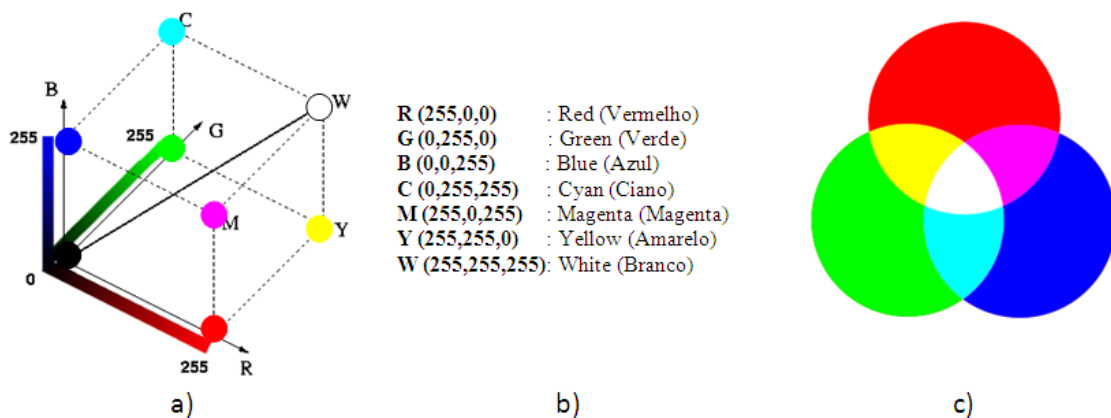


Figura 4 Modelo de cores RGB, representação numérica de cores.

Os tons de cinza são localizados na diagonal entre o branco e o preto e a ausência de cor é identificada como sendo o preto, é obtida quando os três componentes são 0 (0,0,0), isto é, quando as três fontes luminosas estão desligadas. Se as cores primárias forem combinadas duas a duas com a intensidade máxima (255), obtêm-se as três cores secundárias: amarelo (255,255,0), ciano (255,0,255) e magenta (0,255,255), Figura 4b.

Actualmente são muito utilizadas *Strings* de LED RGB, com a grande vantagem de permitirem obter uma grande gama de cores utilizando um único material iluminativo. Existem também projectores de LED RGB com lentes especiais para aumentar o seu rendimento.

3. REDES SEM FIOS

Este capítulo foca essencialmente o interesse e estado actual de sistemas de comunicação sem fios. É feita uma referência a algumas das tecnologias que se pensa poder ser aplicáveis no desenvolvimento deste projecto, procurando-se destacar as principais características e aspectos comparativos de cada uma. Pretende-se com isso evidenciar a tecnologia mais adequada a ser aplicada no desenvolvimento deste projecto.

3.1. TECNOLOGIAS NO MERCADO

O aumento do campo de actuação deste tipo de tecnologia nas mais diversas áreas permitiu o aparecimento de várias alternativas e protocolos no mercado. Inicialmente foi atribuída muita ênfase à transmissão de dados e voz com elevadas taxas de transferência, perspectiva que tornava os equipamentos caros e não suscitava qualquer interesse em aplicações mais simples.

Foi neste contexto que foram desenvolvidas tecnologias, não só com débito mais baixo, mas principalmente com um alcance muito mais pequeno, conseguindo assim um menor custo, como o Wi-Fi™ e o Bluetooth™.

No entanto, para determinadas situações, estas tecnologias eram demasiado complexas de implementar, além de terem um consumo de energia relativamente elevado.

Então, como solução para redes com menores débitos de dados (na ordem dos *kbits*), e um consumo muito baixo, para dar resposta às necessidades de muitos dos sensores e dispositivos de controlo, em que o reduzido consumo, tal como o tamanho compacto do módulo são essenciais, a associação *ZigBee™ Alliance* iniciou o desenvolvimento do protocolo *ZigBee™* [1].

Todas as questões relacionadas com redes sem fios encontram-se nas recomendações do *Institute of Electrical and Electronics Engineers* (IEEE), nomeadamente sobre o padrão 802.11. Este reconhece a existência de quatro grandes grupos, no que toca às redes sem fios:

- ✓ *Wireless Personal Area Network* (WPAN) – Engloba redes de pequeno alcance, entre 10 m a 100 m (IEEE 802.15) e são utilizadas em dispositivos como os PDA ou telemóveis, sendo o *Bluetooth™* (IEEE 802.15.1) a tecnologia mais utilizada. O *Low-Rate Wireless Personal Area Network* (LR-WPAN) é parte integrante deste grupo e procura reunir baixo débito com baixa complexidade e consumo. É sobre este padrão (IEEE 802.15.4) que foi desenvolvido o *ZigBee™*;
- ✓ *Wireless Local Area Network* (WLAN) - Agrupa as tecnologias com alcance entre os 100 m e os 300 m, normalmente como extensão de redes com fios convencionais;
- ✓ *Wireless Metropolitan Area Network* (WMAN) – Engloba os acessos por banda larga em redes metropolitanas, com alcance em torno dos 6 km;
- ✓ *Wireless Wide Area Network* (WWAN) – É o grupo mais amplo em termos de alcance, orientado para serviços de telecomunicações de longa distância.

3.2. COMPARAÇÃO ENTRE DIFERENTES TECNOLOGIAS SEM FIOS

As tecnologias que foram objecto deste estudo comparativo pertencem ao grupo das *Wireless Personal Area Network* (WPAN) e das *Wireless Local Area Network* (WLAN), por serem as que mais se adequam aos requisitos deste projecto. São essencialmente três: o *Wi-Fi™*, *Bluetooth™* e o *ZigBee™*.

Tabela 2 Principais características do Wi-Fi™, Bluetooth™ e ZigBee™

Especificação	Débito	Alcance	Pilha Protocolar	Vantagens	Principais aplicações
Wi-Fi (IEEE 802.11x)	11 Mbps, 54 Mbps	450m	>1M	Elevada taxa de transferência;	Internet, Transferências de Ficheiros, Vídeo/Aúdio;
Bluetooth (IEEE 802.15.1)	1 Mbps	10-100m	250KB	Interoperabilidade Substituição de cabos	Periféricos de PC e Telemóveis, PDA's;
ZigBee (IEEE 802.15.4)	20 kbits, 40 kbits, 250 kbits	10-150m	32K	Baixo consumo, latência N° de nós, fiabilidade, preço;	Controlo Remoto, Sensores, Dispositivos alimentados por bateria;

O Wi-Fi™, baseado nas normas da família IEEE 802.11, é um protocolo orientado para aplicações que requeiram um elevado débito, tais como a transferência de ficheiros ou o acesso à Internet e a conteúdos multimédia. Por outro lado, dados os objectivos deste protocolo, o suporte à sua pilha protocolar, em termos de hardware, é bastante mais exigente (podendo assumir a mesma um tamanho considerável). O seu consumo energético, especialmente quando em *standby*, é também claramente superior ao das outras tecnologias que aqui se comparam, Tabela 2.

Por ambos apresentarem débitos de dados mais reduzidos em relação ao Wi-Fi™, uma dúvida frequentemente surgida é se o Bluetooth™ e ZigBee™ são efectivamente protocolos “concorrentes”.

Segundo a própria ZigBee™ Alliance, o ZigBee™ foi desenvolvido para servir aplicações bastante distintas das do Bluetooth™ e, em especial, atentando a uma optimização máxima do consumo energético. O protocolo ZigBee™ apresenta um consumo muito reduzido quando comparado com o Wi-Fi™ ou Bluetooth™, especialmente nos períodos de inactividade – o seu consumo será, nesta fase, cerca de 100 vezes inferior ao do Bluetooth™.

3.3. TECNOLOGIA SEM FIOS A UTILIZAR NESTE PROJECTO - FUNDAMENTAÇÃO

Tendo em consideração o estudo feito, e apresentado no tópico anterior, foi necessário ponderar qual destas tecnologias seria a mais adequada para os requisitos do projecto.

A tecnologia de comunicação sem fios utilizando o protocolo ZigBee™, tendo em conta os requisitos deste projecto, oferece vantagens face a outras tecnologias sem fios.

São de destacar as seguintes vantagens:

- A baixa complexidade da pilha protocolar, comparativamente aos restantes protocolos, exige menores recursos nos dispositivos que a utilizam, menor formação dos técnicos de instalação e assim sendo, menores custos.
- Possibilita um baixo consumo de energia, importantíssimo para dispositivos remotos alimentados a bateria.
- O débito e alcance, ainda que reduzidos, são suficientes para a monitorização e controlo de equipamentos e sensores.
- Possui boas características de imunidade ao ruído, possibilitando a coexistência no mesmo espaço de outras tecnologias com e sem fios;
- Teoricamente permite criar redes com até 65535 dispositivos. Para este projecto, a rede será constituída por 577 dispositivos (duas fachadas com 288 janelas cada, mais a *pen* de ligação ao PC);
- Foi pensado para aplicações de Domótica, nas quais a fiabilidade e robustez são factores essenciais. Sendo que este projecto é para aplicação num edifício, as exigências técnicas são semelhantes (desempenho face à existência de paredes, electrodomésticos, equipamentos informáticos e outros sistemas de comunicação);
- Em evoluções futuras, existe a possibilidade de integração deste sistema com outros sistemas Domóticos que utilizem ZigBee™;
- A simplicidade em termos de *Hardware* permite o desenvolvimento de um transceiver mais compacto, o ideal para desenvolvimento de uma PEN USB para interface do computador com a rede;
- Permite criar redes complexas e com alcances geográficos grandes, devido à possibilidade de configurar dispositivos como Routers e como terminais de rede.

Foi com base nestes argumentos que se optou por esta tecnologia de comunicação sem fios para o desenvolvimento deste projecto.

4. O PROTOCOLO ZIGBEE™ –UM ESTUDO DETALHADO

Depois de feita uma abordagem geral às diferentes tecnologias de comunicação sem fios, no capítulo anterior, neste capítulo é feito um estudo mais detalhado deste protocolo e das suas características operacionais.

4.1. PEQUENA PERSPECTIVA HISTÓRICA SOBRE O ZIGBEE™

Anteriormente à adopção da nomenclatura ZigBee™ e da constituição da ZigBee™ Alliance, a responsável por esta tecnologia era a empresa Philips®, sendo nessa época designado por protocolo Home RF Lite.

Apesar de actualmente admitir distintas topologias, foi através da topologia em malha que se lhe associou o nome. Uma malha ZigBee™ apresenta múltiplos percursos possíveis entre cada dispositivo, o que permite eliminar um possível ponto de falha, através do “zig” e “zag” da informação pela rede. As abelhas de mel, para além de viverem “em colmeia”, voam em ziguezague e dessa forma é-lhes possível informar os restantes elementos do enxame da distância, direcção e localização dos alimentos descobertos. Assim, o nome deste protocolo surge da analogia da estrutura e modo de funcionamento da rede de comunicações para com o modo de vida das abelhas.

4.2. ESPECIFICAÇÕES TÉCNICAS

O protocolo ZigBee™ usa a especificação IEEE 802.15.4 na definição da camada física (PHY) e de controlo de acesso ao meio (MAC). Esta especificação define três gamas de frequências, isentas de licenciamento, em que pode operar, conhecidas como *Industrial Scientific and Medical* (ISM) [2]:

- 2.4 GHz no mundo inteiro;
- 915 MHz nos Estados Unidos da América;
- 868 MHz na Europa.

Cada uma destas gamas oferece um número fixo de canais de operação, ver Tabela 3, e a taxa de transmissão do protocolo depende da frequência com a qual se opera. A 2.4 GHz obtêm-se 250 kbps, a 915 MHz obtêm-se 40 kbps e a 868 MHz obtêm-se 20 kbps. No entanto, as taxas obtidas em qualquer das gamas de frequências serão menores do que as especificadas, devido ao *overhead* dos pacotes e atrasos de propagação [2].

Em termos de modulação, para a banda de 2,4 GHz é utilizado o *Offset Quadrature Phase-Shift Keying* (O-QPSK) e para as bandas de 915 MHz e 868 MHz é utilizado o *Binary Phase-Shift Keying* (BPSK), Tabela 3.

Tabela 3 Especificações tecnológicas da comunicação ZigBee™

Frequências de operação	Espectro de frequências	Número de Canais	Taxa de transmissão	Tipo de modulação
868 MHz	868-868.6 MHz	1	20 Kbps	DSSS com BPSK
915 MHz	902-923 MHz	10	40 Kbps	DSSS com BPSK
2.4 GHz	2.4-2.4835 MHz	16	250 Kbps	DSSS com O-QPSk

4.3. A ARQUITECTURA PROTOCOLAR ZIGBEE™ / IEEE 802.15.4

A arquitectura do protocolo ZigBee™ é composta por camadas, sendo que cada uma executa serviços específicos ao dispor da camada superior, ou seja, a entidade de dados fornece dados para o serviço de transmissão e a entidade de gestão fornece informação para todos os outros serviços. Cada entidade expõe uma interface para a camada superior através do ponto de acesso ao serviço (SAP) e cada SAP suporta um número de primitivas que activam a funcionalidade que se pretende solicitar, Figura 5.

Embora se baseie no Modelo *Open Systems Interconnection* (OSI), de sete camadas, a pilha protocolar ZigBee™ apenas define as camadas de interesse que permitem atingir as funcionalidades desejadas [2].

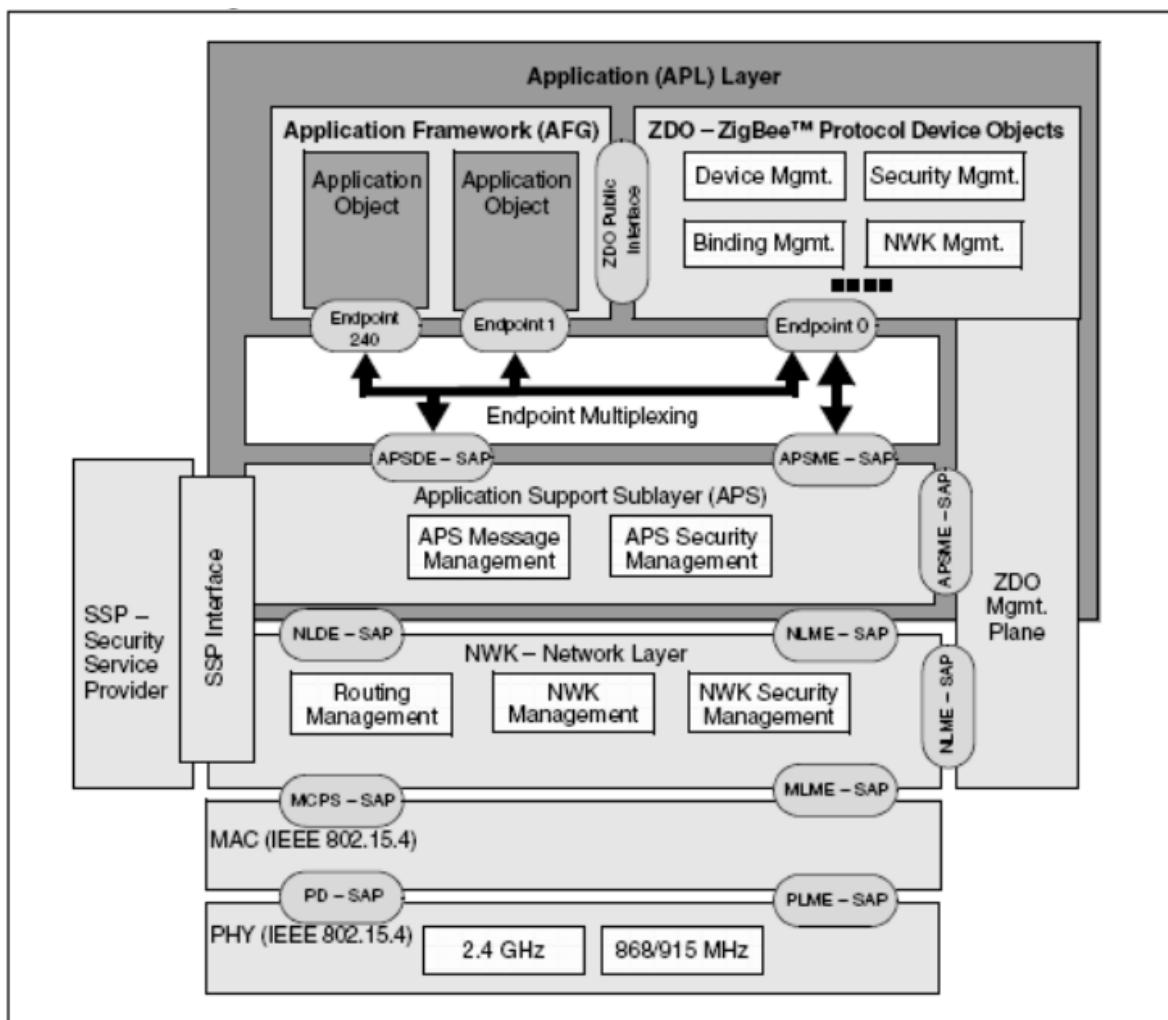


Figura 5 Arquitectura protocolar ZigBee™ [2]

As duas primeiras camadas, a camada física (PHY) e a camada de controlo de acesso ao meio (MAC), são definidas pelo padrão IEEE 802.15.4. No nível imediatamente a seguir encontram-se as camadas definidas pela ZigBee™ Alliance: a camada de rede (NWK) e o interface de aplicação (API). Aqui se incluem a subcamada de suporte à aplicação (APS), o objecto de dispositivo ZigBee™ (ZDO) e a *framework* de aplicação (AF).

A camada PHY é responsável por permitir a transmissão e recepção de mensagens através de um canal físico, o *transceiver* RF, sua activação e desactivação, detecção de energia (ED), indicação da qualidade da ligação (LQI) e selecção do canal.

À camada MAC cabe, essencialmente, o papel de controlar os acessos ao *transceiver* RF, utilizando o protocolo de acesso ao meio *Carrier Sense Multiple Access with Collision Avoidance* (CSMA-CA), para o qual efectua comunicações com a camada PHY.

A trama MAC do IEEE 802.15.4 tem um tamanho máximo de 127 bytes, incluindo um campo CRC (Controlo de Erro) de 16 bits, sendo este o que verifica a integridade da trama. Adicionalmente, o IEEE 802.15.4 usa um mecanismo de confirmação (ACK) de tramas opcional, o que permite que todas as tramas com uma *flag* especial de ACK sejam automaticamente confirmadas pelo receptor. Este método garante a entrega da informação. Quando uma trama é transmitida com esta *flag* de ACK e a confirmação não é recebida durante um determinado período de tempo, o emissor irá retransmitir um número fixo de vezes até declarar erro [2].

É importante notar que o facto de uma trama ser confirmada correctamente na recepção somente indica que essa mesma trama foi recebida pela camada MAC, não implicando contudo que esta tenha sido devidamente processada. É possível que o receptor da camada MAC receba e confirme uma trama mas, devido à falta de recursos de processamento, essa trama seja abandonada pelas camadas superiores. Para evitar essa situação, essas camadas podem requerer mecanismos de confirmação adicionais [1]. Especifica também o tipo de dispositivos permitidos pela rede e a estrutura de tramas admissível.

A camada NWK tem como sua responsabilidade o início e fim de ligações de dispositivos à rede, a descoberta de novos dispositivos na vizinhança e o armazenamento de informação relativa aos mesmos, a atribuição de endereços (apenas em dispositivos Coordenadores), a descoberta de rotas, o encaminhamento de informação e a configuração de novos dispositivos.

Quanto à camada APL, esta pretende assegurar uma correcta gestão e suporte para as diversas aplicações da camada de aplicação (AP). A camada AP, sendo a última da pilha protocolar, é definida também pelo IEEE 802.15.4, mas principalmente pelo utilizador que vai utilizar esta tecnologia [2]. Contém a *Application Framework*, para suporte ao desenvolvimento de aplicações, *ZigBee Device Objects* (ZDO) e *Application Support Sub Layer* (APS).

4.4. DIFERENCIAÇÃO DE DISPOSITIVOS NA REDE

A especificação IEEE 802.15.4 distingue dois tipos de dispositivos físicos, Tabela 4, os quais desempenham diferentes funções na rede [2].

Tabela 4 Especificação IEEE 802.15.4: dispositivos lógicos e suas funções [2]

Tipos de Dispositivos	Serviços Oferecidos	Fonte de Alimentação Típica	Configuração do Receptor Típica
Full Function Device (FFD)	Maioria	Ligado à Corrente	Ligado quando em Idle
Reduced Function Device (RFD)	Limitados	Bateria	Desligado quando em Idle

Sobre estes, o protocolo ZigBee™ considera ainda três tipos de dispositivos, que se relacionam com os utilizados pelo IEEE 802.15.4 da forma apresentada na Figura 9

Tabela 5 Redes ZigBee™: dispositivos lógicos e suas funções [2]

Dispositivo ZigBee	Tipo de dispositivo físico associado (IEEE 802.15.4)	Função Típica
Coordenador	FFD	Forma a rede, aloca endereços de rede e guarda a tabela de <i>binding</i> .
Router	FFD	Permite que mais nós se juntem à rede, ao aumentar o seu alcance físico. Pode também efectuar funções de controlo ou monitorização. A sua existência é opcional.
Terminal (Endpoint)	FFD ou RFD	Efectua acção de controlo ou monitorização através do dispositivo que lhe esteja associado (sensor, controlador, actuador...)

O coordenador é uma variante especial do *Full Function Device* (FFD) que implementa um maior número de serviços. O dispositivo terminal pode ser FFD ou *Reduced Function*

Device (RFD). Este último é o mais pequeno e simples dispositivo do ZigBee e implementa apenas um número mínimo de serviços. O terceiro componente, o *router*, é opcional e está presente em algumas configurações de rede. A necessidade de utilização de dispositivos configurados como *Routers* ocorre quando vários dispositivos da rede estão fora do alcance um do outro.

4.5. TOPOLOGIAS DE REDE

O protocolo de comunicação sem fios ZigBee™ permite a criação de vários tipos de configurações de rede, existindo em todas pelo menos dois componentes principais: o coordenador e o dispositivo terminal.

Em seguida, apresenta-se uma representação gráfica e uma abordagem sucinta do conceito inerente a cada configuração de rede, podendo cada uma delas coexistir dentro da mesma rede: [2]

4.5.1. CONFIGURAÇÃO DE REDE EM ESTRELA (*STAR*)

Uma rede em estrela é constituída por um coordenador e um ou mais dispositivos terminais, como exemplificado na Figura 6. Nesta rede os dispositivos terminais apenas podem comunicar directamente com o coordenador. Se um dispositivo terminal necessitar de comunicar com outro, tem de transferir os dados para o coordenador e este, posteriormente, envia-os para o dispositivo terminal em questão [2].

É ao ZigBee™ *Coordinator* que cabe todo o controlo da rede, assumindo um papel central e de comunicação directa com todos os dispositivos *Endpoints* – é portanto o Coordenador que inicia e mantém os dispositivos na rede.

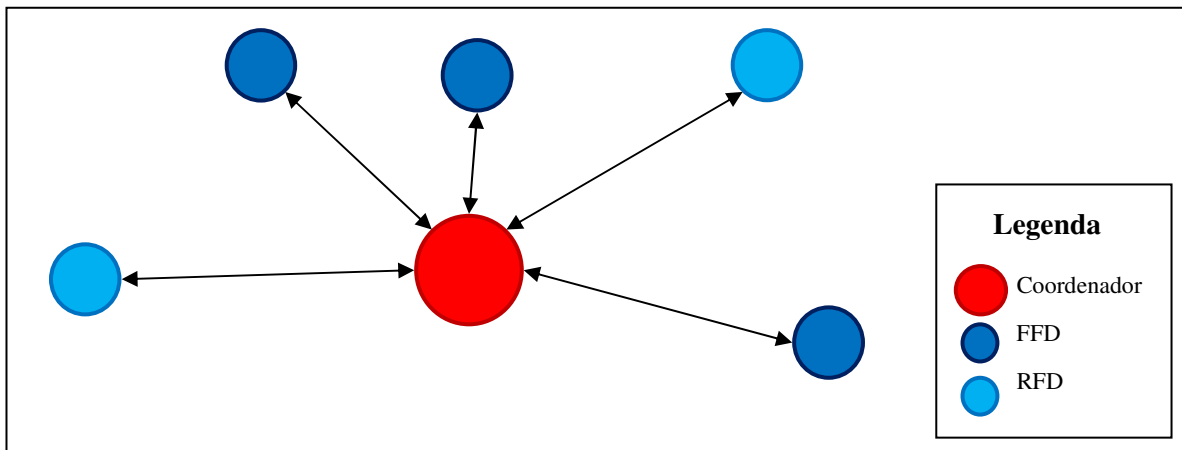


Figura 6 Configuração de rede: Estrela (*Star*)

4.5.2. CONFIGURAÇÃO DE REDE EM ÁRVORE (*CLUSTER TREE*)

Outra configuração de rede é a topologia em árvore, exemplificada na Figura 7. Nesta configuração, dispositivos terminais podem juntar-se tanto a coordenadores como a routers. Os *routers* servem dois propósitos: um é aumentar o número de nós na rede, outro é estender o alcance físico. Com a adição de um *router*, um dispositivo terminal não necessita de estar ao alcance do coordenador. Todas as mensagens neste tipo de rede são encaminhadas por toda a árvore [2].

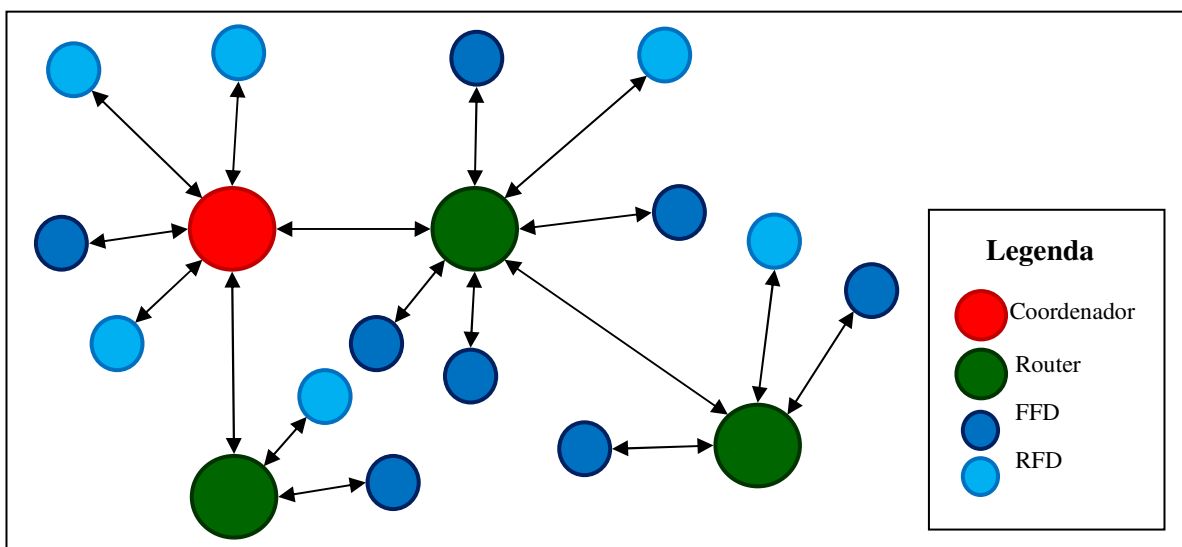


Figura 7 Configuração de rede: Árvore (*Cluster Tree*)

4.5.3. CONFIGURAÇÃO DE REDE EM MALHA (*MESH*)

Uma rede em malha, como mostra a Figura 8, é similar a uma rede em árvore, com a excepção de que os dispositivos FFD podem encaminhar mensagens directamente para outros dispositivos FFD, sem terem de seguir a estrutura da árvore. As vantagens desta topologia são a diminuição da latência das mensagens e aumento da fiabilidade na sua entrega [2].

As topologias em árvore e malha são também conhecidas como redes *multi-hop*, graças às suas capacidades de encaminhar pacotes através de múltiplos dispositivos, ao contrário da topologia em estrela que é uma rede *single-hop*.

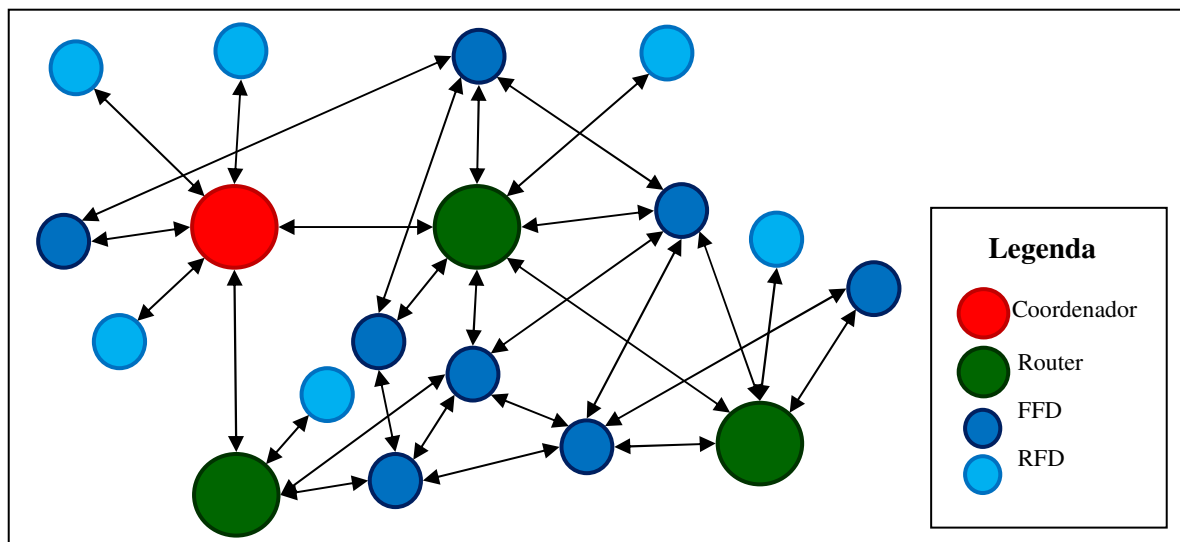


Figura 8 Configuração de rede: Malha (*Mesh*)

4.6. MODOS DE OPERAÇÃO DE UMA REDE ZIGBEE™

Uma rede ZigBee™ é uma rede de acesso múltiplo, isto é, todos os nós numa rede têm igual acesso ao meio de comunicação. Existem dois tipos de mecanismos de acesso múltiplo, *beacon* e *non-beacon*, sendo esta a característica que permite ao ZigBee™, usufruir de uma das suas vantagens, a poupança de energia dos dispositivos na rede.

Numa rede cujo mecanismo activo é o *non-beacon* os seus dispositivos estão constantemente ligados, podendo deste modo haver comunicação entre eles a qualquer altura. É permitido que todos os nós entrem em transmissão a qualquer altura, desde que o

canal esteja em *Idle*. Uma vez detectada esta condição, o dispositivo pode transmitir a trama. Se o destino for um FFD, então o transmissor está sempre ligado e outros dispositivos podem transmitir a qualquer altura. No entanto, se o destino for um RFD, então este pode desligar o transmissor quando está em *Idle* de modo a conservar energia. Neste estado, o RFD não pode receber mensagens [2].

Esta condição é lidada através da obrigatoriedade de todas as mensagens de e para um RFD passarem pelo FFD pai desse mesmo RFD. Quando o RFD liga o seu transmissor, pede as mensagens ao seu pai e, se este tiver recebido alguma para o seu filho, encaminha-a. Isto permite que o RFD conserve energia, mas requer que o FFD tenha memória suficiente para encaminhar mensagens para todos os seus filhos. Se o filho não requisitar qualquer mensagem dentro de um certo período de tempo “*macTransactionPersistenceTime*”, a mensagem irá dar *timeout* e o pai irá abandoná-la.

Numa rede com o mecanismo *beacon* activado, os nós apenas podem transmitir por *slots* de tempo pré-determinados. Periodicamente, o coordenador inicia com uma super trama, identificada como uma trama *beacon* e é esperado que todos os nós da rede se sincronizem com esta trama. A cada dispositivo é atribuído um *slot* de tempo específico na super trama, durante o qual é permitida a transmissão e recepção. A super trama pode também conter um *slot* comum, no qual todos os nós podem aceder ao canal.

4.7. TERMINOLOGIA ZIGBEE™

O *profile* do protocolo ZigBee™ é simplesmente uma descrição de dispositivos lógicos e suas interfaces. Os dispositivos inseridos numa rede ZigBee™ obedecem a um determinado *Profile*. Um *Profile* descreve o ambiente da aplicação, ou seja, contém um conjunto pré definido de mensagens, procedimentos e interfaces de um determinado dispositivo, define também o formato dos atributos.

Cada pedaço de informação que possa ser passado entre dispositivos, através de comandos é denominado por atributo. A cada atributo é atribuído um identificador único, sendo estes agrupados em *clusters*. A cada *cluster* é atribuído também um identificador único. As interfaces são especificadas ao nível do *cluster* e não ao nível do atributo, apesar de os atributos serem transferidos individualmente.

O *profile* define os valores dos identificadores dos atributos e dos identificadores dos *clusters*, quais os mandatários e quais os opcionais. Adicionalmente, o *profile* também pode definir alguns serviços do ZigBee™ como mandatários. O utilizador pode utilizar estas definições e criar o seu próprio código para as utilizar. Pode escrever o código como quiser, agrupando as funções como quiser, desde que use os *clusters* e atributos tal e qual como foram definidos no *profile*. Desta forma, dispositivos de diferentes fabricantes conseguem trabalhar em conjunto. Cada bloco de código que suporte um ou mais *clusters* é denominado por *endpoint*.

Os dispositivos são assim definidos pelos seus *Profiles* e implementados na rede como *Application Objects* (APO). A cada APO é atribuído um único número *endpoint* que o liga à restante *stack* ZigBee™. Um *endpoint* suporta um ou mais *clusters* e é-lhe atribuído um endereço (entre 1 a 240) que é adicionado ao endereço da rede de 16 bits e ao endereço MAC de 64 bits. Deste modo, as APO podem comunicar através dos seus *endpoints*, utilizando *clusters*.

Dispositivos diferentes comunicam via os seus *endpoints* e os *clusters* que suportam. A Figura 9 mostra um exemplo de uso de um *Profile*, diferenciando as várias camadas que o compõe e mostrando como elas interagem entre si.

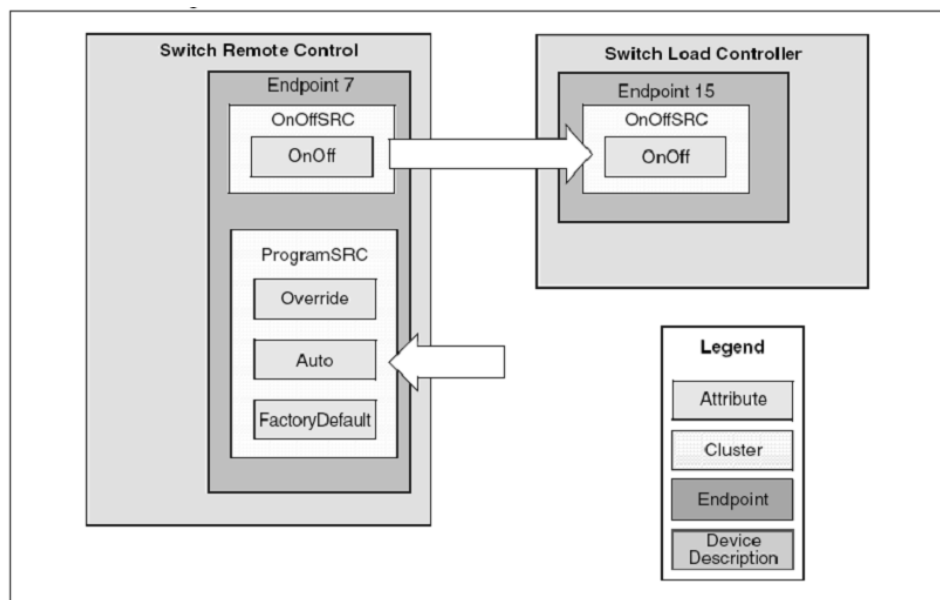


Figura 9 Arquitetura do *profile* do ZigBee™ [2]

Como é ilustrado na Figura 9, cada dispositivo tem apenas um *endpoint*. O *Switch Load Controller* (por exemplo, uma luz) tem um *cluster* de entrada no *endpoint*. O *Switch*

Remote Control (por exemplo, um interruptor) tem um *cluster* de saída e um *cluster* de entrada no seu *endpoint*. É importante notar que os dados apenas fluem ao nível do *cluster*.

4.8. ESTRUTURA DAS TRAMAS DO ZIGBEE™

A norma 802.15.4 prevê quatro tipos de tramas: tramas de dados, tramas de confirmação (*ack*), tramas de comando MAC e tramas *beacon*. Todas estas tramas têm em comum, na sua estrutura, o facto de começarem com um cabeçalho de sincronização com cinco bytes (B) de comprimento, sendo seguido pelo cabeçalho da camada física PHY, com um byte de comprimento. Após estes dois cabeçalhos, a estrutura do cabeçalho MAC varia em função do tipo de trama. Apesar da estrutura do cabeçalho MAC variar devido aos quatro diferentes tipos de trama possíveis, o seu tamanho é sempre 127 B, o que somando com os seis bytes dos outros dois campos da camada física, resulta num tamanho total de 133 B da trama do protocolo ZigBee™.

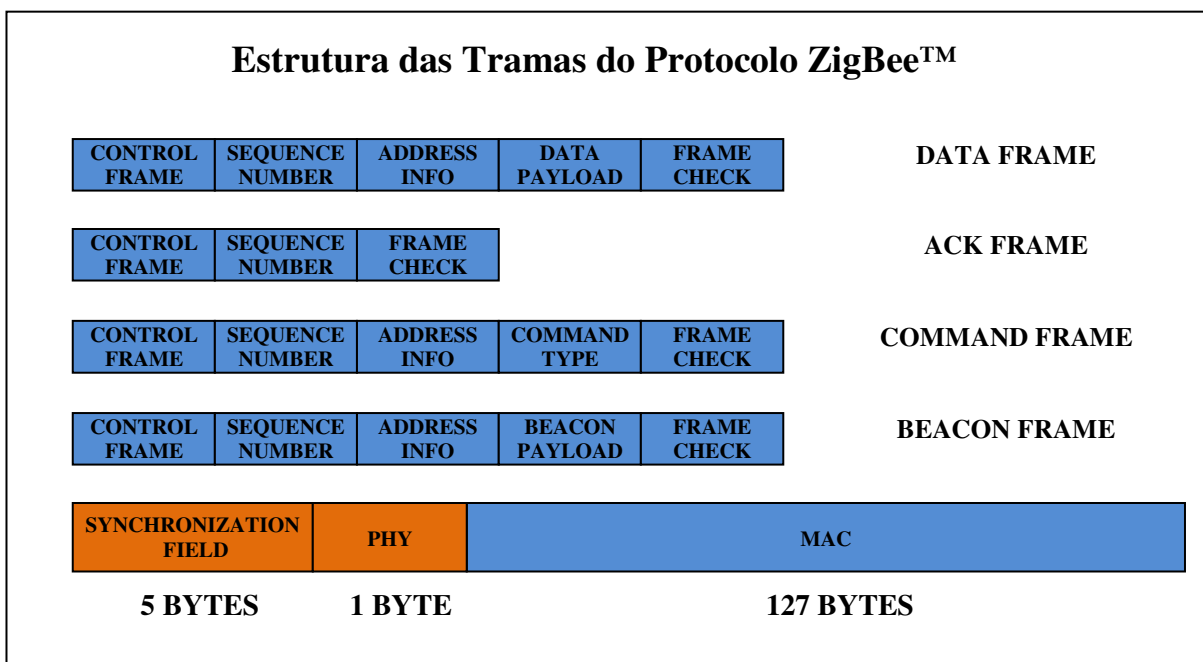


Figura 10 Estrutura das tramas do protocolo ZigBee™

Quanto à estrutura da trama de dados (*Data frame*) é composta por um campo *data payload* de 102 B de tamanho máximo, que é preenchido pela sub camada APS. Esta trama é numerada para garantir a sequência correcta de envio, existindo para este efeito um campo designado de *Sequence number*, cujo tamanho é de um byte. Existe ainda um campo designado de *Frame Check*, com comprimento de dois bytes para garantir que as tramas são recebidas sem erros, um campo com informações sobre endereço do emissor/receptor, cujo tamanho máximo é de vinte bytes. A informação neste campo é preenchida pela camada de rede NWK. Existe ainda o campo *Frame Control* com dois bytes.

A trama do tipo *acknowledgement (ack)* é utilizada pelo protocolo sempre que é utilizada confirmação pelo dispositivo. É composta pelos campos *Frame Check*, *Sequence Number* e *Frame Control*.

Relativamente à trama *MAC command*, é utilizada para controlo e configuração de nós remotos. A sua estrutura é igual à trama de dados, diferenciando-se desta pelo facto de não ter um campo *APS payload*, mas sim um campo *Command type*. Quanto aos tamanhos dos campos, são os mesmos que a trama de dados. Por fim, o tipo de trama *beacon* é utilizada para sincronismo pelos nós quando saem do modo de poupança de energia.

5. DESENVOLVIMENTO DO PROJECTO

Nesta secção é apresentado todo o desenvolvimento prático deste projecto, incluindo técnicas de controlo e softwares utilizados.

5.1. ESTRUTURA DE REDE ADOPTADA PARA O PROJECTO

Depois de um estudo detalhado dos requisitos deste projecto, verificou-se que a arquitectura de rede mais adequada para este projecto é a estrutura em malha, pois é, de todas, a que permite melhores desempenhos de comunicação, tendo em conta que neste caso haverá um grande número de dispositivos “pendurados” na rede e a velocidade de transmissão de dados é um factor importante. Esta topologia de rede também permite a existência de dispositivos configurados como Routers, que possibilitam que dispositivos muito afastados (fora do alcance directo) se comuniquem.

Assim, na rede de comunicação planificada para este projecto, haverá um coordenador de rede, vários dispositivos configurados como *Routers*, e vários dispositivos terminais (FFD e RFD). Todos os dispositivos presentes na rede, à excepção dos configurados como RFD (representados a preto na Figura 11), terão a função de controlar a cor de uma janela diferente e fisicamente serão instalados junto da mesma.

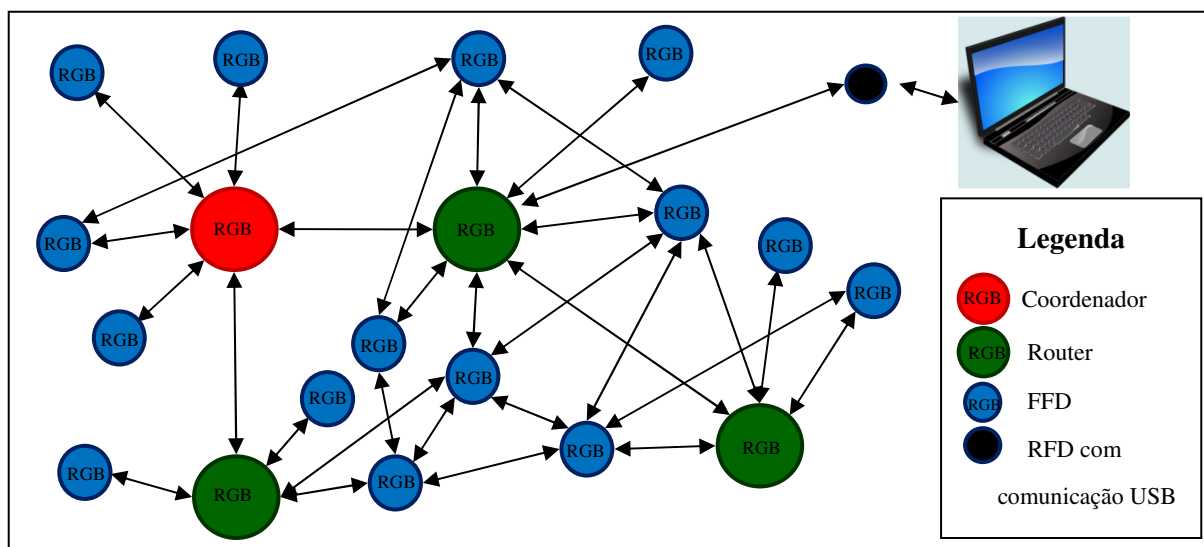


Figura 11 Configuração de rede adoptada para o projecto: Malha (*Mesh*)

Neste projecto optou-se por não utilizar dispositivos configurados como RFD para a recepção de dados, porque as políticas de gestão de energia que estes possuem, tornam a sua velocidade de comunicação mais lenta. Por outro lado, nenhum dos dispositivos ligados à rede com função de receber dados é alimentado a bateria, o que faz com que a gestão energética de cada dispositivo não seja um factor crítico, quando comparado com a eficiência de comunicação entre dispositivos dentro da rede.

O dispositivo que permite fazer a comunicação com o computador é configurado como RFD e é o único que é portátil, permitindo mobilidade em toda a área de alcance da rede. Desta forma, o utilizador pode controlar a cor das janelas em qualquer ponto do edifício, através de um computador portátil, por exemplo. Neste caso, a configuração como RFD permite ao utilizador obter uma maior autonomia da bateria do seu portátil.

5.2. DESCRIÇÃO DO HARDWARE

No desenvolvimento deste projecto foram desenvolvidos dois desenhos de *hardware*, com algumas diferenças resultantes das diferentes funções que os dispositivos têm na rede (Ver Anexos B e C).

Os dispositivos que permitem a comunicação entre o computador e a rede ZigBee™ utilizam o protocolo de comunicação USB, para fazer a comunicação com o computador.

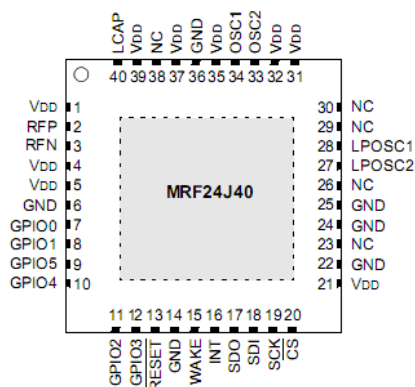
Todo o *hardware* deste dispositivo é, por tanto, alimentado via USB e deverá ter um consumo reduzido, para aumentar a autonomia da bateria, no caso de o computador ser um portátil.

Por outro lado temos os dispositivos que controlam a cor das fitas de LED RGB, que têm que ter o *hardware* de controlo de cada componente da cor e lidar com correntes altas. Este dispositivo, ao contrário do anterior, não é pensado para ser móvel, pois cada um será instalado junto de cada janela, para controlar a respectiva cor e eventualmente exercer outras funções na rede (Coordenador ou *Router*).

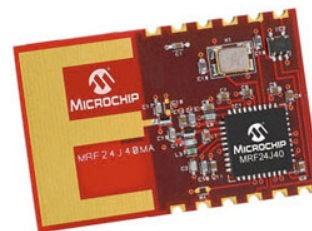
Todos os dispositivos da rede têm em comum o facto de possuírem o mesmo *transceiver* de comunicação (MRF24J40 RF 2.4GHz) e o mesmo processador (PIC18F2620) para correr a *stack* da Microchip[®], variando os restantes componentes com as funções que o dispositivo tem na rede. Nos tópicos que se seguem será feita uma abordagem mais detalhada a cada um destes componentes de *hardware*.

5.2.1. TRANSCEIVER ZIGBEE™

No desenvolvimento deste projecto, foi utilizada a *Stack* e o *transceiver* MRF24J40MA/MB da Microchip[®], ilustrado na Figura 12b.



(a) [7]



(b) [8]

Figura 12 a) Transceiver de Comunicação ZigBee™. b) Antena MRF24J40MA.

Esta usa um transceptor MRF24J40 RF 2.4GHz, ilustrado na Figura 12a, fabricado pela Microchip[®] e compatível com a norma IEEE 802.15.4. para uma frequência de banda de 2.4GHz. É compatível com outros protocolos, como o MiWi[™] e o MiWi[™] P2P.

As suas principais características são:

- Preparado para trabalhar com osciladores de 20 MHz e 32.768 KHZ;
- Comunicação através do protocolo SPI (4 ligações);
- Modo de poupança de energia (modo *Sleep*);
- Consumos na ordem dos 18 mA (modo recepção) e 22 mA (modo transmissão).

O Microcontrolador escolhido para correr a *Stack* foi o PIC18F2620, pois é um dos recomendados pela Microchip[®]. Sendo uma das vantagens da rede ZigBee[™] o baixo consumo de energia, é muito importante que o micro controlador tenha um consumo baixo de energia. Uma das principais características deste micro é a *anowatt technology*, que consiste num método de alternância entre *clocks* com o objectivo de minimizar o consumo energético. O micro controlador pode utilizar o oscilador de *clock* principal ou usar o oscilador do *Timer1*.

Quando o micro controlador funcionar à velocidade máxima utiliza o *clock* principal e quando tiver pouca actividade, e por isso não necessitar de funcionar a uma velocidade tão elevada, utiliza o *clock* fornecido pelo *Timer1*, funcionando a uma frequência mais baixa, o que poderá representar um decréscimo do consumo de energia que poderá chegar aos 90% durante a execução.

5.2.2. INTERFACE DE COMUNICAÇÃO COM O COMPUTADOR – USB

A interface de comunicação entre um computador e a rede de comunicação ZigBee[™] é feita com recurso a um nó da rede, que foi configurado como um RFD “*end device*”. Para

este dispositivo, foi necessário desenvolver uma interface USB, uma vez que actualmente os computadores já não vêm equipados de fábrica com porta de comunicação serie.

Um dos problemas que se verificou no desenvolvimento desta aplicação, foi o facto de a *Stack* da Microchip[®] ocupar a memória do PIC18F4620 na sua quase totalidade e feita uma pesquisa de outro micro controlador PIC com mais memória e com interface USB integrada no próprio chip, verificou-se que não haviam outras opções dentro da família 18.

Um microcontrolador que possui interface USB integrada no próprio chip dentro da família dos PIC 18 é o PIC18F45/2550 e, apesar de não ter memória suficiente para correr a *Stack* da Microchip[®], foi a melhor opção em alternativa a integrados específicos, uma vez que é mais rápido em termos de processamento e poderá ser utilizado como processador auxiliar para o controlo da comunicação USB com o PC. Desta forma, a gestão dos recursos de memória do PIC18LF2620 ficou facilitada e criou-se um sistema mais modular. Por outro lado, este microcontrolador pode funcionar não só com alimentação de cinco Volts como também de 3,3Volts.

O diagrama da figura seguinte (Figura 13) representa esquematicamente a estrutura de comunicação desenvolvida.

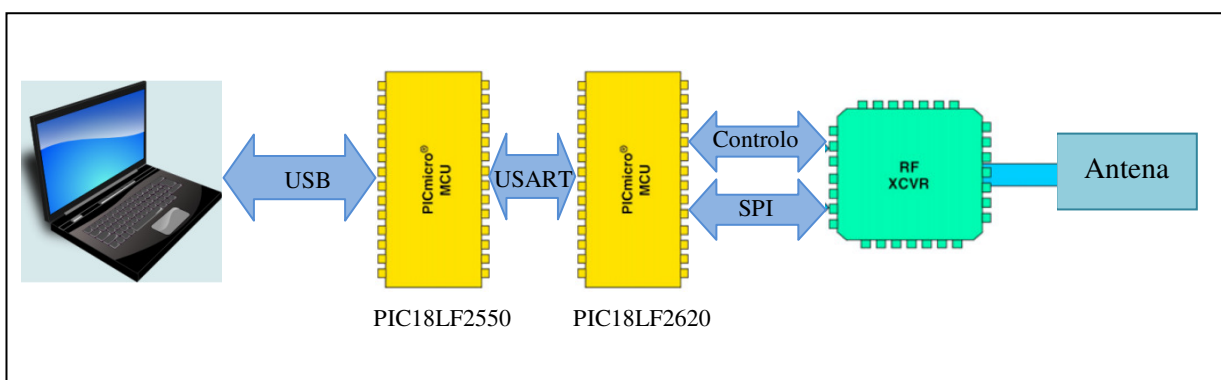


Figura 13 Interface de comunicação com o computador

O PIC18LF2550, é um microcontrolador pertencente à família PIC18 da Microchip[®], e pode funcionar a uma velocidade de até 48 MHz sendo utilizado nesta aplicação com a principal vantagem de permitir a comunicação via USB. Este microcontrolador recebe os dados via USB do computador e transmite-os através da USART por RS232 para o microcontrolador PIC18LF2620. Este, por sua vez, envia estes dados para a rede de comunicação.

Para configurar a comunicação USB no PIC18LF2550 foi utilizado o compilador mikroC™ da mikroElektronica®, ver Figura 14. Este possui várias bibliotecas de funções já desenvolvidas para programação da estrutura de comunicação USB do PIC com o PC e um terminal gráfico que facilita a configuração da mesma.

O mikroC™, à semelhança de outros compiladores, trabalha com o conceito de projecto, ou seja, sempre que se pretende compilar um determinado código, este tem que estar inserido num projecto, no qual é necessário configurar logo à partida cada um dos registos de funções especiais do microcontrolador, janela representada do lado direito da Figura 14.

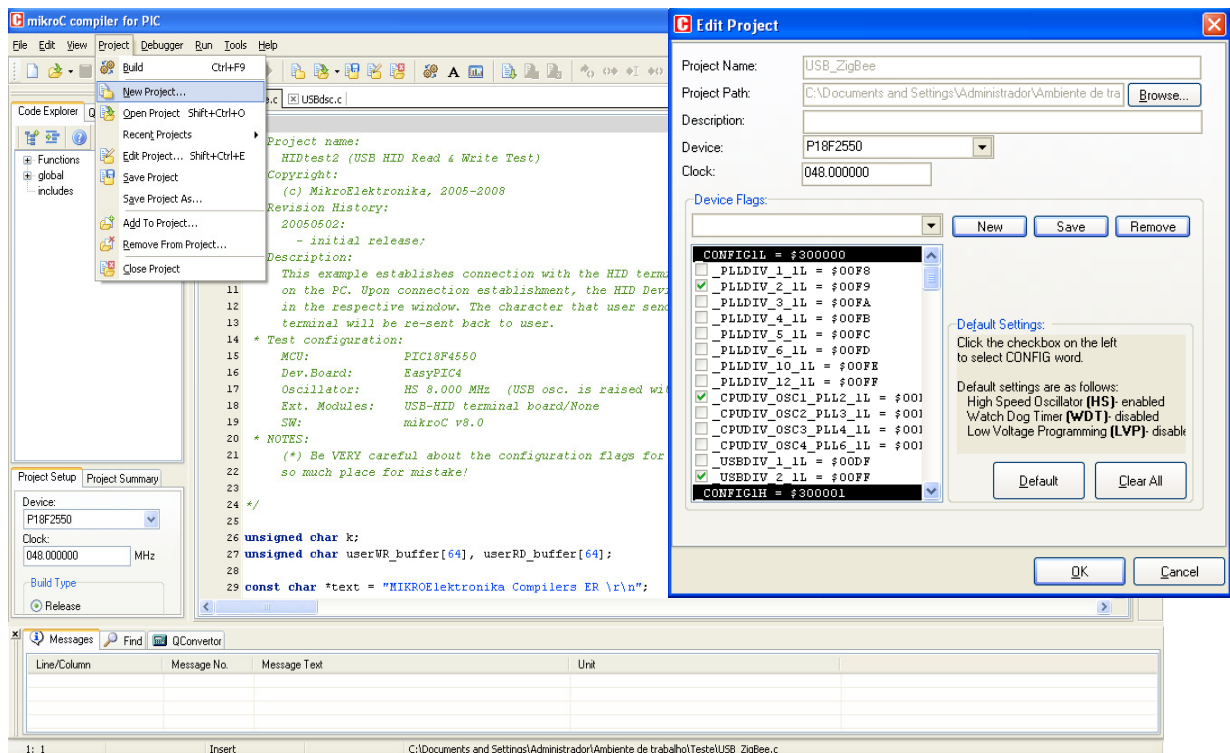
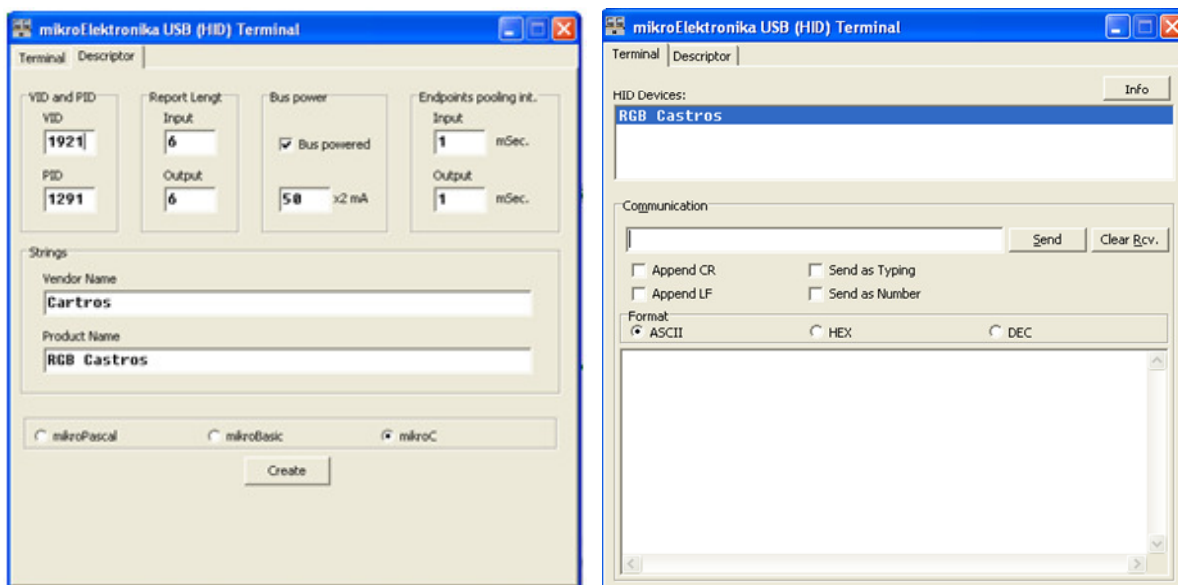


Figura 14 Software de programação de PIC mikroC™ da MikroElektronica®

O ajuste dos bits de configuração dos registos de funções especiais (SFR) do PIC é feito através do campo “*Device Flags*”. Estes não podem ser configurados na estrutura do programa. A configuração destes registos é explicada na tabela presente no Anexo A.

Uma vez criado o projecto, é necessário criar um ficheiro descritor para adicionar ao projecto. A função deste ficheiro é fazer a identificação do dispositivo USB assim que o mesmo for conectado ao PC. Para configurar este ficheiro, o mikroC™ vem equipado com um terminal “*HID Terminal*” que permite, não só criar esse ficheiro, como também testar o funcionamento da comunicação via USB, Figura 15 a e b.



a)

b)

Figura 15 HID Terminal – Ferramenta do *microC*TM para apoio ao desenvolvimento para USB.

É neste terminal que são definidos por exemplo o Vendor ID (VID) e o Product ID (PID) do dispositivo. Estes dois parâmetros são fornecidos directamente pela USB.org, e sendo pagos, optou-se, numa primeira fase, por colocar dois valores provisórios.

Outra informação que é necessário ajustar é o tamanho do *buffer* de entrada e de saída, através do campo *Report Length*. Nesta aplicação em concreto, a trama de comunicação terá um tamanho base de seis bytes, necessários para enviar o código RGB e funções de controlo.

É também importante definir, no campo *Bus power*, se o dispositivo será do tipo *bus powered*, ou seja, alimentado pela própria porta USB ou *self powered*, alimentado por uma fonte própria, à qual o equipamento é ligado. Neste caso, pretende-se que a alimentação seja feita via USB, pelo que, a opção *bus powered* foi marcada. É de notar que no campo abaixo da configuração da alimentação do dispositivo, é possível definir a corrente máxima que o dispositivo irá utilizar. Neste caso, pela análise dos consumos, prevê-se que este seja inferior a 100mA (50mA x 2).

Os campos *Strings* de *Vendor Name* e *Product Name* permitem a identificação do dispositivo sempre que este for conectado à porta USB.

Depois de criado e guardado no directório do projecto com o nome “usbdisc.c”, este ficheiro deve ser adicionado ao projecto. Para além deste ficheiro, é necessário também adicionar os ficheiros “var.h” e “definit.h”, presentes no directório “...\mikroelektronika\microC\examples\extra_examples\hid-library”.

Feitas todas as configurações, qualquer computador com interface USB pode agora identificar e comunicar com este Hardware. Falta agora desenvolver o programa que irá fazer a interface entre o computador e o PIC que corre a *Stack*, o qual será apresentado num dos capítulos seguintes.

5.2.3. UTILIZAÇÃO DO PWM PARA CONTROLO DA COR RGB DE CADA JANELA

Depois de desenvolvida a estrutura de comunicação sem fios, foi importante pensar qual seria a interface de controlo de potência mais adequada para a iluminação decorativa. Tendo em conta que para cada janela do edifício vai ser utilizado um *transceiver* ZigBee, que vai ter a função de receber o código RGB correspondente à cor a exibir na respectiva janela, cada nó da rede terá que permitir o controlo da intensidade luminosa correspondente a três fontes de cor (Vermelho, Verde e Azul).

O diagrama da figura seguinte (Figura 16) representa esquematicamente a estrutura de comunicação desenvolvida para o *hardware* de controlo da cor RGB das fitas de LED.

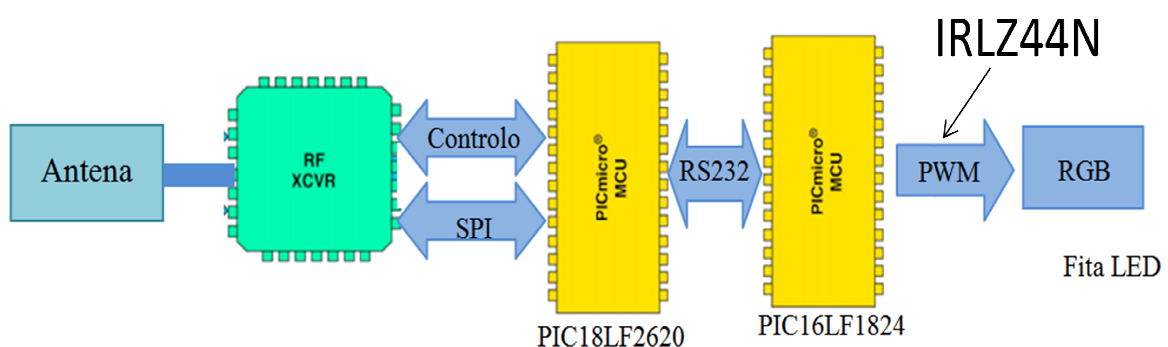


Figura 16 Interface de controlo das fitas de LED.

Sendo que a alimentação dos LED é feita com corrente contínua, uma das técnicas mais utilizadas nos dias de hoje, para fazer o controlo da cor de sistemas de iluminação de LED

RGB é o controlo PWM, sigla de Pulse Width Modulation, ou seja, Modulação por Largura de Pulsos. Trata-se de um método para controlar a energia - e consequentemente a potência - entregue à carga em dispositivos que trabalham em sistemas de Corrente Contínua (DC).

Sendo que cada LED RGB é na realidade constituído por três LED no mesmo encapsulamento, cada controlador terá que produzir três sinais PWM, para tornar possível o controlo individual da intensidade luminosa de cada uma destas cores.

A análise da Figura 17 permite perceber como a técnica de controlo PWM é realizada. Quanto maior a duração do Tempo Ligado (tempo em que o PWM permanece em *On* - T_{on}), maior a energia entregue à carga. Um PWM é, por definição, um sistema de frequência (f) e período (T) constantes e de Largura de Pulso (ciclo activo) ajustável. Denominando de T_{off} o tempo em que a carga é mantida desactivada, o período total do PWM será:

$$T = T_{on} + T_{off} . \quad (1)$$

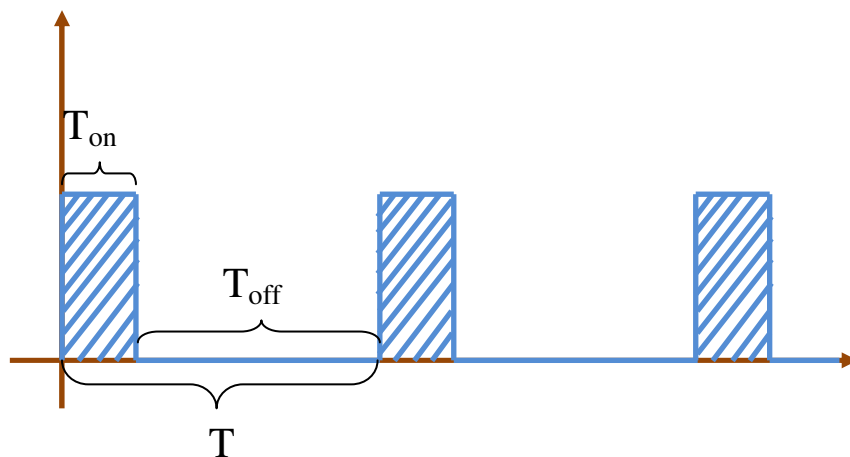


Figura 17 Forma de onda típica de um sistema PWM.

O tempo em que a carga é mantida activa “ T_{on} ” é também definido como Ciclo Activo (*Duty Cycle* - D.C.).

Também é importante abordar algumas considerações sobre o estágio de potência. A escolha foi por um transistor MOS de Efeito de Campo (MOSFET) da International Rectifier[®]: o IRLZ 44 N. A letra “L” no prefixo IRLZ indica que o seu gate é adequado para controlos lógicos e que o mesmo pode ser saturado com tensões de 3,3 volts, o que não ocorre com os tipos IRF ou IRFZ. Nestes, um valor tão baixo de tensão pode não levá-lo à saturação, fazendo o MOSFET operar na região linear, o que poderia provocar um considerável aquecimento do componente, de acordo com o *datasheet* do transistor IRLZ44.

Operando na condição de Corte e Saturação, o aquecimento é mínimo e não ocorre perda de energia por dissipação de potência na junção Dreno – *Source*. Assim, o transistor pode manipular correntes mais altas (de até 47 A, segundo o *datasheet* do fabricante), exibindo (quando saturado) uma resistência DRENO-SOURCE (R_{ds}) de apenas 0,022ohms. A tensão máxima admissível entre D e S é de 55 volts.

5.3. DESCRIÇÃO DO *FIRMWARE*

No desenvolvimento deste Projecto foi desenvolvido *hardware* para quatro dispositivos de comunicação da rede.

- Coordenador da rede;
- Dispositivo RFD com interface USB para comunicação com um computador;
- Dispositivo *Router e terminal*;
- Dispositivo FFD terminal,

5.3.1. ESTRUTURA BASE DA STACK ZIGBEE™ DA MICROCHIP[®]

Todas as funções implementáveis pela *stack* são invocadas através de primitivas, que não são mais do que instruções que são dadas para esta executar. Cada processo da *stack*, como formação de rede, procura de rede, envio de dados, recepção de dados, processos relacionados com acesso ao meio ou nível físico tem uma primitiva associada e é frequente um pedido que despoleta uma primitiva levar à execução de outras primitivas auxiliares para a execução desse mesmo pedido.

Quando o processador arranca, são executadas algumas funções e inicializações que vão possibilitar o seu funcionamento normal.

É inicializada a variável `currentPrimitive` com a primitiva `NO_PRIMITIVE`. Esta primitiva será sempre invocada quando a *stack* termina a tarefa de uma primitiva e não é lançado um pedido de execução de uma nova primitiva. Esta primitiva vai permitir a execução de funcionalidades introduzidas pela aplicação.

```
void main(void)
{
  (...)
  currentPrimitive = NO_PRIMITIVE;

  ConsoleInit();
  HardwareInit();

  (...)

  ZigBeeInit();

  (...)

  while (1)
  {
    (...)
    ZigBeeTasks( &currentPrimitive );
    ProcessZigBeePrimitives();
    ProcessNONZigBeeTasks();
  }
}
```

Em seguida é feita a inicialização das configurações da porta série, através da chamada da função `ConsoleInit`, que permitirá criar a interface de comunicação entre o módulo de comunicação ZigBee e o microcontrolador que fará a comunicação USB com o PC.

Em seguida é chamada da função `HardwareInit`, que faz a inicialização de todas as configurações de hardware, necessárias para o funcionamento da *stack* - configuração de portos do micro e do módulo de comunicação SPI para comunicação com a antena. A função `ZigBeeInit` inicializa a *stack*.

Depois destas inicializações iniciais, essenciais ao funcionamento da *stack*, todo o processamento é feito dentro de um ciclo infinito, que só será interrompido se for efectuado um *reset* externo ao microcontrolador.

Dentro deste ciclo são executadas três funções, que organizam o código ao nível da aplicação:

- A função `ZigBeeTasks`: Faz o processamento das várias primitivas ao nível da *stack*;
- A função `ProcessZigBeePrimitives`: Faz a gestão de qual será a próxima primitiva a ser executada pela *Stack*;
- A função `ProcessNONZigBeeTasks`: Permite processar todas as funções da aplicação que não estejam relacionadas com a *stack* e será executada quando não há mais processos relacionados com a comunicação sem fios – a *stack* invoca esta função de 18 em 18ms. É no corpo desta função que serão colocadas todas as funções a executar pela aplicação.

5.3.2. FORMAÇÃO DE REDE

A tarefa de formação de rede está sempre a cargo do coordenador, tendo sempre que existir um em cada rede que se venha a formar.

Quando a primitiva é inicializada com o parâmetro `NO_PRIMITIVE`, ao ser executada, é verificado se já existe uma rede formada. Se já existe rede formada, a *stack* fica à espera de um pedido por parte do utilizador, através da `USARTE`, que faça despoletar outra primitiva e executar funções adicionadas à *stack*, enquanto não houver nova primitiva a executar. Caso não haja rede formada, é iniciado o processo de formação de rede.

A *stack* sabe se a rede está formada ou não através da *flag* `ZigBeeStatus.flags.bits.bNetworkFormed`. A primitiva deixa de ser `NO_PRIMITIVE` para passar a ser `NLME_NETWORK_FORMATION_request`.

5.3.3. ENTRADA DE UM NOVO DISPOSITIVO NA REDE

Assim como no código do coordenador, para um `end device`, a primitiva é inicializada com o parâmetro `NO_PRIMITIVE`, no entanto como se trata de um `end device`, a *stack* em vez de verificar se existe rede formada vai querer saber se o dispositivo já está inserido ou não em alguma rede. A *flag* que permite esta informação é a *flag*

ZigBeeStatus.flags.bits.bNetworkJoined. Caso não esteja inserido numa rede, a primitiva passa a ser NLME_JOIN_request:

```
case NO_PRIMITIVE:
    if (!ZigBeeStatus.flags.bits.bNetworkJoined)
    {
        if (!ZigBeeStatus.flags.bits.bTryingToJoinNetwork)
        {
            (...)

            currentPrimitive = NLME_NETWORK_DISCOVERY_request;
        }
    }
}
```

No final do processo, a *stack* fica a saber se o *end device* entrou com sucesso na rede ou não, através da *flag* *params.NLME_JOIN_confirm.Status*. Em caso negativo inicia o processo novamente. Em caso positivo, invoca a primitiva *APSDE_DATA_request* para anunciar ao coordenador que entrou na rede.

```
case NLME_JOIN_confirm:
    if (!params.NLME_JOIN_confirm.Status)
    {
        (...)

        currentPrimitive = NO_PRIMITIVE;

        (...)
    }
    break;
```

5.3.4. ENVIO DE DADOS DO COMPUTADOR PARA A REDE – PEN ZIGBEE™

Para controlar remotamente a cor de cada janela do edifício, foi criado um dispositivo ZigBee™ com interface de comunicação via USB para ligação a um computador – PEN ZigBee™. Este dispositivo foi configurado como RFD para que o sistema possa beneficiar de todas as potencialidades do protocolo ZigBee™, nomeadamente no que respeita aos baixos consumos, importantes em dispositivos portáteis. O utilizador pode alterar a cor de cada janela através de um computador portátil, conectando o dispositivo de comunicação ZigBee™ (PEN ZigBee™) à porta USB e utilizando a aplicação de Software desenvolvida para este efeito – aplicação *RGBCastros* que será apresentada num dos capítulos seguintes.

Quando o dispositivo que está conectado ao computador recebe dados enviados pelo *Software da aplicação*, verifica qual foi a opção, do menu de opções, que foi invocada. Desta forma poderá saber que tratamento dar à informação recebida e quais as funções da *stack* a ser evocadas. Este menu possui as seguintes opções:

- 2: Solicitação de dados de outro dispositivo
- 3: Solicitação de dados de um grupo de dispositivos
- 4: Enviar dados para outro dispositivo
- 5: Enviar dados para um grupo de dispositivos
- 6: Adicionar / Remover Dispositivo para / de um grupo
- 7: Informação dos dispositivos da rede

Para não perder a possibilidade de depuração do código da *stack* da Microchip[®], através da USART, bem como a aplicação de exemplo, ver Figura 18, optou-se por não retirar estas opções da *Stack*. Ao invés disso foram adicionadas novas opções similares, mas “limpas” de Strings de texto de ajuda à depuração de código. Assim, sempre que o software RGBCastros se comunica com a PEN ZigZee[™], fá-lo-á indicando o identificador do serviço requerido, enviado no primeiro byte de trama de dados, ver Figura 18.

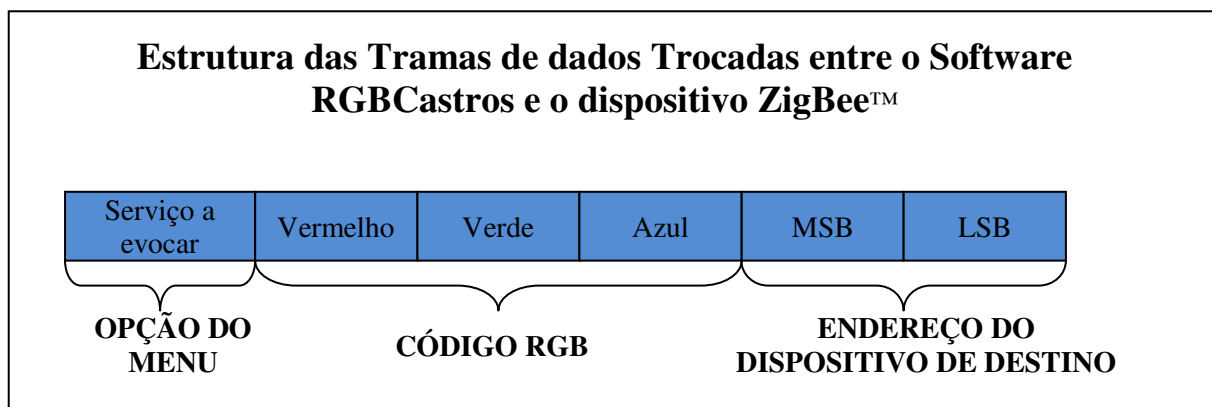


Figura 18 Tramas de dados Trocadas entre o Software RGBCastros e o dispositivo ZigBee[™]

Definiu-se que esta trama é composta por seis bytes de dados, sendo que o primeiro byte é o identificador da opção da *stack* a evocar e sempre que se pretender enviar um código

RGB para um determinado dispositivo, é utilizada a opção “a” do MENU. A nível da aplicação *RGBCastros*, esta opção é representada pelo hexadecimal correspondente (o caracter ‘a’ em hexadecimal é representado por 61, tabela ASCII).

A nível da *stack* as várias opções do menu são tratadas pela função `ProcessMenu()`, que vai comparar o primeiro byte da trama de dados recebida com os vários *cases* da função `switch(c)`, para determinar o tratamento a dar à informação recebida.

Sendo a opção “a”, os três bytes do código RGB são carregados para o `TxBuffer`, *buffer* que contem os dados a transmitir e é carregado o endereço de destino dos dados no vector `v[0]`.

```
void ProcessMenu( void )
{ (...)
  switch (c)
  { (...)
    case 'a':
      (...)
      for(i = 0; i < temp2; i++)
      {
        while (!ConsoleIsGetReady());
        TxBuffer[TxData++] = ConsoleGet();
      }
      params.APSDE_DATA_request.DstAddress.
      ShortAddr.v[1]=GetHexDigit();

      params.APSDE_DATA_request.DstAddress.
      ShortAddr.v[0] = GetHexDigit();

      params.APSDE_DATA_request.SrcEndpoint      = 1;
      params.APSDE_DATA_request.DstEndpoint      = 240;
      (...)
      ZigBeeBlockTx();
      currentPrimitive = APSDE_DATA_request;
      break;
    (...)
  }
}
```

Os parâmetros para que a primitiva seja executada, como o end point de origem e destino, são inicializados antes da *stack* executar a função `ZigBeeBlockTx`. Esta função bloqueia o buffer de transmissão da porta série. Depois disso é executada a primitiva para envio de dados `APSDE_DATA_request`.

A Figura 19 contem os fluxogramas das rotinas que correm nos microcontroladores da PEN ZigBee. O PIC18LF2550 faz a interface entre o computador e o PIC18LF2620 e este, por sua vez, desempenha as funções de comunicação ZigBee™, correndo a *stack* da Microchip®.

```
while (1)
{
    k = HID_Read();
    i = 0;
    while (i < k)
    {
        ch = userRD_buffer[i];
        userWR_buffer[i] = ch;
        Delay_ms(15);
        USART_Write(ch);
        i++;
        if(k==i){while(!HID_Write(&userWR_buffer,6));}
    }
}
```

Sempre que é enviada uma trama de dados do computador para o PIC18LF2550, a função `HID_Read()` retorna o numero de bytes recebidos para a variável “K”, e é feita a leitura dos dados do buffer de entrada USB `userRD_buffer[i]`. Estes dados são enviados logo de seguida pela USART para o PIC18LF2620 e para o computador, para confirmação de trama bem recebida.

O PIC18LF2620 recebe os dados da USART (código RGB e endereço do dispositivo de destino) e envia-os para a rede ZigBee™.

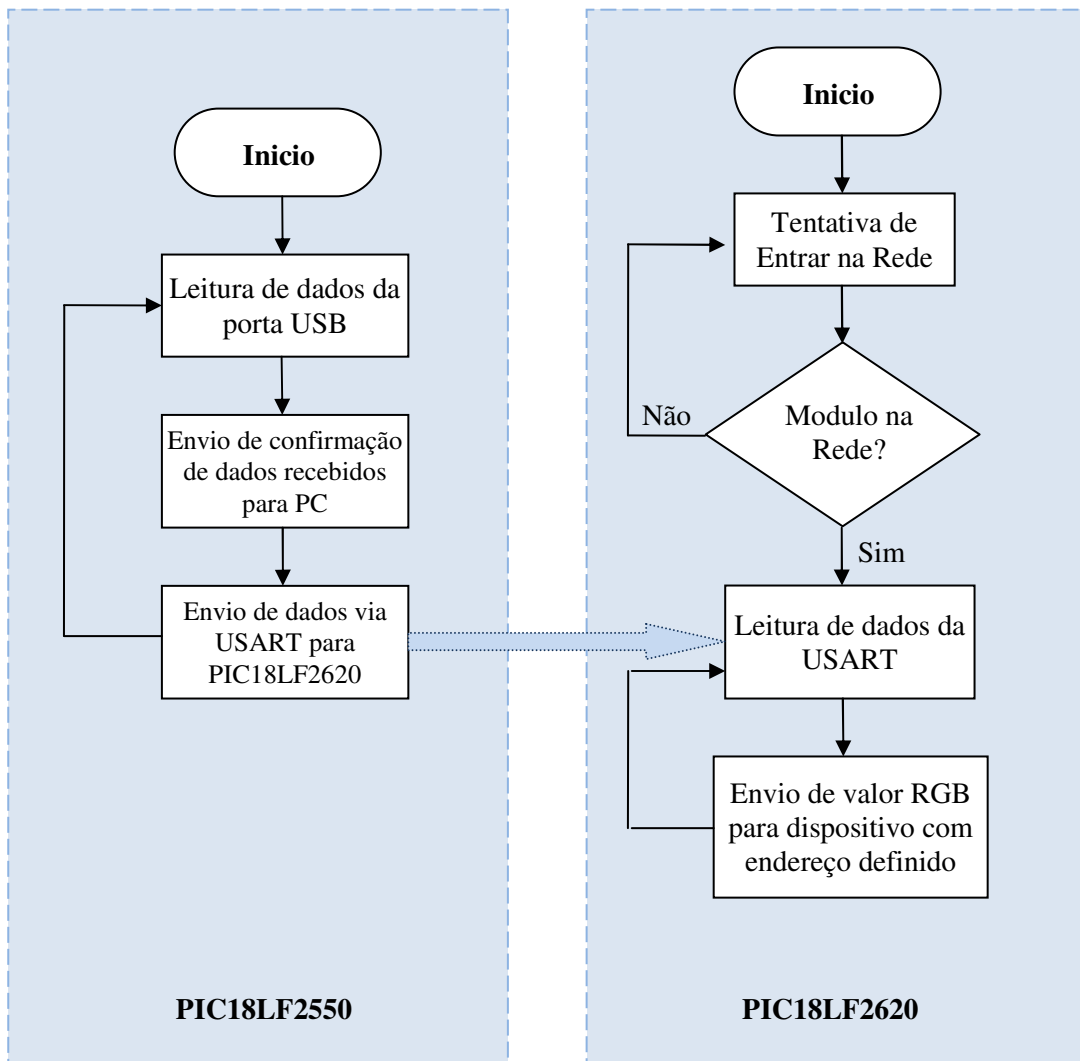


Figura 19 Rotinas da PEN ZigBee™

5.3.5. RECEPÇÃO DOS DADOS PELOS NÓS DA REDE

Sempre que um dispositivo da rede recebe dados dos end devices a primitiva que a *stack* vai executar é a `APSDE_DATA_indication`.

O nó da rede verifica qual o `endpoint` de destino da informação. Caso não seja o ZDO executa a opção `default`, que lida com toda a informação destinada a um `endpoint` para o qual não esteja especificado num procedimento específico. É aqui que o dispositivo está a receber a informação da cor RGB que lhe é endereçada.

```

case APSDE_DATA_indication:
{
  (...)
  switch (params.APSDE_DATA_indication.DstEndpoint)
  {
    case EP_ZDO:
      (...)
      break;

    default:
      {
        (...)
        for(transaction=0;transaction<frameHeader;
            transaction++)
          {
            switch( clusterID.Val )
            {
              case TRANSMIT_COUNTED_PACKETS_CLUSTER:
                {
                  (...)
                  for(i = 0; i < PacketLen-2; i++)
                    ConsolePut (APLGet ());
                }
              break;
            }
          }
      }
  }
}

```

Uma vez recebida essa informação, o nó da rede envia-a, via USART, para o microcontrolador PIC16LF1824, para que este possa gerar o sinal PWM para os MOSFET que controlam as fitas de LED RGB. Isso é feito invocando a função `ConsolePut()`.

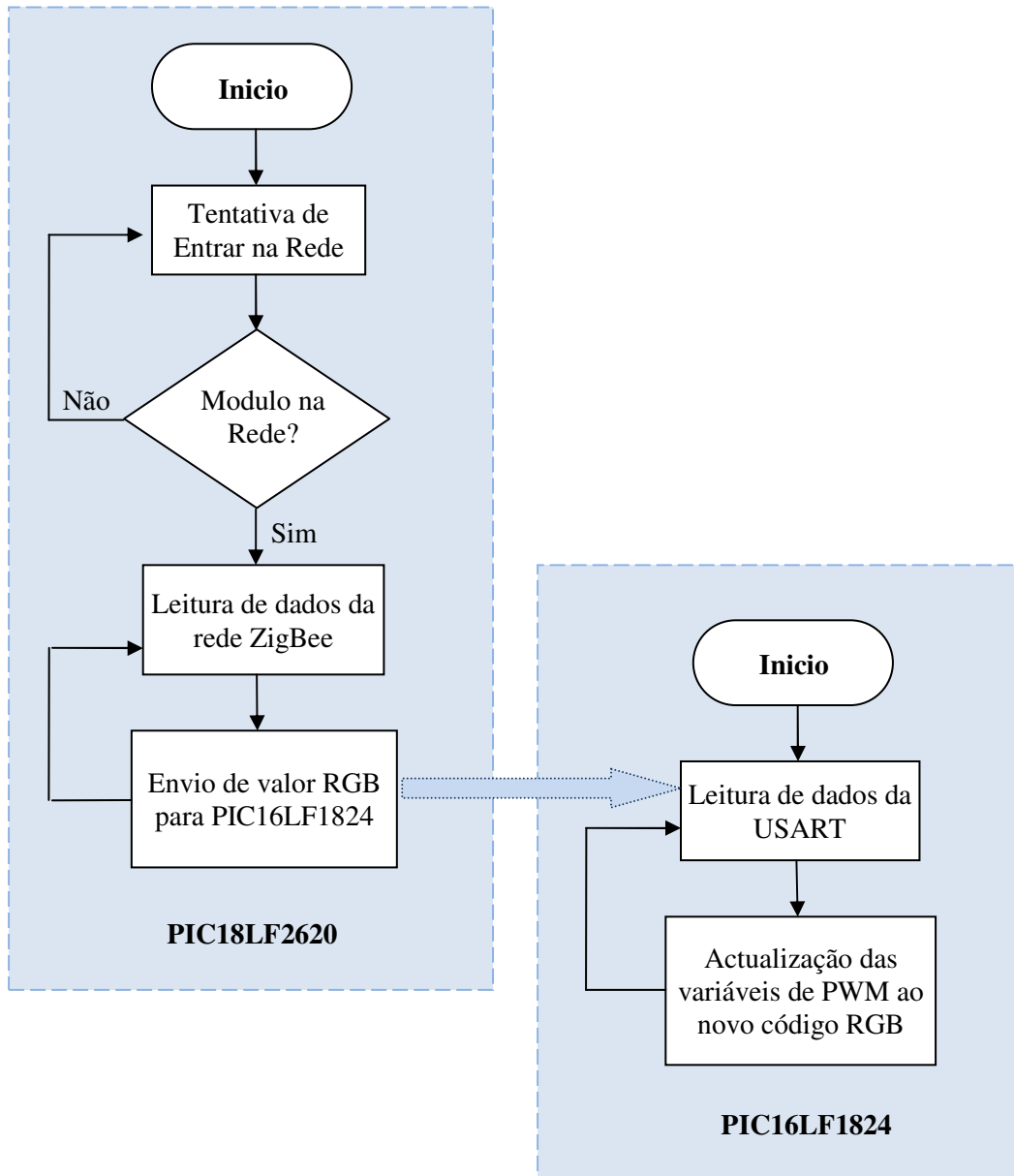


Figura 20 Rotinas de um nó da rede – Controlo RGB,

5.4. DESCRIÇÃO DO SOFTWARE DA APLICAÇÃO

Para tornar o projecto desenvolvido mais funcional e amigável para o utilizador, desenvolveu-se uma aplicação de *Software* para correr em ambiente Windows. A ferramenta de desenvolvimento utilizada foi o Visual Studio™ 2008 e a linguagem de programação utilizada para o desenvolvimento foi o C#.

Em termos funcionais, pretendia-se que a aplicação de *Software* permitisse controlar a cor de cada janela individualmente, possibilitando a criação de ambientes cromáticos para aplicar à fachada do edifício. Em seguida são listadas as características que se pretendiam para este Software:

1. Pretende-se que reconheça automaticamente a PEN ZigBee™, logo que esta seja conectada ao PC;
2. Possibilidade de criar programas com as sequências de passos das animações pretendidas;
3. Possibilidade de controlar os efeitos de cor de cada janela em tempo real;
4. Possibilidade de editar, apagar e criar novas sequências;
5. Possibilidade de endereçar os dispositivos que controlam a cor de cada janela individualmente para teste de uma determinada cor;
6. Desenvolver as opções de ajuda ao utilizador, através da criação do *Help*;

5.4.1. AMBIENTE GRÁFICO

Na Figura 21 é apresentada a janela principal do *Software* desenvolvido para controlar a cor de cada janela do edifício.

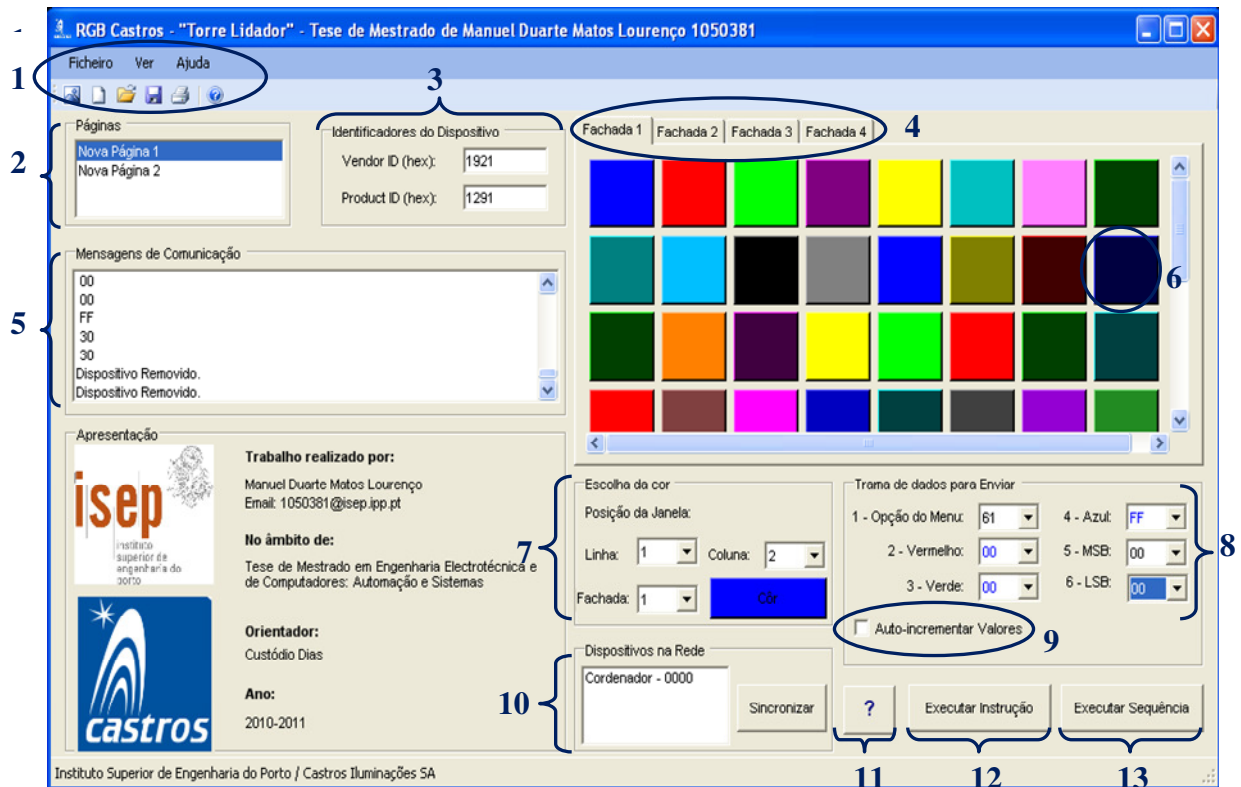


Figura 21 RGBCastros, Aplicação de software para edição de cenários cromáticos.

Em seguida são identificados os principais recursos, presentes na Figura 21, que o utilizador pode utilizar:

1. Controlos que permitem ao utilizador realizar opções como pesquisar dispositivo, Criar uma nova página, consultar tópicos de ajuda do *Software*, fechar a aplicação, entre outras opções;
2. Janela que dá ao utilizador a possibilidade de escolher o ambiente cromático a utilizar, de entre os previamente criados, guardados ou importados.*¹ Clicando no botão <Nova Pagina>, a aplicação abrirá a caixa de diálogo apresentada na Figura 22. Nesta janela, o utilizador pode definir os vários parâmetros de configuração do projecto: Nome do projecto, número de fachadas do edifício e número de janelas – em matriz de linhas por colunas.

*¹ Esta funcionalidade não foi totalmente desenvolvida até à data da apresentação deste projecto.

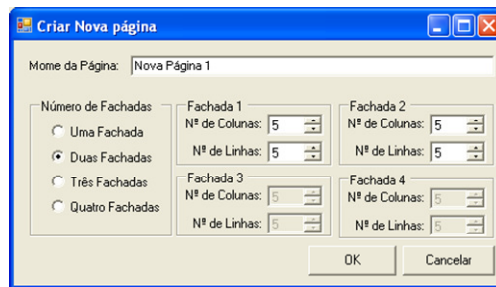


Figura 22 Janela de criação de um novo projecto.

3. Definição dos identificadores do dispositivo USB. A aplicação só vai comunicar com o dispositivo que estiver conectado à porta USB que tiver os identificadores presentes neste campo;
4. Separadores que permitem seleccionar a fachada que se pretende editar, testar ou ver;
5. Janela que retorna as várias mensagens informativas do *software* que poderão ser alertas, erros ou pontos de execução. Esta janela tem como principal função facilitar a depuração do *software* na fase de desenvolvimento, podendo posteriormente ser eliminada na versão a entregar ao cliente;
6. Botão identificador da janela a alterar;
7. Caixa de edição de cores – permite ao utilizador editar a cor das várias janelas, Figura 23;

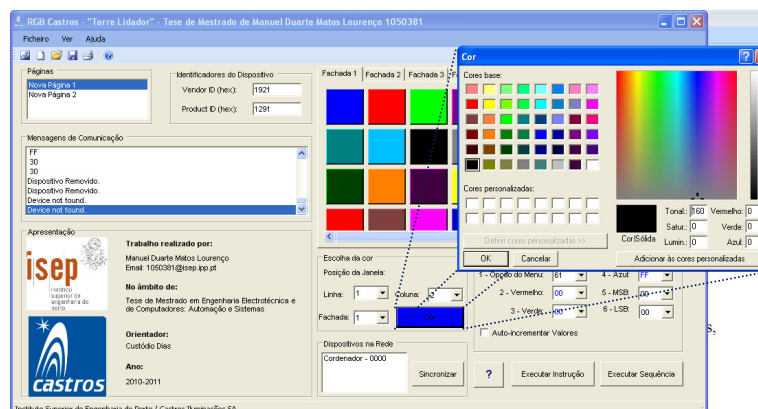
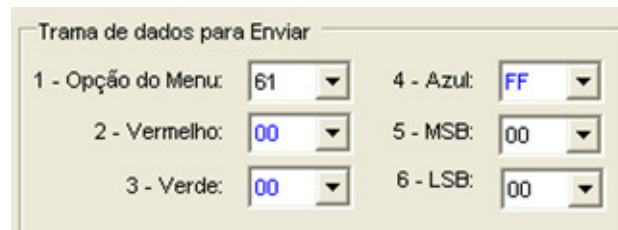


Figura 23 Edição da cor da janela presente na Fachada 1 que está na Linha 1 e Coluna 1.

8. Constituição da trama de dados que é enviada para o dispositivo USB, ver Figura 24.



Trama de dados para Enviar	
1 - Opção do Menu:	61
2 - Vermelho:	00
3 - Verde:	00
4 - Azul:	FF
5 - MSB:	00
6 - LSB:	00

Figura 24 Constituição da trama de dados que é enviada para o dispositivo USB.

O primeiro byte da trama de dados enviada pelo *software*, Figura 24 é relativo à opção do menu. Identifica qual será o tratamento que os bytes seguintes deverão ter. Os três bytes que se seguem são relativos ao código RGB e os dois últimos identificam o dispositivo para o qual esse código se destina.

9. Auto-incrementar Valores – se colocar um visto nesta opção, o *software* incrementa de forma automática o valor do endereço do dispositivo de destino dos dados a cada código RGB que envia.
10. Ao clicar na opção Sincronizar, o *software* pede ao *Hardware* uma listagem dos endereços de todos os dispositivos que estão na rede e apresenta-os na caixa de diálogo;
11. Botão de ajuda do *software* – Contém informação relevante sobre como utilizar o *software*;
12. Envia o código RGB seleccionado para o dispositivo correspondente;
13. Envia a sequência das várias cores para os vários dispositivos.

5.5. ANÁLISE DOS RESULTADOS

O sistema desenvolvido neste projecto foi validado através de alguns testes práticos, que permitiram testar a robustez e fiabilidade. Existiu no entanto a condicionante do número de dispositivos disponíveis para ensaio ser reduzido a três, o que não tornou possível a realização de testes com redes mais extensas e com vários dispositivos fora do alcance directo uns dos outros.

A nível das soluções de hardware utilizadas neste projecto, em todos os nós da rede ZigBee™ foi utilizado mais do que um processador, descentralizando-se assim o processamento da informação. As razões pelas quais se optou por este caminho foram várias. O facto de o PIC18LF2620 não possuir interface de comunicação USB sugeriu a utilização do PIC18LF2550 para fazer a interface com o computador. Este por sua vez não possuía memória suficiente para correr a *stack* ZigBee™ e pesquisando na família PIC18 não haviam alternativas com mais memória interna e periféricos de comunicação USB.

Por outro lado, o PIC18LF2620 também não possui módulos PWM para controlo dos MOSFET. Quando se tentou implementar o PWM por *software* verificou-se que a *stack* ZigBee™ da Microchip® implementava interrupções de alta prioridade que periodicamente interrompiam o sinal PWM por atrasarem a rotina de interrupção do timer utilizado para o sinal PWM. Para contornar esta questão foi utilizado um microcontrolador auxiliar para gerar o sinal PWM (PIC16LF1824). Este microcontrolador foi utilizado com o objectivo de receber o código RGB e actualizar o sinal PWM dos MOSFET.

6. CONCLUSÕES

Em conclusão pode afirmar-se que os desenvolvimentos feitos neste projecto superam os objectivos inicialmente propostos. A solução desenvolvida permite não só endereçar a informação a partir de um computador portátil para toda a rede, como também possui uma interface de programação dos ambientes decorativos bastante amigável para o utilizador.

Verificou-se no entanto que, por vezes, para que um dispositivo entre na rede é necessário fazer um reset ao micro. Isto acontece mesmo com a aplicação de demonstração fornecida pela Microchip[®], utilizando mesmo os hexadecimais do código criados por eles. A Microchip[®] possui uma versão da *stack* ZigBee[™], que é paga, e neste projecto foi utilizada uma versão gratuita. Esta não está tão optimizada a nível de gestão dos fluxos de informação em redes mais complexas e com um número maior de dispositivos. Será por isso importante, em futuros desenvolvimentos, adquirir a versão paga desta *stack*.

Para futuros desenvolvimentos seria de todo o interesse tornar a estrutura de *hardware* dos nós da rede mais simples, com menos componentes. Desta forma conseguir-se-ia reduzir o custo de produção de cada módulo e ao mesmo tempo tornar o sistema mais robusto e com menos estruturas de comunicação. Para dispensar a utilização de processadores auxiliares e a conseqüente necessidade de comunicação entre eles, poderá estudar-se a possibilidade de utilizar-se micros da família PIC24 ou dsPIC compatíveis com a *stack* e ao mesmo tempo

com os periféricos necessários para este projecto (comunicação USB e PWM por hardware).

A nível da aplicação RGBCastros apesar de funcional e de fácil utilização para o utilizador, ainda não possui a estrutura de gestão de dados que seria desejável. Assim, para futuros desenvolvimentos, seria interessante a criação de uma base de dados e do conceito de projecto para facilitar a gestão e armazenamento de informação, com todas as vantagens que lhe estão associadas. Isto daria ao utilizador a possibilidade de utilizar e editar projectos previamente criados e guardados por exemplo.

A nível de compatibilidades com outros sistemas, actualmente existentes no mercado, em futuros desenvolvimentos poder-se-á adicionar a cada dispositivo de comunicação da rede, uma interface de comunicação para DMX. Assim, este sistema tornar-se-ia compatível com os vários tipos de projectores de LED RGB existentes no mercado que utilizam este protocolo. O DMX é o protocolo de excelência actualmente mais utilizado na área da iluminação arquitectural.

Referências Documentais

- [1] ZigBee® Alliance - Markets & Solutions, ZigBee Alliance, ZigBee Alliance Wireless That Simply Works, visitado em 05/11/09, disponível em <http://zigbee.org/>
- [2] MICROCHIP – AN1232 Microchip ZigBee-2006 Residential Stack Protocol Microchip Technology Inc, 2008.
- [3] MICROCHIP - PICDEM™ Z Demonstration Kit User's Guide, Microchip Technology Inc, 2008
- [4] MICROCHIP – ZENA™ Wireless Network Analyzer User's Guide, Microchip Technology Inc, 2008.
- [5] As Redes com ZigBee, [eletronica.org](http://www2.eletronica.org), visitado em 04/10/09, disponível em: <http://www2.eletronica.org/artigos/eletronica-digital/as-redes-com-zigbee/>
- [6] Application Note AN654 da Microchip
- [7] MRF24J40 Datasheet – “IEEE 802.15.4 2.4 GHz RF Transceiver”.2008.
- [8] MICROCHIP – MRF24J40MA Transceiver , visitado em 02/04/10, disponível em: <http://www.microchip.com/wwwproducts/Devices.aspx?dDocName=en535967>
- [9] Primetec - Engenharia de Sistemas, Lda, visitado em 02/04/10, disponível em: <http://www.primetec.pt/>
- [10] Nicolaudie - DMX Lighting Control, visitado em 12/05/10, disponível em: <http://www.nicolaudie.com/>

Anexo A. Configurações do PIC18F2550 para comunicação USB

Neste anexo é apresentada uma tabela com a listagem de todos os registos de funções especiais, que é necessário habilitar no PIC18F2550 para configurar o módulo de comunicação USB e um diagrama com os registos de configuração do *clock*.

Tabela 1 Ajuste dos bits de configuração do PIC18LF2550

Bits de configuração no <i>microC</i>	Descrição
PLLDIV_2_1L	Oscilador Externo de 8Mz
CPUDIV_OSC1_PLL2_1L	USB full-speed 48Mz – Velocidade 2.0
USBDIV_2_1L	A fonte de clock para o USB vem do PLL 96Mz, dividido por 2.
FOSC_HSPLL_HS_1H	Utilizar o oscilador externo com o PLL activo
FCMEM_OFF_1H	Em caso de falha no oscilador externo o micro faz a comutação para o oscilador interno
BOR_ON_2L	Opção de economia de energia
BORV_43_2L	Tirar o visto deste registo para que o PIC possa funcionar a 3.3V (Vop>2,05V)
VREGEN_ON_2L	Habilitar o regulador interno de 3,3V para alimentação dos periféricos de USB
WDT_OFF_2H	Watchdog Timer habilitado
WDTPS_256_2H	Postscale (1:256)
MCLRE_OFF_3H	Registo que permite desabilitar o MCLRE
LPT1OSC_OFF_3H	Oscilador do Timer 1 configurado para o modo de alta potência (fica mais imune ao ruído)
PBADEN_OFF_3H	PORTB <04:00> pinos A/D são configurados como I/O digitais

O diagrama seguinte apresenta as configurações realizadas no PIC18F2550 para a fonte de *clock*. Foi utilizado um oscilador primário de 8Mz.

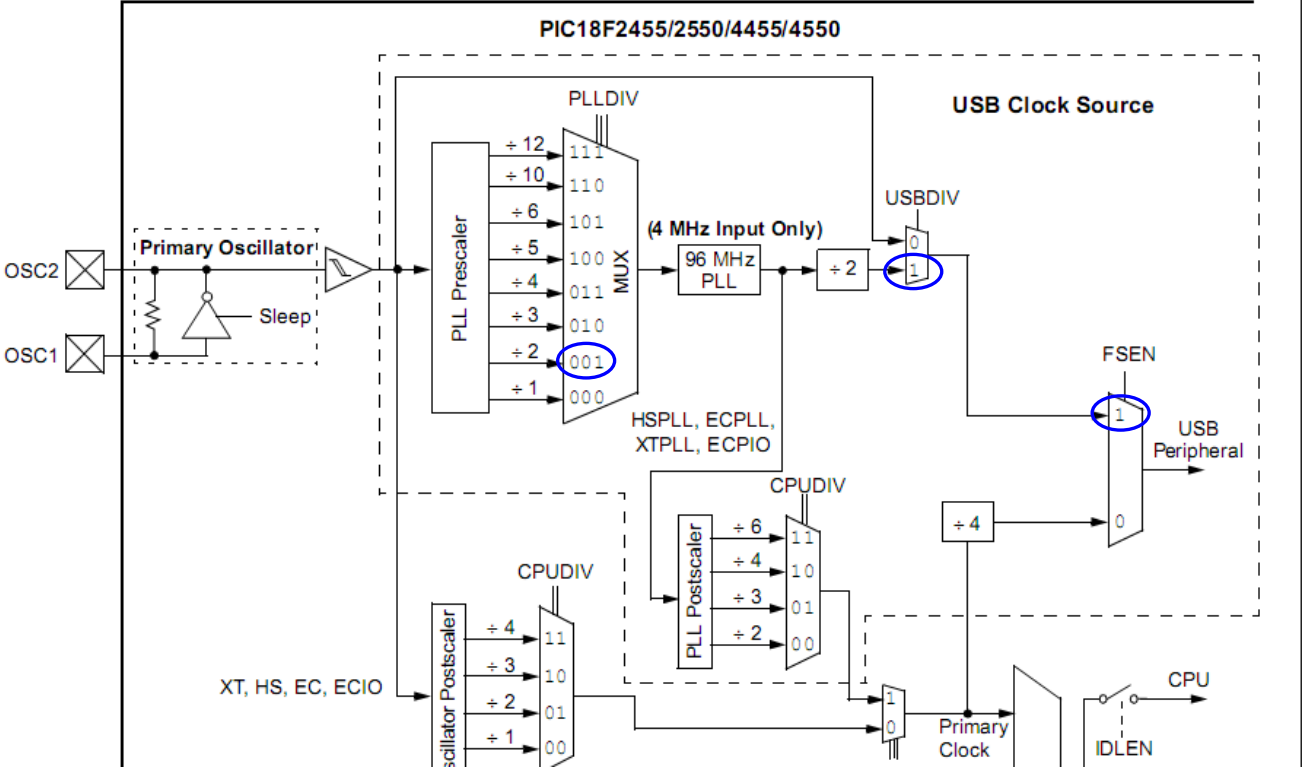
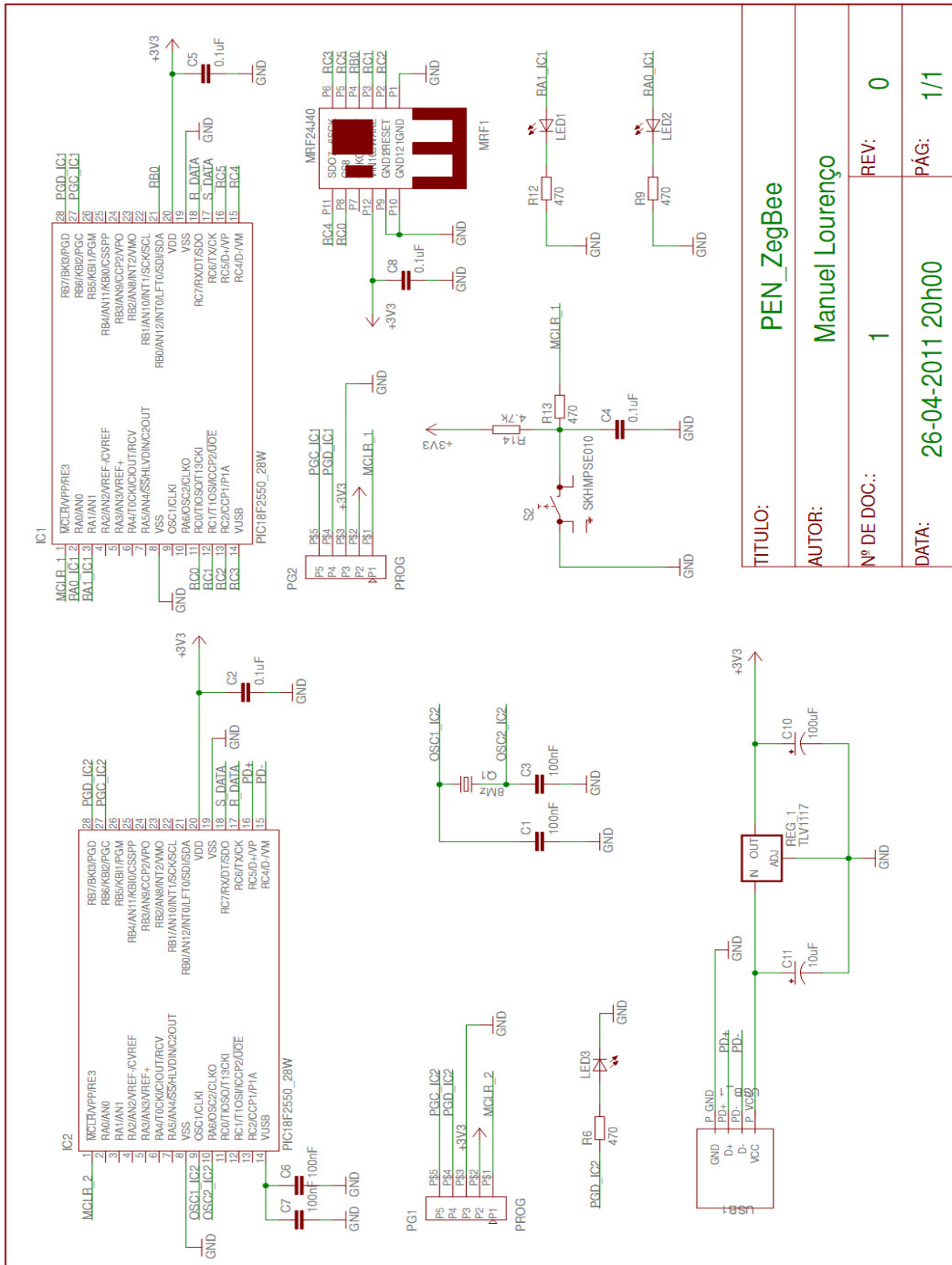


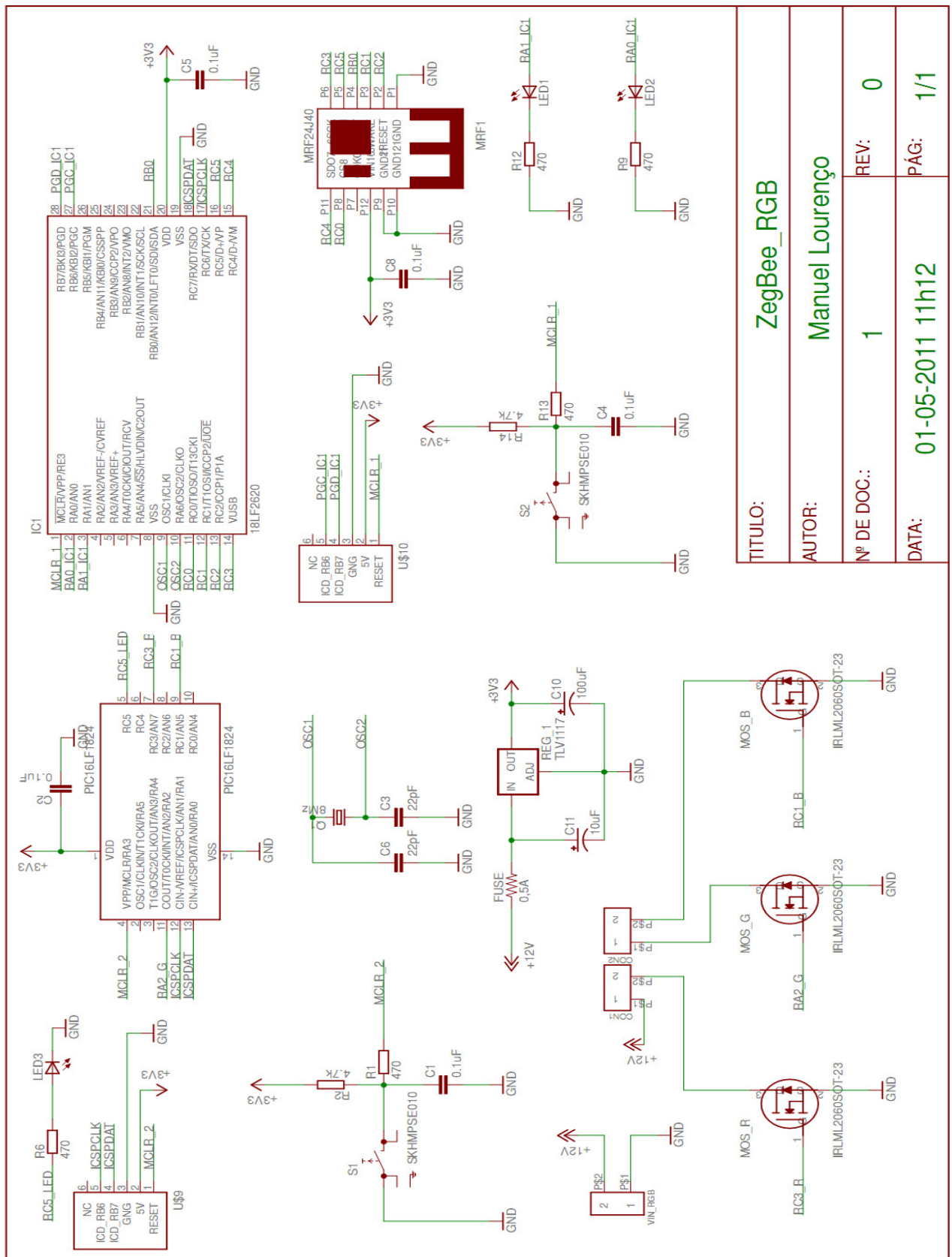
Figura 25 Diagrama de configurações da fonte de *clock* utilizada na comunicação USB

Anexo B. Esquemáticos das PCBs

1 - Esquemático, feito no eagle™, do hardware desenvolvido para o dispositivo ZigBee™ que vai permitir a comunicação com o computador.



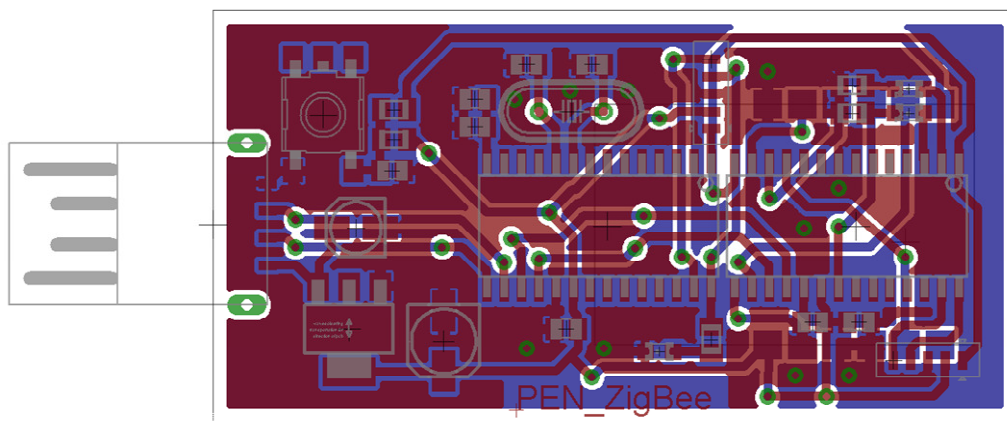
2 - Esquemático, feito no eagle™, do hardware desenvolvido para o dispositivo ZigBee™ que vai permitir o controlo da cor RGB do material iluminativo.



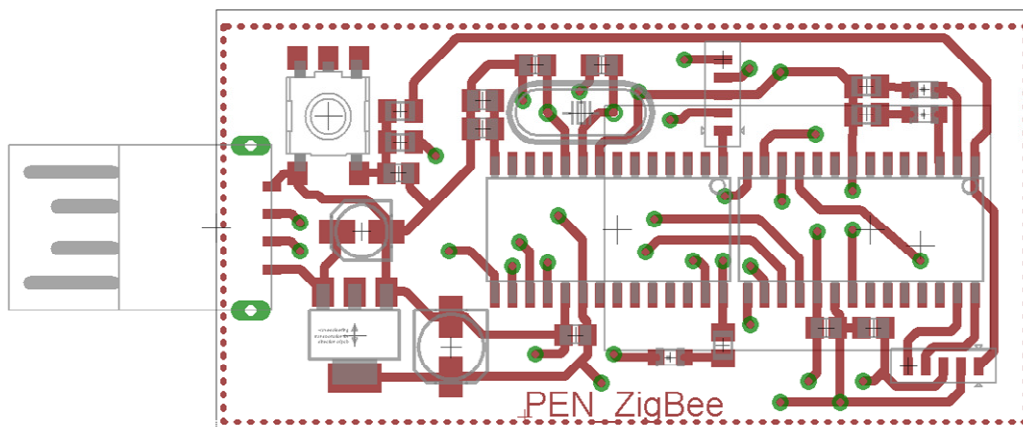
TITULO:	ZegBee_RGB
AUTOR:	Manuel Lourenço
Nº DE DOC.:	1
DATA:	01-05-2011 11h12
REV:	0
PÁG:	1/1

Anexo C. Layout das placas de Circuito Impresso

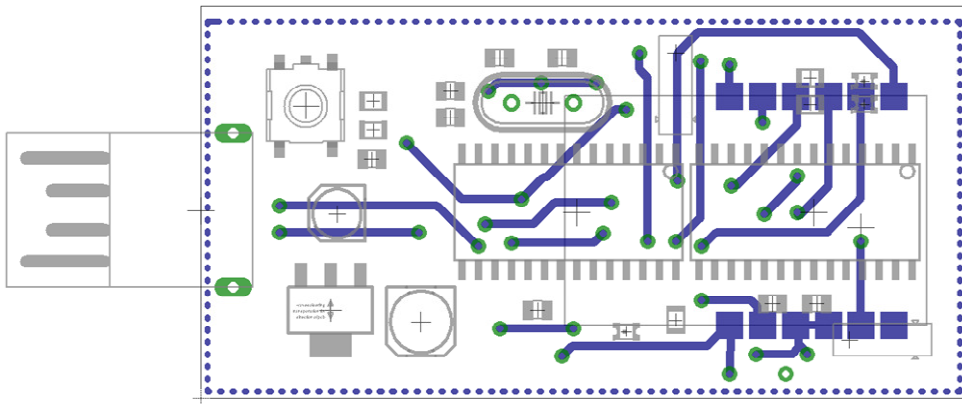
1 - Layout da placa de Circuito Impresso do hardware desenvolvido para o dispositivo ZigBee™ que vai permitir a comunicação com o computador. Como se trata de um circuito sujeito à captação de ruído, foi colocada uma malha de terra, ligada ao GND, cobrindo as duas faces da placa.



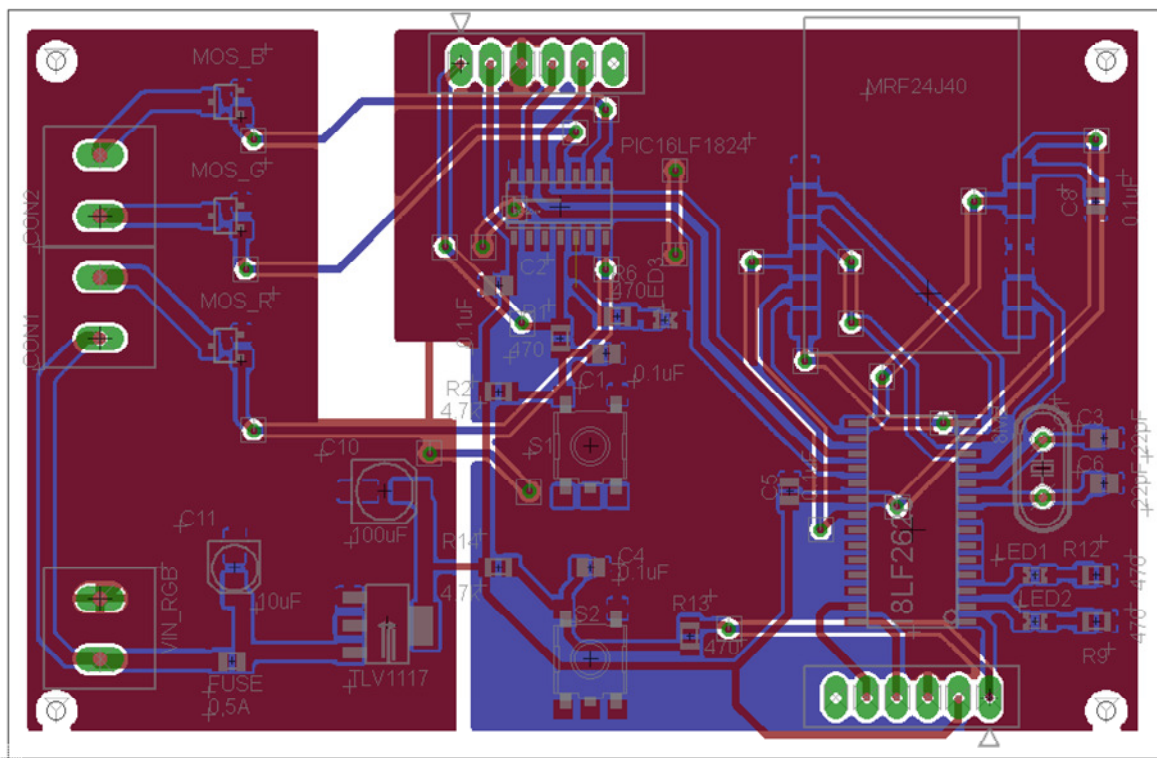
1.1. Vista da face superior



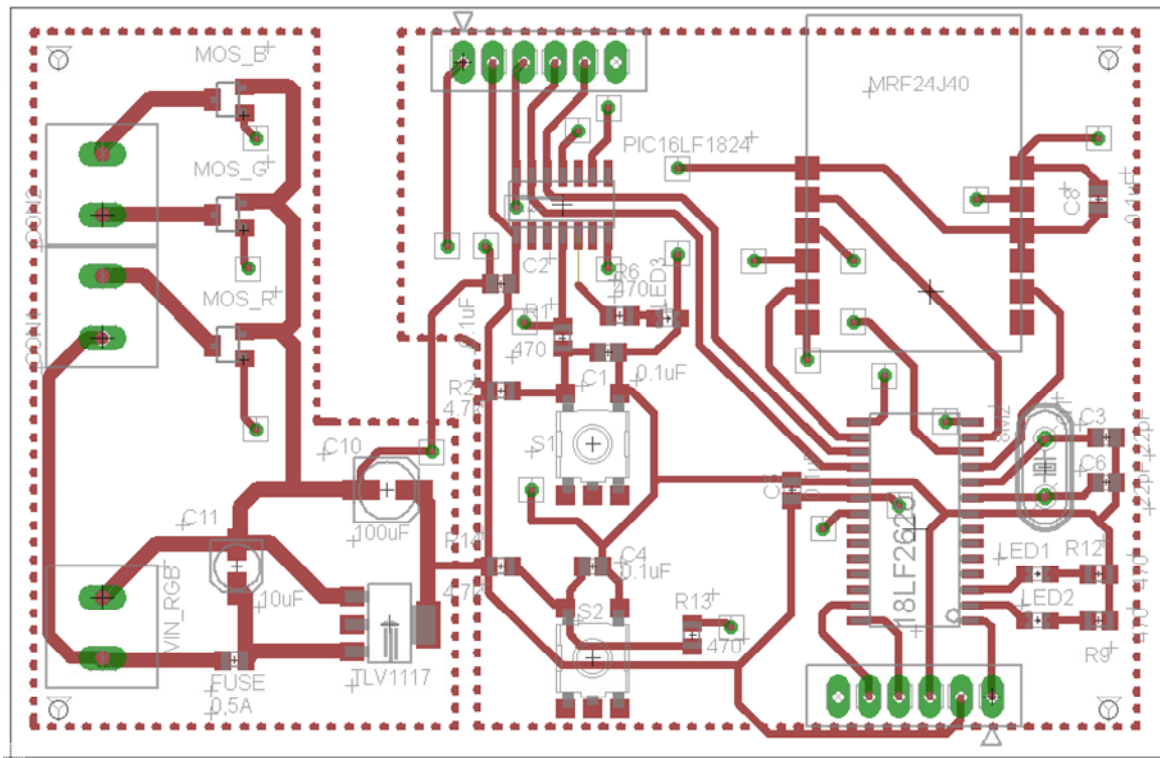
1.2. Vista da face Inferior



2 - Layout da placa de Circuito Impresso do hardware desenvolvido para o dispositivo ZigBee™ que vai permitir o controlo da cor RGB do material Iluminativo. Como se trata de um circuito sujeito à captação de ruído, foi colocada uma malha de terra, ligada ao GND, cobrindo as duas faces da placa. Foi também feita uma separação dos componentes de hardware de potência dos de sinal.



2.1. Vista da face superior



2.2. Vista da face Inferior

