



Mestrado em Engenharia Eletrotécnica

Comunicação sem fios DMX512

Dissertação apresentada para a obtenção do grau de Mestre em Engenharia Eletrotécnica – Área de Especialização em Automação e Comunicações em Sistemas Industriais

Autor

Tiago Manuel de Oliveira Crespo

Orientadores

Doutor Frederico Miguel do Céu Marques dos Santos

Professor do Departamento de Engenharia Eletrotécnica
Instituto Superior de Engenharia de Coimbra

Instituição

Instituto Superior de Engenharia de Coimbra

Coimbra, maio, 2018

AGRADECIMENTOS

A aprendizagem durante o meu tempo no desenvolvimento do projeto Comunicação Sem Fios DMX-512 foi inigualável, uma vez que experienciei uma enorme preparação e conhecimento do protocolo de comunicação em estudo, que o mundo industrial e/ou empresarial exige em termos de sabedoria, gestão e organização.

Em primeiro lugar, agradeço ao Professor Doutor Frederico Miguel do Céu Marques dos Santos, Professor do Departamento de Engenharia Eletrotécnica, pela ótima orientação ao longo do desenvolvimento do presente projeto, mostrando-se compreensivo, paciente, disponível e solícito. Todas as correções e sugestões foram fundamentais para a conclusão do mesmo.

Ao Instituto Superior de Engenharia de Coimbra e aos seus professores pelos conhecimentos transmitidos e aprendizagens proporcionadas.

Ainda aos meus amigos pessoais e colegas de Engenharia, em especial ao colega e amigo Eng.º Bruno Vieira pelo apoio fornecido e pelo acompanhamento constante durante o desenvolvimento deste projeto.

Aos meus pais e irmã pelo inestimável apoio recebido ao longo deste ano, pelos valores transmitidos, compreensão e fonte de motivação constante.

Por último agradeço, ao Sr. João Paulo, superior hierárquico na empresa onde atualmente exerço funções na área de automação industrial e robótica.

“O segredo da felicidade no trabalho está contido numa só palavra: excelência. Saber como fazer bem alguma coisa é desfrutá-la”. *Pearl Buck*

A todos, Muito Obrigado

RESUMO

Este relatório tem como objetivo descrever o projeto de investigação realizado no âmbito da conclusão do Mestrado em Engenharia Eletrotécnica, na área da Automação e Comunicações em Sistemas Industriais, ministrado no Instituto Superior de Engenharia de Coimbra, pertencente ao Instituto Politécnico de Coimbra.

O presente projeto de investigação intitulado “*Comunicação Sem Fios DMX512*” pretende essencialmente efetuar o estudo e análise da melhor metodologia a utilizar na construção de uma rede sem fios para implementação do protocolo de comunicação DMX512. Atendendo a este objetivo principal, torna-se premente apurar as possíveis configurações deste tipo de comunicação, as diretrizes a seguir nos diferentes métodos, a avaliação da segurança e fiabilidade dos mesmos e, por último a otimização da melhor solução encontrada. Pretende-se, assim, analisar e comparar as melhores topologias que se adaptam ao meio, bem como seleccionar o protocolo e tecnologias a utilizar.

Face ao exposto, o presente projeto incide sobre a construção de três protótipos de sistema de Comunicação Sem Fios, com a finalidade de controlar equipamentos de iluminação cénica utilizados em eventos de diferente natureza.

Neste documento encontram-se descritos os principais pressupostos teóricos que fundamentam as diversas tecnologias utilizadas nos projetos descritos, assim como explicações aprofundadas das diversas etapas associadas ao desenvolvimento dos respetivos projetos. Ou seja, serão apresentados conceitos teóricos e tecnologias utilizadas, destacando-se: DMX512, RDM, Art-Net, Radiofrequência, *Ethernet* e o *hardware e software* em sistemas embebidos.

De seguida, apresenta-se a construção de dois protótipos sem fios usando módulos RF para a conversão e transmissão dos sinais do protocolo DMX512 que deverão atender com êxito à proposta inicial, isto é, eliminar a necessidade do uso de cabos entre os equipamentos envolvidos, de forma segura e com baixo custo de desenvolvimento. Uma vez que é proposto, na primeira fase do projeto, a conceção de dois controladores sem fios principais que irão ser responsáveis pela emissão e receção dos dados DMX, entre a zona técnica e a zona cénica. Numa segunda fase, procede-se à conversão do elemento recetor, num elemento de retransmissão sem fios de forma a estabelecer a ponte de comunicação com os módulos de receção terminais, dedicados a cada dispositivo.

Por último, será apresentado, a título exemplificativo um dos *softwares* de programação de iluminação profissional utilizado, apresentando-se também algumas das ferramentas auxiliares de trabalho.

Palavras-chave: DMX512; Comunicação Sem fios; Módulos RF; Controlo, Sistemas Embebidos.

ABSTRACT

This report is intended to describe the research project made to conclude the Master's degree in Electrical Engineering, in Automation and Communications in Industrial Systems, taught at the Coimbra Institute of Engineering, a School belonging to the Polytechnic Institute of Coimbra.

The present research project entitled "**DMX512 Wireless Communication**" essentially intends in the study and analysis of the best methodology in the construction of a wireless network, to implement the DMX512 communication protocol. According to the main objective, it's necessary to predict the possible configurations of this type of communication, such as the guidelines to follow, the security and reliability assessment, and finally the optimization of the best solution found.

Therefore, it's intended to analyze and compare the best topologies that adjust to the environment, as well as the selection of the protocol and technologies to use.

Considering the above, this project focuses on the construction of three prototypes of Wireless Communication Systems, to control stage lighting equipment used in events of different nature.

In this document, the main theoretical basis supporting the various technologies that will be used in the developed projects will be described, as well as the detailed explanations for the various phases associated on the development of each of the projects. In other words, theoretical concepts and technologies will be presented, such as: DMX512, RDM, Art-Net, Radiofrequency, Ethernet and hardware and software in embedded systems.

Following, is presented the construction of two wireless prototypes using RF modules for the conversion and transmission of the DMX512 protocol signals that must successfully meet the initial proposal, that consists in eliminate the need to use cables between the equipment's involved, in a safe way and with low development cost. Since it is proposed, the first phase of the project, consists in the design of two main wireless controllers that will be responsible for the emission and reception of the DMX data between the technical zone and the scenic zone. In a second phase, the receiver element will be converted into a wireless transceiver element to establish a bridge with the dedicated reception terminal modules.

Finally, it will be presented, the professional lighting programming software used, as well as some of the auxiliary work tools.

Key-words: DMX512; Wireless Communications; RF Module; Control, Embedded Systems.

ÍNDICE

AGRADECIMENTOS.....	iii
RESUMO	v
ABSTRACT	vii
ÍNDICE	ix
ÍNDICE DE FIGURAS	xiii
ÍNDICE DE TABELAS	xvii
ÍNDICE DE EQUAÇÕES.....	xix
SIMBOLOGIA	xxi
ABREVIATURAS.....	xxiii
1 INTRODUÇÃO	1
1.1 Considerações gerais.....	2
1.2 Objetivos	3
1.2.1 Problemática.....	3
1.2.2 Justificativa.....	3
1.2.3 Identificação de objetivos específicos	4
2 CONCEITOS TEÓRICOS E TECNOLOGIAS UTILIZADAS.....	7
2.1 DMX512	7
2.1.1 História.....	7
2.1.2 RS-485	8
2.1.3 Topologia da rede DMX	10
2.1.4 Cabos e fichas de comunicação	13
2.1.5 Características Elétricas	15
2.1.6 Pacote DMX.....	16
2.1.7 Estado da arte	19
2.2 RDM	20
2.2.1 Definição	20
2.2.2 História.....	21
2.2.3 Funcionamento	21
2.2.4 Topologia da rede	23
2.2.5 Estado da arte	23

2.3	Art-Net	24
2.3.1	Definição	24
2.3.2	História.....	24
2.3.3	Pacote de dados	25
2.3.4	Estado da arte	26
2.4	Rádio Frequência.....	26
2.4.1	Definição	27
2.4.2	Estado da arte	27
2.5	Ethernet	28
2.5.1	<i>Internet Protocol (IP)</i>	29
2.5.2	TCP/UDP	29
2.5.3	Wi-Fi.....	29
2.6	SPI.....	30
2.7	UART.....	31
2.8	Sumário	32
3	HARDWARE E SOFTWARE EM SISTEMAS EMBEBIDOS	33
3.1	Sistemas embebidos	33
3.2	Microcontroladores	34
3.3	Módulo de rádio frequência nRF24L01	36
3.3.1	Propriedades elétricas e características do módulo	37
3.3.2	Estado da arte	38
3.4	Módulos de RS-485	40
3.5	<i>Human Machine Interface (HMI)</i>	42
3.6	Desenvolvimento de placas de circuito impresso.....	43
3.7	Programação dos microcontroladores.....	45
3.8	Programação do HMI.....	46
3.9	Sumário	47
4	DESENVOLVIMENTO	49
4.1	Fundamento da seleção da tecnologia e metodologia adotadas	49
4.1.1	<i>Wi-Fi</i>	49
4.1.2	Bluetooth / BLE4.0.....	50
4.1.3	Zigbee	51
4.1.4	Outros <i>Transceivers</i> Rádio Frequência.....	52

4.1.5	Módulo selecionado para a elaboração do projeto: nRF24L01	52
4.2	Controlador DMX sem fios principal (Régie) – 1ª fase.....	54
4.2.1	Diagrama geral de funcionamento.....	54
4.2.2	<i>Hardware do controlador da transmissão DMX</i>	55
4.2.3	Programação desenvolvida.....	58
4.3	Retransmissor DMX sem fios principal (<i>Stage</i>) – 1ª fase.....	65
4.3.1	Diagrama geral de funcionamento.....	65
4.3.2	<i>Alterações ou hardware do controlador de retransmissão DMX (Stage)</i>	66
4.3.3	Programação desenvolvida.....	66
4.4	Controlador recetores sem fios individuais DMX512 – 2ªfase	68
4.4.1	<i>Diagrama geral de funcionamento</i>	68
4.4.2	<i>Hardware do controlador de receção DMX Terminal</i>	68
4.4.3	MAX 485	70
4.4.4	Placa desenvolvida.....	70
4.4.5	Programação desenvolvida.....	71
4.5	Sumário	72
5	SOFTWARE DE DESENVOLVIMENTO E CONTROLO DMX	73
5.1	Notas introdutórias.....	73
5.1.1	<i>Program Mode</i>	73
5.1.2	<i>Live Mode</i>	74
5.1.3	<i>Time Code</i>	74
5.2	<i>Software de programação e controlo</i>	74
5.2.1	Hardware e software Avolites	74
5.2.2	<i>Softwares para Computadores</i>	76
5.3	Ferramentas auxiliares de trabalho	77
5.3.1	Avolites Titan Simulater	77
5.3.2	Avolites Titan Visualiser	77
5.3.3	Avolites Titan Remote	77
5.3.4	Avolites Personality Builder	78
5.4	Sumário	78
6	CONCLUSÕES	79
6.1	Conclusões relativas ao desenvolvimento dos projetos	79
6.2	Trabalhos futuros.....	80

7	BIBLIOGRAFIA	83
	Anexo I - Programa do controlador DMX Sem Fios Principal	91
	Anexo II - Esquema elétrico do controlador DMX sem fios principal	95
	Anexo III - Programa do controlador DMX sem fios Secundário	97
	Anexo IV - Programa do controlador de DMX sem fios terminal	101
	Anexo V - Esquema elétrico do controlador DMX sem fios terminal	105
	Anexo VI - Programa do mapeamento controlador de DMX sem fios terminal.....	107
	Anexo VII - Esquema elétrico do controlador de DMX sem fios terminal	117
	Anexo VIII - <i>Software</i> de Simulação e visualização	123
	<i>WyziWyg Lighting Design Software</i>	125

ÍNDICE DE FIGURAS

Figura 1.1 - Concerto Vangelis em Doha realizado com DMX512 sem fios pela empresa W-DMX [2].	2
Figura 1.2 - Exemplo de uma possível solução DMX512 sem fios [3].	4
Figura 1.3 - Diagrama de configuração da comunicação ponto a ponto (1ª fase).	5
Figura 1.4 - Diagrama de configuração final da comunicação do projeto de investigação (2ª fase).	5
Figura 1.5 - Exemplo de duas topologias de dispositivos comerciais [4] [5].	6
Figura 2.1 - Exemplo topologia de rede DMX512 [13].	10
Figura 2.2 - Exemplo de controladores DMX512.	10
Figura 2.3 - Exemplo de inúmeros dispositivos recetores ou <i>fixtures</i> .	12
Figura 2.4 - Exemplo de elementos terminais de comunicação DMX512 [17] [18].	12
Figura 2.5 - Exemplo de um cabo regulamentado de comunicação DMX512A [19].	13
Figura 2.6 - Exemplo de um <i>splitter</i> DMX512 [21].	13
Figura 2.7 - Exemplo de fichas e tomadas DMX512 conforme normas do protocolo DMX512A [22].	14
Figura 2.8 - Circuito transmissor do controlador (esquerda) e circuito do recetor (direita) [22].	15
Figura 2.9 – Exemplo da variação do par diferencial [23].	16
Figura 2.10 - Pacote de dados do protocolo DMX512 [24].	17
Figura 2.11 - Diagrama estrutural de um pacote de dados DMX [25].	17
Figura 2.12 - Diagrama geral do espectro eletromagnético.	26
Figura 2.13 - Relação do protocolo <i>Ethernet</i> com o modelo OSI. Adaptado de [35].	28
Figura 2.14 - Formato do pacote <i>Ethernet</i> na camada de ligação.	29
Figura 2.15 - Exemplo da topologia e comunicação da rede SPI. Adaptado de [39] e [40].	31
Figura 2.16 - Exemplo de um pacote de dados série. Adaptado de [42].	31
Figura 2.17 - Conversor UART – USB.	32
Figura 3.1 - Diagrama de blocos do sistema embebido a desenvolver.	33
Figura 3.2 - Placa de desenvolvimento Arduino Uno. Adaptado de [45].	35
Figura 3.3 - Placa de desenvolvimento Arduino Pro Mini [45].	35
Figura 3.4 - Módulo nRF24L01 e respetivo diagrama de blocos [47].	36

Figura 3.5 - Módulo nRF24L01 e respetiva legenda da identificação de pinos. Adaptado de [48]	37
Figura 3.6 - Módulo nRF24L01+ e respetiva antena [49].....	39
Figura 3.7 - Módulo RS-485 da LC Technology [51].	40
Figura 3.8 - Esquema de ligação do módulo RS485 em modo emissor [52].	41
Figura 3.9 - Esquema de ligação do módulo RS-485 em modo recetor [52].	41
Figura 3.10 - HMI Nextion 7 polegadas [53].	42
Figura 3.11 - Esquema Eagle da placa PCB desenvolvida para o protótipo DMX sem fios. ...	44
Figura 3.12 - IDE Arduíno.	46
Figura 3.13 - Ambiente Nextion Editor.	47
Figura 4.1 - Módulo Wi-Fi ESP8266 [59].....	49
Figura 4.2 - Módulo Bluetooth V4.0 nRF51822 BLE [62].....	51
Figura 4.3 - Módulos ZigBee [64].	51
Figura 4.4 - Módulo MRF24J40MA. Adaptado de [65].	52
Figura 4.5 - Diagrama de interfaces do controlador sem fios DMX principal.....	55
Figura 4.6 - Imagem real do primeiro layout desenvolvido.	55
Figura 4.7 - Imagem real do primeiro protótipo desenvolvido.....	56
Figura 4.8 - Esquema de ligações do controlador desenvolvido.	56
Figura 4.9 - Esquema de ligações do controlador principal.	57
Figura 4.10 - Esquema de ligações Eagle do controlador principal.	58
Figura 4.11 - Diagrama de funcionamento do controlador sem fios DMX512.....	59
Figura 4.12 - Ambiente gráfico HMI.	62
Figura 4.13 – Janela de confirmação da realização de mapeamento	62
Figura 4.14 - Janela de realização de mapeamento das máquinas a controlar.	63
Figura 4.15 – Inserção do endereço DMX.	64
Figura 4.16 - Diagrama do recetor sem fios DMX512.	65
Figura 4.17 - Controlador recetor sem fios DMX512.....	66
Figura 4.18 - Diagrama de funcionamento do recetor sem fios DMX512 principal.	67
Figura 4.19 - Diagrama do controlador recetor terminal sem fios DMX.....	68
Figura 4.20 – Protótipo controlador terminal desenvolvido.....	69
Figura 4.21 - Mapa de pinos do ATMega328P-AU [66].	69
Figura 4.22 – Esquema equivalente de ligação do MAX485 [68].	70

Figura 4.23 - Desenho da PCI do controlador DMX sem fios terminal e respetivo resultado final.	70
Figura 4.25 – Etapas de desenvolvimento do controlador desenvolvido.....	71
Figura 4.26 - Controlador DMX sem fios terminal.	71
Figura 4.27 - Diagrama de funcionamento do controlador DMX sem fios terminal.....	72
Figura 5.1 – <i>Dongle</i> Avolites Titan One [69].....	76
Figura 7.1 - Diagrama de criação de <i>Layout</i> do evento e respetivo cenário [71]......	124
Figura 7.2 - Diagrama de Simulação e Visualização [71]......	124
Figura 7.3 - Logotipo do <i>software</i> Wysiwyg Lighting Design [72].	125
Figura 7.4 - Simulação Wysiwyg (esquerda) e evento real (direita) de Karel Gott tour 2014 [73].	125
Figura 7.5 - Imagem da simulação <i>Wysiwyg</i> do evento <i>Swedish Melodifestivalen</i> [74].	126

ÍNDICE DE TABELAS

Tabela 2.1 - Características elétricas do padrão RS-485 [12].	9
Tabela 2.2 - Número de canais conforme universo DMX.	11
Tabela 2.3 - Esquema de ligação das fichas XLR-5 recomendada pela norma [9].	14
Tabela 2.4 - Esquema de ligação das fichas RJ-45 recomendada pela norma [9].	14
Tabela 2.5 - Resumo dos limites de parâmetros de temporização do protocolo DMX512A. ...	19
Tabela 2.6 – Formato do pacote de dados Art-Net DMX [29].	25
Tabela 2.7 – Algumas das bandas ISM mais usuais segundo a ANACOM [32].	27
Tabela 2.8 - Protocolos inerentes à tecnologia Wi-Fi.	30
Tabela 3.1 - Especificações do microcontrolador ATmega328P [46].	36
Tabela 3.2 - Formato do pacote de dados <i>Enhanced Shockburst</i> [47].	37
Tabela 3.3 - Especificações do módulo radiofrequência nRF24L01 [47].	38
Tabela 3.4 - Especificações do módulo de radiofrequência nRF24L01+ [49].	40
Tabela 3.5 - Especificações do módulo RS-485 [50].	41
Tabela 3.6 - Especificações do HMI [53].	43
Tabela 4.1 - Tabela de comparação das diferentes tecnologias.	53
Tabela 4.2 - Lista de canais do padrão 802.11	60
Tabela 5.1 – Principais características de funcionamento do <i>software</i> Avolites Titan [69]. ...	75

ÍNDICE DE EQUAÇÕES

Equação 2.1 - Taxa mínima de transmissão do protocolo RS-485.....	16
Equação 4.1-Equação de cálculo da frequência do canal RF.....	60

SIMBOLOGIA

ampère (símbolo: **A**) - É a unidade de corrente elétrica do Sistema Internacional de Unidades (SI), que representa a intensidade de corrente elétrica.

baud (símbolo: **Baud**) - É a unidade de velocidade de sinalização e representa o número de mudanças na linha de transmissão (seja em frequência, amplitude, fase etc...) por segundo.

bits por segundo (símbolo: **bps**) - É uma unidade de transmissão de dados e representa o número de bits transmitidos por segundo.

decibel (símbolo: **dB**) - É uma unidade logarítmica que indica a proporção de uma quantidade física (geralmente energia ou intensidade) em relação a um nível de referência implícito.

hertz (símbolo: **Hz**) - É a unidade de frequência derivada do Sistema Internacional de Unidades (SI), para a frequência. É expressa em ciclos por segundo, a frequência de um evento periódico.

metro (símbolo: **m**) - É a unidade de medida derivada do Sistema Internacional de Unidades (SI), para medir comprimentos.

ohm (símbolo **Ω**) - É a unidade de resistência elétrica do Sistema Internacional de Unidades.

segundo (símbolo: **s**) - É a unidade de frequência derivada do Sistema Internacional de Unidades (SI), para medir intervalos de tempo.

volt (símbolo: **V**) - É a unidade de tensão elétrica do Sistema Internacional de Unidades (SI), que representa a diferença de potencial elétrico entre dois pontos.

watt (símbolo: **W**) - É a unidade de potência do Sistema Internacional de Unidades, é equivalente a um joule por segundo.

ABREVIATURAS

ADC - *Analog-to-Digital Converter*

AMX 192 - *Analog MultipleX*

ANACOM - *Autoridade Nacional de Comunicações*

ANSI - *American National Standards Institute*

ARM - *Advanced RISC Machine*

ASCII - *American Standard Code for Information Interchange*

CAD - *Computer-aided design*

CC – *Corrente Contínua*

CD – *Channel Data*

CI – *Circuito Integrado*

CRC - *Cyclic Redundancy Check*

CSMA/CD - *Carrier Sense Multiple Access with Collision Detection*

DC - *Direct Current*

DCF - *Distributed Coordination Function*

DHCP - *Dynamic Host Configuration Protocol*

DMX-512 - *Digital MultipleX*

DSSS - *Direct Sequence Spread Spectrum*

EEPROM - *Electrically-Erasable Programmable Read-Only Memory*

EIA - *Electronics Industry Association*

ESTA - *Entertainment Services and Technology Association*

ETSI - *European Telecommunications Standards Institute;*

FHSS - *Frequency-Hopping Spread Spectrum*

FM - *Frequency Modulation*

GFSK - *Gaussian Frequency-Shift Keying*

GUI - *Graphical User Interface*

GSM - *Global System for Mobile Communications*

HMI - *Human–Machine Interface*

I/O – *Input/Output*

I²C - *Inter-Integrated Circuit*

IDE - *Integrated Development Environment*

IEEE - *Institute of Electrical and Electronics Engineers*

IP - *Internet Protocol*

ISM - *Industrial, Scientific and Medical*

LAN – *Local Area Network*

LED - *Light-Emitting Diode*

LTE - *Long Term Evolution*

MAB - *Mark After Break*

MAC - *Media Access Control*

MISO - *Master In-Slave Out*

MOSI – *Master Out-Slave In*

MTBF - *Mark Time Between Frames*

MTBP - *Mark Time Between Packets*

PSK - *Phase-Shift Keying*

OFDM - *Orthogonal Frequency Division Multiplexing*

OEM – *Original Equipment Manufacturer*

OSI - *Open Systems Interconnection*

PCI – *Placa de Circuito Impressa*

PLC - *Programmable Logic Controller*

PSRR - *Power Supply Rejection Rádio*

PWM - *Pulse-Width Modulation*

RF – Rádio Frequência

RGB – *Red Green Blue*

RISC - *Reduced Instruction Set Computer;*

RS - *Recommended Standard*

sACN - *Streaming Architecture for
Control Networks*

SC – *Start Code*

SCLK - *Serial Clock*

SIP - *Session Initiation Protocol*

SS – *Slave Select*

WPAN - *Wireless Personal Area Network*

WLAN – *Wireless Local Area Network*

1 INTRODUÇÃO

Nos últimos anos, o método de comunicação sem fios, conhecido como *wireless*, encontra-se presente em diversas áreas tecnológicas e com o decorrer do tempo e os avanços da tecnologia, tornou-se fundamental para a mobilidade das pessoas e empresas.

O setor do entretenimento e televisão foi um dos primeiros a seguir esta tendência, consequência do elevado espólio financeiro e do impacto a vários níveis que estes sistemas podem apresentar em todos os aspetos (despesas, logística, capacidade, etc.).

A forte demanda e procura de sistemas específicos e sistemas integrados, levou a um crescimento exponencial do número de empresas a optarem pelo desenvolvimento de novos produtos e aplicações disponibilizando esta tecnologia para todos os tipos de mercado.

O meio audiovisual, não foi exceção e surgiu uma linha de dispositivos, equipamentos e aplicações, envolvendo sistemas FM (*Frequency Modulation*) como o *Bluetooth* até protocolos mais sofisticados, como por exemplo a família 802.11x (sistemas *Wi-Fi*). A iluminação profissional constitui um dos principais fatores de criação de ambientes cénicos diferenciados. Através de diversos recursos é possível recriar diferentes situações, ambientes e efeitos com o intuito de transmitir sensações aos espetadores ou utilizadores de um determinado espaço físico.

O setor audiovisual tem-se caracterizado como uma das áreas da indústria com um enorme recurso financeiro, efeito resultante de dimensão, criado pela existência e exigência de um grande mercado, oferecendo aos fabricantes de produtos audiovisuais possibilidades bem mais extensas do que as atuais. O crescimento da indústria audiovisual fomenta o investimento em novas tecnologias e a profissionalização do setor, tornando este uma fonte importante na criação de empregos diretos e indiretos. Até mesmo os pequenos eventos (reuniões empresariais, congressos, simpósios, concertos etc.) são responsáveis pela expansão do setor, onde se observa, cada vez mais, a promoção de novas tecnologias e o uso de novos métodos de trabalho que melhoram a qualidade do serviço e permitem uma redução substancial do tempo alocado ao desenvolvimento do mesmo. O setor audiovisual, na Europa, encontra-se em crescimento, o número de empresas ativas aumentou 22% (quase 100.000 empresas), entre 2008 e 2012, o que consequentemente impulsionou o aumento do número de trabalhadores no setor, estimado entre 739 mil em 2008 e 811 mil em 2013. Este fenómeno caracteriza-se principalmente pelo aumento do setor de produção, uma vez que em muitos países, o setor é caracterizado por micro e pequenas empresas de prestação de serviços que, não raras vezes, são formadas por um número inferior a 20 pessoas. O número reduzido de colaboradores, aliado aos requisitos de qualidade e segurança, implicam diretamente, um alto nível de especialização dos profissionais e forçam a evolução tecnológica dos recursos envolvidos [1]. Face ao exposto, este projeto enfoca a criação de um sistema de comunicação sem fios para efetuar o controlo dos equipamentos DMX512 de um modelo de iluminação cénica utilizado em eventos.

A sua conceção tem como objetivo eliminar a passagem de cabos do barramento DMX512, responsável por efetuar a comunicação entre a mesa de luz ou dispositivo controlador até às máquinas ou *robots* envolvidos na atividade, propiciando uma redução do custo de construção e/ou aquisição.

Existem atualmente no mercado equipamentos sem fios, que funcionam segundo o padrão 802.11b (11 Mbps), ou até mesmo com módulos Rádio Freqüência (RF) sem protocolo específico, que transmitem sinais do protocolo DMX512. Estes equipamentos são pouco utilizados, pois são produtos recentes: surgiram há menos de cinco anos e possuem um alto custo de aquisição. Existe também, um crescimento exponencial de cópias destes equipamentos que possuem reduzida qualidade ou fiabilidade.

Da análise de todos estes sistemas verificou-se a necessidade de desenvolvimento do presente projeto: efetuar um sistema de comunicação via RF para implementar o controlo de todo o sistema de iluminação presente, garantindo desta forma, a supressão de toda a cablagem de controlo. Algumas empresas do setor audiovisual, especializaram-se no desenvolvimento e utilização deste tipo de equipamentos, e muitas delas apresentam resultados bastante satisfatórios através da utilização de equipamentos de grande qualidade (ver Figura 1.1).



Figura 1.1 - Concerto Vangelis em Doha realizado com DMX512 sem fios pela empresa W-DMX [2].

1.1 Considerações gerais

O presente projeto visa a obtenção do grau de Mestre em Engenharia Eletrotécnica - Área de especialização em Automação e Comunicações em Sistemas Industriais. Este assenta sobre o desenvolvimento e construção de um protótipo de sistema de comunicação sem fios, com a finalidade de controlar equipamentos de iluminação cénica utilizados em eventos de diferentes naturezas. Inicialmente, apresenta-se uma revisão da literatura em diversas áreas como eletrónica analógica e digital, sinais e sistemas, microcontroladores e ferramentas de gestão. Na fase correspondente ao desenvolvimento construíram-se dois protótipos usando módulos *RS-485* e *RF* para conversão e transmissão dos sinais do protocolo *DMX512*. Estes atenderam com êxito a proposta inicial, uma vez que eliminam a necessidade do uso de cabos entre os equipamentos envolvidos, de forma segura, fiável e com custos reduzidos. Ainda na fase de desenvolvimento foram criados novos protótipos semelhantes, mas apenas direcionados à receção da comunicação.

1.2 Objetivos

O principal objetivo deste projeto de investigação consiste no desenvolvimento de interfaces de comunicação sem fios, de baixo custo, que permitam a transmissão e receção segura e fiável de sinais contendo informações do protocolo DMX512, eliminando por completo a necessidade de utilização de cablagem de comunicação.

De seguida enumeram-se os objetivos gerais do presente projeto:

- ✓ Eliminar a necessidade de cablagem de comunicação;
- ✓ Estudar as definições do protocolo DMX512;
- ✓ Pesquisar, analisar e selecionar as melhores metodologias para a implementação e compreensão das suas configurações;
- ✓ Conceber uma interface de comunicação sem fios funcional, exequível e segura;
- ✓ Estudar as técnicas de processamento de dados mais adequadas para o envio e receção do sinal DMX512;
- ✓ Desenvolver protótipos para a transmissão e receção de sinais DMX512;
- ✓ Efetuar ensaios para verificar a viabilidade técnica do projeto;
- ✓ Estudar a atitude da comunicação com crescimento exponencial de equipamentos;
- ✓ Conceber um protótipo de comunicação sem fios com maior alcance, de modo a garantir a comunicação entre a zona técnica e o palco;
- ✓ Otimizar os métodos de transmissão a utilizar nas máquinas de modo a minimizar a latência nestes equipamentos.

1.2.1 Problemática

Este projeto surgiu da necessidade de eliminar cabos de interligação entre os equipamentos de iluminação DMX512 e o controlador da rede DMX512 utilizados em salas de espetáculo, concertos ao vivo, ou qualquer outro tipo de evento realizado em ambientes interiores ou exteriores, mantendo a integridade e fiabilidade da rede.

1.2.2 Justificativa

Nas décadas de 80 e 90, quando os sistemas de iluminação ainda eram baseados em comunicação analógica, cada equipamento utilizava um canal dedicado para o seu controlo, sendo que, deste modo, para controlar 50 equipamentos seriam necessários obrigatoriamente 50 canais diferentes.

Com a criação do DMX, além da padronização da comunicação e dos equipamentos de iluminação, foi proporcionada a redução do comprimento de cabos e da dimensão dos equipamentos de controlo, pois passou-se a utilizar apenas um cabo para efetuar a comunicação entre duas máquinas com a capacidade de possuir até 512 canais e um ou mais controladores de iluminação cénica.

Contudo, mesmo com as melhorias introduzidas pelo DMX, permanecia por solucionar a problemática das limitações do ambiente, pois nem todos os eventos permitem a passagem ou a exposição dos cabos entre o palco e a *régie*.

Por questões de segurança, apresentação e custos, é cada vez mais premente a necessidade de reduzir os cabos utilizados em eventos.

Para além de possuírem custos relativamente elevados de aquisição e manutenção, não são de descurar os custos dos recursos operacionais envolvidos (humanos ou não) para montagem e respetiva desmontagem e os custos inerentes ao armazenamento e transporte.

Em função destas necessidades, a utilização de equipamentos sem fios tornou-se numa tendência e posterior exigência devido à flexibilidade e mobilidade propiciadas (ver Figura 1.2).



Figura 1.2 - Exemplo de uma possível solução DMX512 sem fios [3].

1.2.3 Identificação de objetivos específicos

Para a concretização dos objetivos da primeira parte do projeto é proposto o desenvolvimento de duas interfaces de comunicação sem fios, para realizar a comunicação entre um controlador e um recetor DMX512.

Face ao exposto, constituem objetivos específicos:

- ✓ Analisar o tipo de comunicação a implementar, e efetuar um estudo das possíveis tecnologias que permitam atingir os objetivos, entre as quais, são ponderadas tecnologias como o *Wi-fi*, o *Bluetooth*, o *ZigBee*, entre outras.
- ✓ Estudar e conceber um protótipo de comunicação ponto a ponto, num cenário físico delimitado (zona técnica e palco), onde os dois pontos poderão estar no máximo a 500 metros de distância;
- ✓ Estudar e projetar o melhor método e topologia a implementar, e selecionar os equipamentos para o desenvolvimento de protótipos, tendo em consideração variáveis como: custo, qualidade e fiabilidade;
- ✓ Definir o método de transmissão de dados e as respetivas configurações;
- ✓ Otimizar os sistemas relativamente à fiabilidade da comunicação de modo a reduzir qualquer atraso de comunicação, aumentar a capacidade de evitar a ocorrência de bloqueios, através da realização de testes dos protótipos em ambiente interior e exterior;
- ✓ Gestão das frequências de transmissão, a fim de maximizar a fiabilidade;
- ✓ Averiguar a possibilidade de criar algoritmo de comutação de canais de forma a verificar ocupação da frequência;
- ✓ Criar interface gráfica de configuração, gestão e parametrização do sistema.

A Figura 1.3 ilustra o diagrama de funcionamento da comunicação:

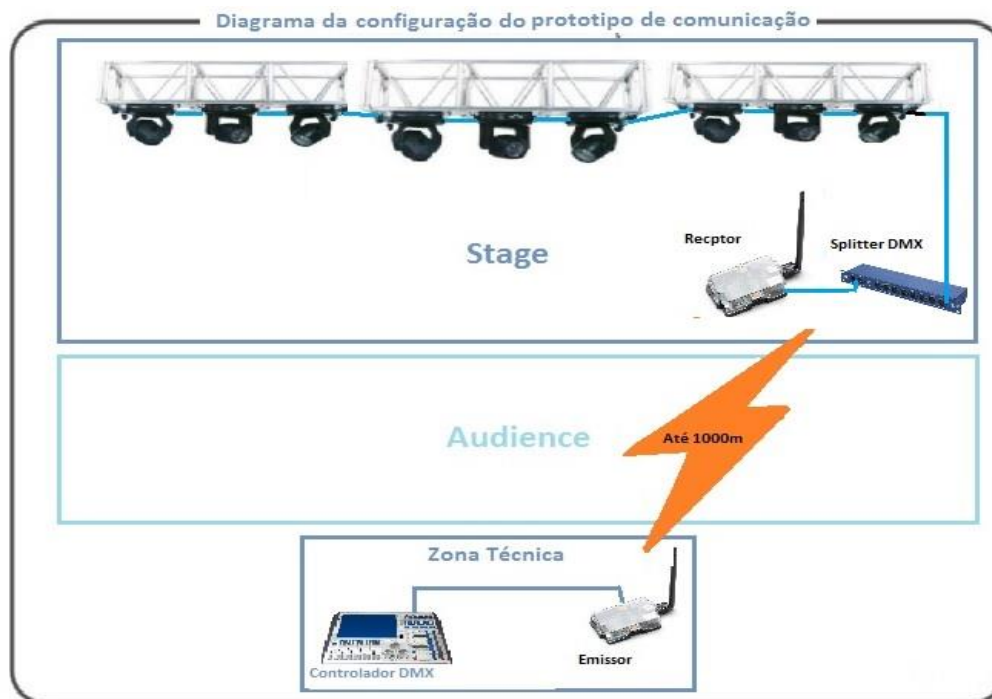


Figura 1.3 - Diagrama de configuração da comunicação ponto a ponto (1ª fase).

Na segunda fase do projeto, pretende-se complementar a comunicação, aplicando o controlo sem fios a todos os dispositivos recetores. Para tal, é proposto a conversão do dispositivo do palco, desenvolvido na primeira fase do projeto, num elemento capaz de efetuar a ponte com todas as máquinas recetoras do protocolo DMX512. Pretende-se, assim, integrar a capacidade de retransmitir a informação recebida por parte do módulo emissor da zona técnica a todos os recetores de comunicação (ver Figura 1.4).

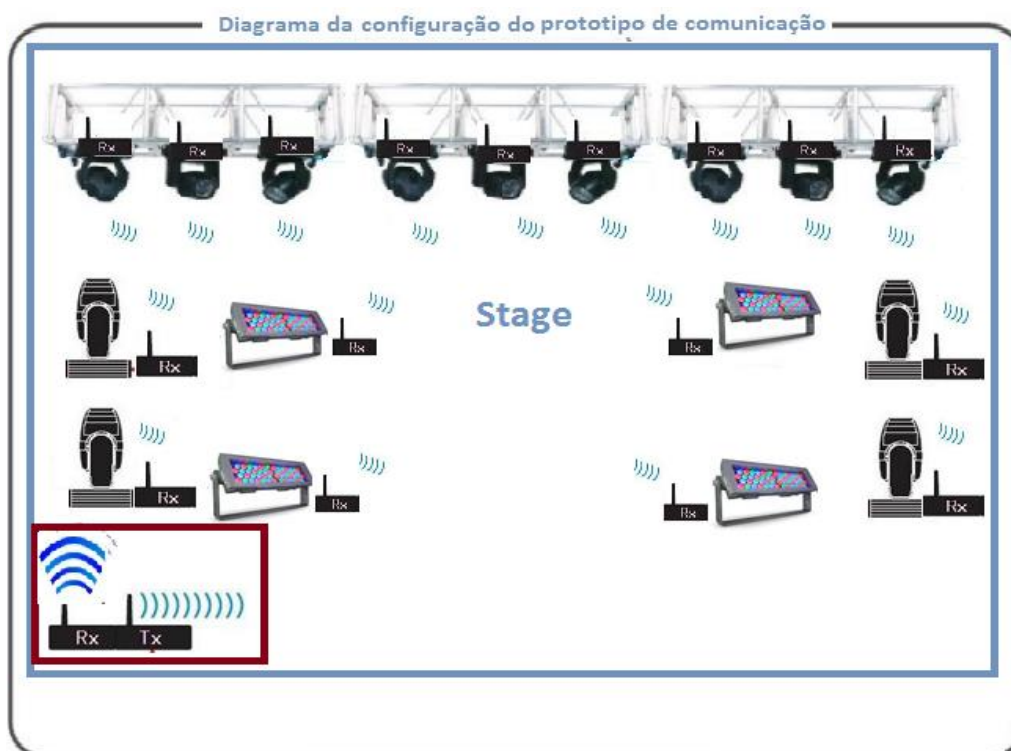


Figura 1.4 - Diagrama de configuração final da comunicação do projeto de investigação (2ª fase).

Para tal, é proposto o estudo do método que apresenta a maior fiabilidade ao nível da comunicação, tendo em consideração o crescimento exponencial do número de máquinas, o meio envolvente, bem como a forma e tecnologia de comunicação.

Através da análise de mercado efetuada, verificou-se a existência de dois métodos de transmissão: um deles consiste num único transmissor para todos os recetores, o outro assenta no pressuposto de que cada máquina requer um recetor e um transmissor para retransmitir para a máquina seguinte (ver exemplo da Figura 1.5).



Figura 1.5 - Exemplo de duas topologias de dispositivos comerciais [4] [5].

Além do descrito, pretende-se investigar a possibilidade de criar uma separação por linhas dos recetores, ou seja, aplicar o princípio do *splitter* à comunicação sem fios, avaliando posteriormente as vantagens e desvantagens.

O estudo e análise destas tipologias são o principal objetivo da segunda parte deste projeto, centrando-se na seleção e na construção de dispositivos recetores através de módulos recetores sem fios recorrendo à sua integração com microcontroladores.

De forma conclusiva, apresentar vantagens e desvantagens relativas aos produtos comerciais existentes e avaliar a viabilidade técnica do projeto, na íntegra.

Procurar a otimização ao nível de:

- Eficácia;
- Manutenção;
- Produtividade;
- Tempo;
- Economia;
- Autonomia.

2 CONCEITOS TEÓRICOS E TECNOLOGIAS UTILIZADAS

Neste capítulo serão apresentadas as diversas tecnologias abordadas durante o desenvolvimento do presente projeto, tendo em consideração as especificações do mesmo e a sua evolução, na perspectiva da aplicação desenvolvida.

2.1 DMX512

DMX512 (Digital MultipleX) é um padrão série de comunicação digital unidirecional, criado especificamente para o controlo de dispositivos de iluminação de espetáculos. Neste encontram-se definidas as regras de comunicação e interligação de dispositivos que, na altura, constituíam uma solução inovadora e económica, sendo o meio físico constituído por cabos XLR. O protocolo DMX512 assenta sobre o padrão série RS-485, definindo todas as características elétricas e de estrutura dos dados. O DMX512, curiosamente, não inclui a deteção de erros e correção automática de colisões, o que não impede que este seja amplamente utilizado nos equipamentos atuais de iluminação profissional e até doméstica. Atendendo às características referidas, este não é um protocolo adequado para aplicações de alto risco, tais como a pirotecnia, o controlo de movimentos de equipamentos, estruturas e plataformas de iluminação. Falsos disparos ou movimentos não programados podem ocorrer causados por interferência eletromagnética, descargas elétricas, estática, cabos excessivamente longos ou cabos de reduzida qualidade [6] [7].

Neste capítulo é efetuada uma breve revisão da história e evolução deste protocolo, tendo em conta as suas características, especificidades e aplicações.

2.1.1 História

O DMX512 foi desenvolvido em 1986 para substituir o *Analog MultipleX 192 (AMX 192)* pelo Instituto de Teatro e Tecnologia dos EUA (USITT), que iniciou o seu desenvolvimento e definição, como um protocolo assente no padrão RS-485 e um modo de comunicação entre controladores e *dimmers* (mesa e equipamento). Em 1989 este protocolo é apresentado de forma integral com a necessidade de executar pequenas revisões. Em 1990 surge uma versão revista com o intuito de solucionar problemas de flexibilidade e criar melhorias (DMX512/1990). Este foi dado como finalizado em 1994. Com um conceito simples, é rapidamente adotado pela indústria e aplicado no controlo de equipamentos de iluminação para teatros, televisão, prédios, centro comerciais e inclusive no controlo de estruturas e câmaras de videovigilância.

A partir de 1998 o *Entertainment Services and Technology Association (ESTA)* passou a ser o responsável pela manutenção do protocolo e iniciou uma revisão profunda para dar suporte não só a novas e melhores mesas de controlo, controladores dedicados e *dimmers*, mas também a outros equipamentos como projetores, *scanners*, *strobs*, *moving heads*, *laser's*, máquinas de fumo e efeitos. O término desta revisão e desenvolvimento de novas normas, regras e orientações fizeram com que o padrão DMX512 fosse outorgado, em novembro de 2004, pelo *American National Standards Institute (ANSI)*, como padrão regulamentado e orientado por esta entidade.

Esta efetua a sua publicação oficial, com o nome de "Tecnologia de Entretenimento - USITT DMX512-A (*Assíncronous Serial Digital Data Transmission Standard*) para Controlo de Equipamentos e Acessórios de Iluminação". Em 2008, uma nova atualização a este padrão, levada a cabo pela entidade criadora, voltou a definir o seu nome para "E1.11 - 2008, USITT DMX512-A", ou apenas "DMX512-A" [8]. Em janeiro de 2011, a entidade reguladora ESTA, efetua uma fusão com uma organização análoga do Reino Unido denominada PLASA, a qual mantém agora todos os padrões geridos pela ESTA, após a sua dissolução [7]. O DMX512 também se expandiu para usos tão diversos como: iluminação interior e arquitetónica, em escalas que vão desde luzes de ambiente e decoração a *outdoors* eletrónicos. O DMX pode atualmente ser usado para controlar inúmeros equipamentos, refletindo a sua importância nos mais diversos edifícios de cultura, lazer, comércio, entre outros [9].

2.1.2 RS-485

O RS-485, onde, "RS" é sigla para *Recommended Standard* (padrão recomendado) assente na transmissão diferencial de dados, é um protocolo ideal para transmissões de dados a longas distâncias e em ambientes suscetíveis a interferências eletromagnéticas. Este padrão série foi criado e desenvolvido pela *Electronic Industries Alliance* (EIA), até 1997 nominada *Electronics Industry Association*, entidade credenciada pela ANSI para desenvolver padrões e especificações técnicas de componentes eletrónicos, telecomunicações e Internet.

Esta entidade foi responsável igualmente pelo desenvolvimento dos padrões de comunicação série: RS-232, (EIA-232) e RS-422, (EIA-422), entre outros. No entanto, desde 1998, este padrão também é denominado por TIA/EIA-485, por ter sido publicado oficialmente, em conjunto, pela *Telecommunications Industry Association* (TIA) e pela *Electronics Industry Association* (EIA), esta última foi oficialmente dissolvida e o padrão agora é mantido pelo TIA como TIA-485 [10]. O RS-485 consiste num protocolo de camada física, pelo que não existem regras ou restrições referentes ao conteúdo ou forma de codificação das mensagens enviadas. Este facto possibilita a sua utilização para enviar qualquer tipo de dados. Como tal, apenas específica as características elétricas e os modos de operação da rede, não especificando os protocolos de comunicação a utilizar. A distância de transmissão depende da velocidade, tornando possível alcançar 35 Mbps em distâncias inferiores a 10 metros e até 100 kbps em distâncias até 1200 metros. Este permite a comunicação entre vários elementos participantes numa mesma rede de dados, normalmente interligados através de um cabo de par entrançado, para aumentar a imunidade a ruídos [11] [12].

De seguida, enumeram-se, as principais características básicas do padrão RS-485 [13]:

- Característica *multipoint*, de até 32 terminais remotos de comunicação;
- Grande imunidade a interferências eletromagnéticas devido à comunicação diferencial;
- Distância máxima de 1200 metros com um débito de 100 kbps;
- Necessidade de aplicação de resistência de término da rede;
- Transmissão de dados a 10 Mbps numa distância máxima de até 10 metros;
- Transmissão de dados em modo comum com tensões compreendidas entre -7V a +12V;
- Flexibilidade de configuração da rede e de protocolo de comunicação;
- Redes locais económicas e robustas, quando comparadas a outras, tais como: *Ethernet*, *FieldBus*, entre outras;
- Possibilidade de migração de padrão sem perda de características elétricas.

A transmissão diferencial de dados funciona da seguinte forma: qualquer transmissor RS-485 possui dois canais independentes conhecidos como A e B, que transmitem níveis de tensão iguais, porém com polaridades opostas (VOA e VOB ou simplesmente VA e VB). Por esta razão, é importante que a rede esteja ligada com a polaridade correta.

Este sinal diferencial, lido em relação ao ponto central da carga, é que é interpretado como sinal de transmissão. Qualquer tensão maior que 200 mV é um nível alto ou “marca”. Uma tensão menor que -200 mV é um nível baixo ou “espaço”. Níveis compreendidos entre -200 mV e +200 mV são indefinidos e interpretados como ruído [13].

Embora os sinais sejam opostos, um não é o retorno do outro, isto é, uma vez que constituem um par diferencial, cada sinal tem o seu retorno pela massa ou por um terceiro condutor de retorno. Não obstante, o sinal deve ser lido pelo recetor de forma diferencial sem referência à terra ou ao condutor de retorno.

Com a comunicação *RS-485* podemos estabelecer *simplex*, *half-duplex* e *full-duplex*. No entanto, para a comunicação *full-duplex*, o fluxo de dados é bidirecional, os dados podem ser enviados e recebidos ao mesmo. Um exemplo da sua aplicação como camada física adjacente é o protocolo *Modbus* ou o protocolo *Profibus*.

O *RS-485* possui assim, uma aplicação muito ampla e variada, que inclui a sua utilização nos mais diversos ramos da automação industrial, na comunicação entre dispositivos em setores como o ramo automóvel ou até mesmo o aeronáutico ou espacial. Outros exemplos da sua aplicação incluem automação de edifícios, de sistemas de energia, sistemas de transporte e logística, entre outros [10] [12].

A título de resumo, descrevem-se na Tabela 2.1, as características elétricas do padrão RS-485:

Tabela 2.1 - Características elétricas do padrão RS-485 [13].

Parâmetro	Valor
Tipo de comunicação	Diferencial
Modo de operação	<i>Half-Duplex</i>
Topologia da rede	<i>Multipoint</i>
Número máximo de transmissores	32 unidades
Número máximo de recetores	32 unidades
Comprimento máximo de linha	1200 metros
Taxa máxima de comunicação	10 Mbps
Tensão mínima em modo comum	-7 a 12 V
Tensão mínima de transmissão em carga	+/- 1,5 V
Impedância mínima de carga	60Ω
Impedância de entrada do recetor	≥12KΩ
Sensibilidade do recetor	+/- 200 mV

2.1.3 Topologia da rede DMX

Uma rede DMX512 emprega uma topologia de barramento *multi-drop* do tipo *master-slave* de nós muito próximos, que é usualmente denominado por cadeia (*Daisy Chain*). A rede padrão consiste num único controlador DMX512 (que é o mestre da rede) e um ou mais dispositivos, muitas das vezes denominados por *Fixtures* ou máquinas (ver Figura 2.1).



Figura 2.1 - Exemplo topologia de rede DMX512 [14].

Uma rede tem a capacidade de transmitir 512 *bytes* de forma assíncrona. Cada *byte* corresponde a um endereço e cada dispositivo pode necessitar de alocar vários endereços para ser controlado.

O desenvolvimento tecnológico levou à utilização de equipamentos cada vez mais complexos e com mais parâmetros de controlo, como a intensidade, a posição, gobos, cores, entre outros, o que se traduz numa ramificação do tipo de máquinas de acordo com as suas características, tal como se encontra representado na Figura 2.3. Cada parâmetro é controlado por, no mínimo, um canal, pelo que o desenvolvimento de *moving heads* mais complexos levou a que os universos atingissem o limite de largura de banda, e os 512 canais passaram a não ser suficientes para ligar todos os dispositivos em aplicações de maior dimensão. É normal hoje em dia haver *moving heads* com 16 canais endereçados, o que significa que apenas 32 unidades podem ser ligadas a um universo (que corresponde também a uma limitação do RS-485, como explicitado na secção 2.1.2).

Os controladores DMX permitem o controlo e programação de diversos equipamentos, estes podem ser interfaces para computador conectados à linha DMX, controladores DMX programáveis, também designados de mesas de iluminação ou controladores dedicados DMX (ver Figura 2.2) [15] [14] [16].



Figura 2.2 - Exemplo de controladores DMX512.

Na literatura podemos, frequentemente, encontrar as denominações DMX 1024 e DMX 128. Estas denominações referem-se ao tamanho do universo do controlador DMX utilizado [9].

O universo DMX identifica a porta física do controlador, normalmente corresponde ao terminal de saída e representa a capacidade de controlar os elementos que integram a sua rede.

Os controladores mais simples, geralmente, apenas possuem uma ficha de saída permitindo que estes controlem apenas um universo, enquanto as grandes mesas de controlo (mesas de operador) podem ter a capacidade de controlar múltiplos universos, com uma ficha de saída fornecida para cada universo.

O DMX1024 significa que existem 2 universos DMX512, em paralelo, enquanto o DMX128 significa que o controlador apenas pode criar uma rede até 128 canais.

Na prática, somente os endereços até 127 podem ser usados. Isto porque o endereço 0 é considerado um endereço de sincronização e inicialização e não pode ser considerado um endereço válido no dispositivo elétrico.

Esta configuração é válida para todos os outros tamanhos de universos, uma vez que se trata de uma das especificações do protocolo DMX512 [8] [9]. Na Tabela 2.2 é possível verificar o número de canais correlacionado com o universo DMX.

Tabela 2.2 - Número de canais conforme universo DMX.

Universo	Canais
1	1-512
2	513-1024
3	1025-1536
4	1537-2048
5	2049-2560
6	2561-3012

Na verdade, o único fator que restringe o aumento de mais universos DMX512 é a velocidade do barramento de comunicação.

As operações dos equipamentos são as mesmas, independentemente do universo em que se encontrem, diferindo apenas na seleção da interface de saída do controlador.

Em sistemas de universo múltiplo, devemos identificar qual a distribuição da cablagem integrante das diversas linhas, uma vez que se os equipamentos não estiverem conforme a parametrização, aquando do endereçamento, a comunicação não se estabelecerá de forma correta.



Figura 2.3 - Exemplo de inúmeros dispositivos recetores ou *fixtures*.

Por especificação, a rede DMX exige um elemento "terminal" que terá de ser implementado no último elemento constituinte da linha de comunicação. Este elemento é, uma ficha DMX macho autónomo com uma resistência de 120 Ω com pelo menos ¼ de watt, ligada ao par diferencial de dados primário, esta resistência corresponde à impedância característica do cabo e serve para evitar a reflexão na linha de comunicação RS-485.

Atualmente, a maior parte dos recetores DMX, possui internamente, a montante da sua saída de comunicação, um circuito terminal embutido, que pode ser ativado manualmente, por *software* ou mesmo por deteção automática da ausência de uma ficha de saída.

Um exemplo destes pode ser visto na Figura 2.4 [15] [17].

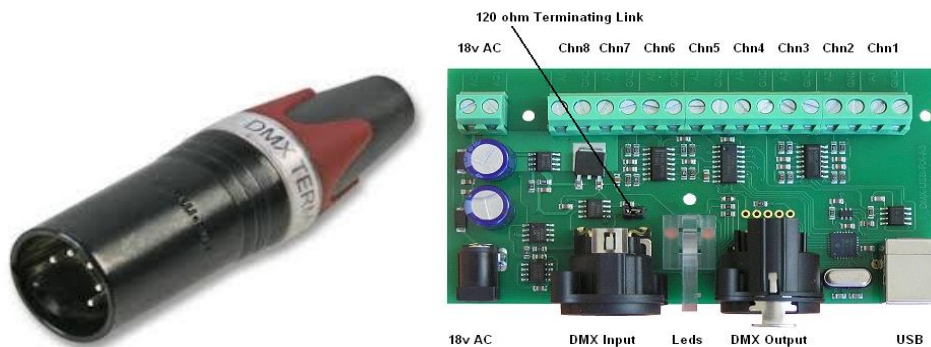


Figura 2.4 - Exemplo de elementos terminais de comunicação DMX512 [18] [19].

Para comunicações a curtas distâncias e poucos dispositivos, é possível operar sem término no entanto, a especificação do uso do terminal e a utilização de cabos com impedância correta deve ser seguida, com o aumento do comprimento do cabo e/ou o número de dispositivos.

2.1.4 Cabos e fichas de comunicação

A interligação dos elementos constituintes da rede é tradicionalmente efetuada por um cabo DMX512, sendo que este é composto por dois pares entrançados e respetiva blindagem. Atualmente, as especificações apenas definem o uso de um dos pares para estabelecer a comunicação diferencial, o segundo par apesar de indefinido, é exigido pela especificação elétrica.

As características elétricas do cabo DMX512 são especificadas em termos de impedância e capacitância, embora haja muitas vezes considerações mecânicas e outras que também devem ser consideradas.

Os tipos de cabo que são apropriados para o uso de DMX512 terão uma impedância característica nominal entre os 110 e 120 ohms, e uma capacitância entre condutores de 65 pF/m e entre os condutores e a blindagem menor de que 115 pF/m, segundo a especificação ANSI E1.27-1 2006 (R2016) [17].



Figura 2.5 - Exemplo de um cabo regulamentado de comunicação DMX512A [20].

Os cabos regulamentados pelo padrão, utilizados nas redes DMX512 utilizam fichas XLR5, com uma ficha macho numa extremidade e uma ficha fêmea na outra extremidade, como se pode observar na Figura 2.5.

A norma USSITT E1.11 DMX512A recomenda que as fichas de entrada (*Input*) sejam formadas através de fichas macho e as fichas de saída (*Output*) sejam fichas fêmea [21]. Esta norma, também recomenda que no máximo a cada 32 dispositivos de uma linha seja inserido um repetidor, usualmente designado *splitter DMX* (ver Figura 2.6). O deterioramento da blindagem ou a má conservação dos cabos, pode gerar ruídos que não são reconhecidos pelos aparelhos e dispositivos, o que pode originar falhas de comunicação graves [17].



Figura 2.6 - Exemplo de um *splitter* DMX512 [22].

A norma DMX512-A (ANSI E1.11-2008) especifica que, quando as fichas são utilizadas, devem ser usadas fichas do tipo XLR de cinco pinos (XLR-5), e que não é recomendado o uso de qualquer outra ficha do tipo XLR.

A norma definiu, também, o uso de fichas modulares de oito pinos (8P8C ou RJ-45) para instalações fixas onde a união e o acoplamento amovível do equipamento não são necessários.

De salientar que vários fabricantes usaram outros *pinouts* para fichas RJ-45 antes desta inclusão no padrão. São permitidas outras formas de fichas em equipamentos onde o XLR e o RJ-45 não têm as dimensões apropriadas ou são considerados inadequados, como por exemplo, em equipamentos destinados a instalação permanente [8]. A Figura 2.7 apresenta as fichas e tomadas admitidas pelo padrão DMX512-A.



Figura 2.7 - Exemplo de fichas e tomadas DMX512 conforme normas do protocolo DMX512A [23].

A Tabela 2.3 e a Tabela 2.4 referenciam os contatos das fichas XLR5 e RJ-45, respectivamente.

Tabela 2.3 - Esquema de ligação das fichas XLR-5 recomendada pela norma [8].

XLR-5 Pinout	
1	Sinal comum
2	Dados 1- (<i>Primary Data Link</i>)
3	Dados 1+ (<i>Primary Data Link</i>)
4	Dados 2- (<i>Link de dados secundário opcional</i>)
5	Dados 2+ (<i>Link de dados secundário opcional</i>)

Tabela 2.4 - Esquema de ligação das fichas RJ-45 recomendada pela norma [8].

RJ-45 Pinout	
1	Dados 1+ (<i>Primary Data Link</i>)
2	Dados 1- (<i>Primary Data Link</i>)
3	Dados 2+ (<i>Link de dados secundário opcional</i>)
4	Não atribuído
5	Não atribuído
6	Dados 2- (<i>Link de dados secundário opcional</i>)
7	Sinal comum (0 V) para dados 1
8	Sinal comum (0 V) para dados 2

O encaixe da ficha modular 8P8C corresponde ao esquema de emparelhamento de pares entrançados do cabo de categoria 5 (Cat5).

Contudo, muitos fabricantes de baixa qualidade e fabricantes de réplicas de máquinas desenvolvidas adotam a utilização das tomadas XLR-3, de forma a aumentar o seu poder de venda uma vez que entendem que uma pequena empresa ou utilizador podem não necessitar de adquirir cabos normalizados, com fichas XLR-5, por já possuírem cabos de áudio com fichas XLR-3.

De salientar que, como foi visto anteriormente, estes cabos não possuem as características mínimas exigidas pelo padrão e a utilização de fichas XLR-3 é proibida pela secção 7 do ANSI E1.11 - 2008 [8] [16].

2.1.5 Características Elétricas

Como visto anteriormente, os requisitos elétricos são geridos pela camada física do padrão TIA/EIA-485. O alcance de saída do *driver* varia entre o mínimo de +/- 1.5V a +/- 6V e a sensibilidade do recetor deve ser sempre superior a 200 mV para detetar que não existem erros de comunicação, tornando apenas a diferença relativa entre os dois fios [23].

A vantagem da utilização da transmissão diferencial consiste na capacidade de utilização a longas distâncias e no facto de ser bastante resistente a interferências eletromagnéticas.

Para cumprir os requisitos propostos no padrão DMX512A, segundo as especificações ANSI E1.11-2008 [8], é necessário que os recetores sejam isolados, o que ajuda a estabelecer o mesmo potencial na comunicação e a evitar diferenças de tensão potencialmente elevadas, prejudiciais para a comunicação e para as próprias máquinas. Muitos recetores de baixo custo não são isolados e podem causar problemas de comunicação.

A especificação elétrica E1.11 (DMX512 2004) aborda a interligação da blindagem do cabo e de todos os elementos metálicos comuns do sinal DMX512 à massa ou elemento terra da máquina.

Especificamente, o padrão recomenda que as portas do transmissor (porto de saída do controlador DMX512) possuam uma ligação de baixa impedância entre o sinal comum e a massa. Recomenda ainda que, os recetores tenham uma conexão de alta impedância entre sinal comum e terra; Tais portas são referidas como isoladas [6]. Também o recetor deve, por norma, ser isolado e possuir valores de resistência, entre qualquer pino da ficha e o *chassi* da máquina, maiores que $22\text{M}\Omega$ a 42V, conforme a Figura 2.8.

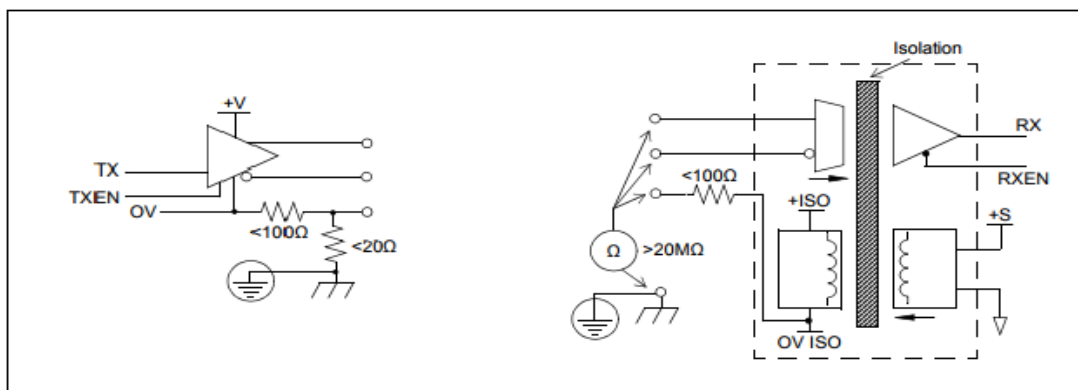


Figura 2.8 - Circuito transmissor do controlador (esquerda) e circuito do recetor (direita) [23].

De acordo com a Figura 2.9, a transmissão dos *bits* DMX, é efetuada através de uma comunicação diferencial entre dois condutores, sendo implementada pelo sinal digital de nível lógico '1' ou "mark", sinal de alto potencial entre o condutor positivo e o negativo e, em oposição, pelo sinal digital de nível lógico '0' ou "space", o sinal de baixo potencial entre o condutor positivo e o negativo [24].

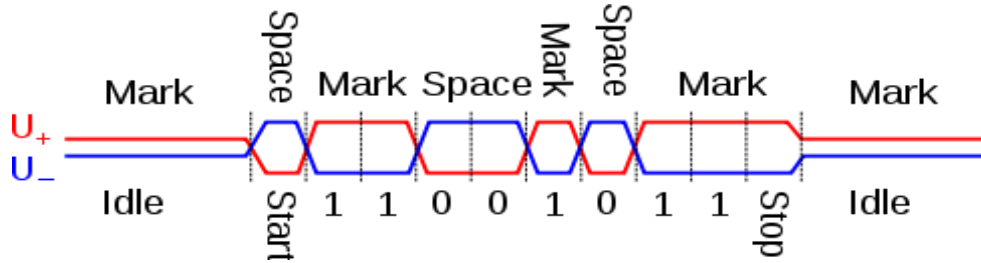


Figura 2.9 – Exemplo da variação do par diferencial [24].

Esta metodologia apresenta uma grande vantagem face à presença de ruídos inseridos no sistema, pois o ruído afeta de igual forma ambos os condutores do par diferencial, mantendo-se a diferença de potencial inicial constante. Esta comunicação série assíncrona, é efetuada com um débito de 250kbps, o que permite o tráfego de 40 pacotes de 512 canais a cada segundo.

A taxa mínima de transmissão do protocolo RS-485, é calculada através da Equação 2.1, considerando a taxa de transmissão (T), e a distância máxima (D) entre pontos de comunicação.

Equação 2.1 - Taxa mínima de transmissão do protocolo RS-485

$$D < \frac{10^8}{T}$$

Como a taxa mínima da transmissão do padrão RS-485 é de 100kbps a distância máxima D é de 1000 metros e como a taxa mínima de transmissão do DMX512A é de 250kbps a distância real máxima do protocolo DMX é de 400 metros.

Os dados são enviados em pacotes que se repetem continuamente, começando com bits de inicialização que fornecem a informação aos recetores de que o pacote se encontra em atualização. Em sequência são enviados os dados correspondentes a cada canal, iniciando no canal 1 e terminando no canal 512, ou em qualquer outro inferior, dependendo do último canal endereçado.

2.1.6 Pacote DMX

Um pacote de dados consiste num bloco formatado de informação transportado numa rede com comutação de pacotes, os quais podem conter até 512 *frames* de 8 *bits* de dados.

O pacote de dados é retransmitido continuamente, o que ajuda a minimizar quaisquer falhas de dados, sendo estas corrigidas aquando da receção do pacote seguinte. Se nenhum pacote for transmitido no intervalo de 1 segundo, a maioria dos recetores muda para uma configuração padrão ou para um modo de *standby*. A taxa de atualização máxima de 512 canais é de 44 atualizações por segundo, o que permite um *dimming* com pouca ou nenhuma cintilação visível em lâmpadas incandescentes [25].

Na Figura 2.10, é possível observar uma trama de dados DMX. Um pacote completo contém 512 frames e é composto por um *Start Frame* e os restantes 511 frames e terá uma duração de 22 milissegundos. Cada *frame* é constituído por um total de 11 bits, sendo que 1 bit é de início (*Start Bit*), 8 bits são de dados e 2 bits de paragem (*Stop Bits*).

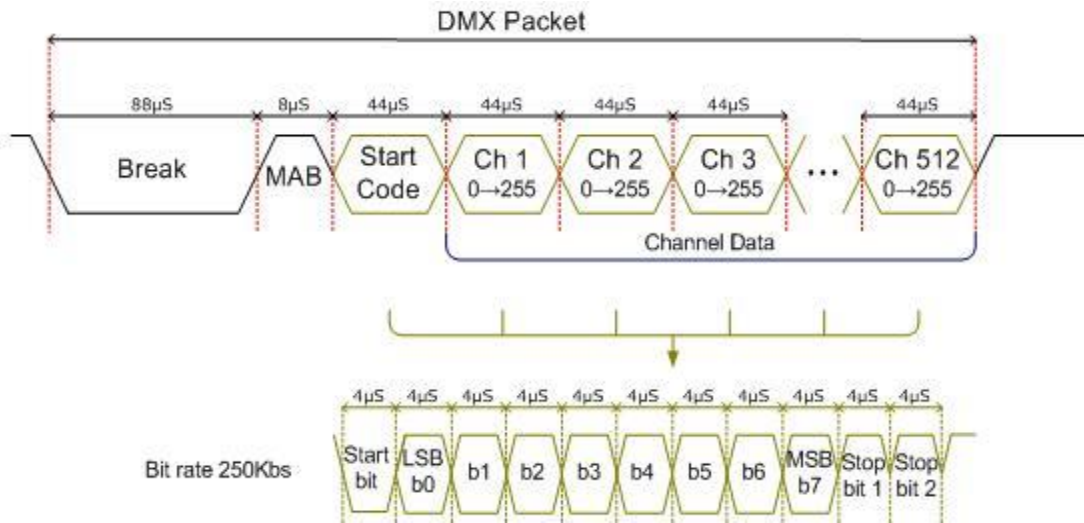


Figura 2.10 - Pacote de dados do protocolo DMX512 [25].

Normalmente, apenas os endereços utilizados são enviados, o que possibilita reduzir a dimensão do pacote DMX. O *frame 0* ou endereço 0 é um *frame* de sincronização que não pode ser endereçado nos equipamentos.

O pacote em si, não contém informações de endereço, sendo que este é obtido a partir da posição do *frame* no pacote e, portanto, todos os *frames* devem ser enviados de forma individual, coordenada e incremental [26].

Na Figura 2.11 é possível observar a estrutura de comunicação, anteriormente referida, de forma mais pormenorizada, encontrando-se descritos, abaixo desta, na respetiva legenda cada delimitador da *frame*.

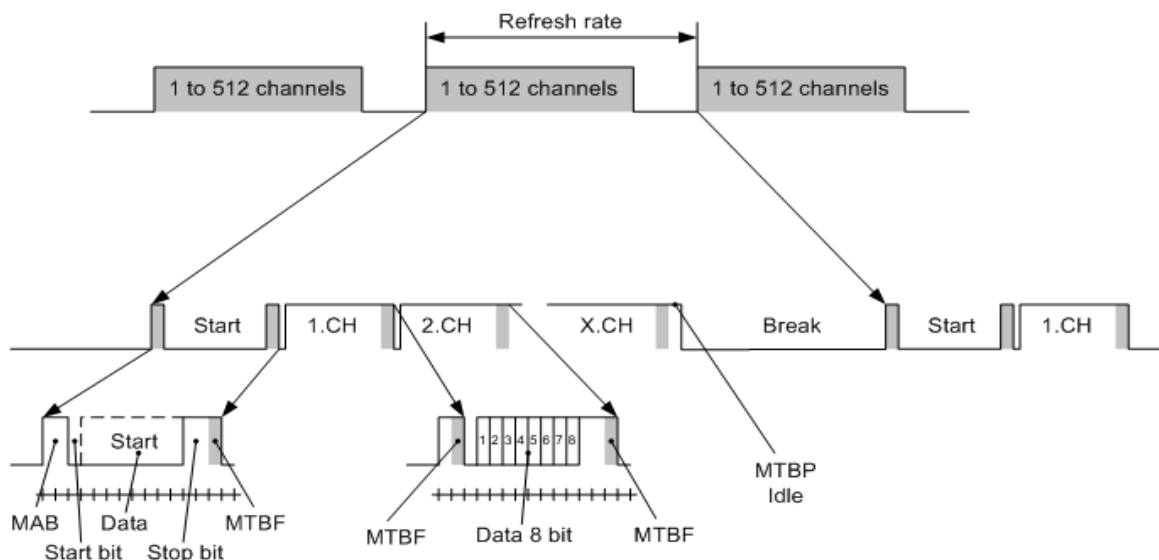


Figura 2.11 - Diagrama estrutural de um pacote de dados DMX [26].

- 1) **IDLE**: em espera ou sem sinal DMX, na falta de um pacote DMX válido, o sinal da saída DMX será um sinal de nível alto contínuo (*High*);
- 2) **Break**: delimitador de início do pacote DMX, efetuado pela transição do sinal para um nível baixo (*Low*) por um período mínimo de 88 microssegundos ou 2 pulsos. Isto significa que são enviados 22 *bits* no nível baixo (*Low*) de forma consecutiva. A esta sequência, atribui-se a designação de *Break*, e este deve ocorrer com uma duração máxima de um segundo. A sua função, seguida de uma marca designada *Mark-After-Break* (MAB) é sincronizar todos os recetores com o transmissor.
- 3) **Mark After Break (MAB)**: Este delimitador surge imediatamente após o delimitador *Break*, efetuando uma alteração do sinal de nível baixo (*Low*) para um sinal de nível alto (*High*) por um período mínimo de 8 microssegundos ou 2 pulsos.
- 4) **Start Code (SC)**: O *SC*, ou código de início, representa o início do envio consecutivo de *frames* de dados para cada canal. O *SC* possui sempre a mesma estrutura, formato e ocupação temporal, exatamente 44 microssegundos ou 11 pulsos, o que corresponde a 1 *frame* ou canal. Por definição, o canal 0, também designado *Null Start*, deve ser sempre nulo, (*databyte=0*), e nenhum outro valor pode ser utilizado. Esta opção foi deixada em aberto, pela ESTA, pois no futuro pretende-se utilizar os *bits* do delimitador *SC* para segregar os pacotes de dados, de forma a habilitar os recetores a identificar para que tipos de equipamentos serão os dados enviados [7].
- 5) **Channel Data (CD)**: É o delimitador que indica que o código de início é válido para recetores DMX e inicia o envio de pacotes de dados para cada canal de forma consecutiva. Os *frames* de *CD* possuem a mesma estrutura e adotam a mesma lógica de envio do delimitador *SC*, contudo, estes definem os canais de 1 a 512 (ou menos) [27]. Como visto anteriormente, cada *frame* é composto por 11 pulsos:
 - a. O primeiro pulso, designado *Start Bit* ou *bit* de início, especifica o tipo de dados do pacote e é sempre de nível baixo (*Low*), o que principia a transição de alto para baixo, indicando o início do *frame*.
 - b. Este é seguido pelo atual *byte* de dados, composto por 8 *bits* de configuração de controlo do respetivo recetor, onde cada *bit* pode ser qualquer valor compreendido entre 0 a 255.
 - c. O *frame* termina com dois *bits* consecutivos de nível alto (*High*), indicando os dois *bits* de paragem (*Stop bits*), o fim de entrega de informação a um determinado canal e a mudança de estado para modo inativo.
- 6) **Mark Time Between Frames (MTBF)**: Delimitador de modo inativo, pode variar temporalmente entre o valor 0 e 1 segundo, contudo, deve ser o mais curto possível. Este é enviado após o término de um *frame*, ou seja, antes de cada *start bit* do seguinte
- 7) **Mark Time Between Packets (MTBP)**: Depois do último *bit* de paragem do delimitador *CD* ter sido enviado, o pacote atual é enviado. Após este envio, o próximo pacote pode começar com um novo delimitador *Break* e *MAB*. No entanto, uma espera, (*IDLE*), pode ser inserida na comunicação, aplicando um sinal de nível alto (*High*) inserido entre os pacotes, esta espera designa-se *MTBP*, e o seu tamanho varia de zero a um segundo. Este tempo é definido pelo fabricante, aquando o desenvolvimento da arquitetura do seu recetor DMX, no entanto, como visto este valor deve ser o mais baixo possível.

Os parâmetros de temporização do protocolo DMX512A, segundo a normativa ANSI E1.11 - 2008 (R2013), podem variar numa ampla gama, analisando o seu comportamento, resumem-se na Tabela 2.5 [27].

Tabela 2.5 - Resumo dos limites de parâmetros de temporização do protocolo DMX512A.

Descrição	Valor	Requisição	Valor Ideal ou Típico	Requisição	Valor	Unidades
<i>Break</i>	92	≥92	176	≥88	1000000	μs
<i>MAB</i>	8	≥8	12	≥8	1000000	μs
<i>Bit</i>	4	4	4	4	4	μs
<i>Frame</i>	44	44	44	44	44	μs
<i>MTBF</i>	0	<1000000	Indefinido	<1000000	1000000	μs
<i>MTBP</i>	0	<1000000	Indefinido	<1000000	1000000	μs
<i>Break to</i>	1204	<1000000	Indefinido	>1196	1250000	μs

2.1.7 Estado da arte

➤ 8 bits versus 16 bits

O DMX não exige um método de codificação de 16 *bits* para pacotes *Null Start Code*, no entanto muitos parâmetros em iluminação robotizada usam codificação superior a 8 *bits*. Para controlar estes parâmetros de forma mais precisa, alguns dispositivos elétricos usam dois canais para parâmetros que requerem maior precisão.

O primeiro dos dois canais controla o valor grosseiro (256 passos para toda a amplitude de movimento) e o segundo permite a sua aproximação (256 passos para cada passo grosseiro). Tal cenário proporciona um intervalo de valores de 16 *bits* de 65536 etapas, permitindo uma maior precisão, comparativamente a um método de codificação de 8 *bits*. Esta codificação é muitas vezes aplicada aos parâmetros de controlo de movimento verticais e horizontais, designados como *Tilt* e *Pan*, respetivamente.

➤ Comunicação sem fios

Recentemente, os adaptadores DMX512 sem fios tornaram-se populares, especialmente em instalações de iluminação arquitetónica, onde o comprimento dos cabos, de acordo com as normas, não deve ser excessivamente longo. Tais redes empregam tipicamente um transmissor sem fios no controlador, com recetores estrategicamente colocados próximos dos dispositivos para converter o sinal sem fios para sinais de rede com fios DMX512 convencionais.

Embora as redes sem fio DMX512 possam funcionar em distâncias superiores a 910 metros em condições ideais, a maioria dos *links* sem fios DMX512, são limitados a uma distância máxima compreendida entre 50 a 100 metros, para garantir uma operação confiável.

O primeiro sistema DMX512 sem fios comercializado foi baseado em tecnologia de espectro de difusão em frequência variável, FHSS (*Frequency-Hopping Spread Spectrum*) usando modems sem fios comerciais. Outros sistemas deste tipo continuam a ser utilizados, mas com maior largura de banda. Os sistemas FHSS tendem a perturbar outros tipos de sistemas de comunicação sem fios.

Nos sistemas DMX sem fios mais recentes, estas perturbações foram solucionadas usando o recurso de conversão adaptativa e a coexistência cognitiva, uma técnica para detetar e evitar sistemas sem fios circundantes, para evitar transmitir em frequências ocupadas. Enquanto os protocolos *DMX-over-Ethernet*, como *E1.31 – Streaming Architecture for Control Networks (sACN)*, podem ser utilizados para enviar dados DMX através de *WiFi* [17].

➤ *System Information Packet*

Muitas alternativas ao DMX512 foram propostas para abordar limitações percebidas, como a contagem de *slot* máxima de 512 por universo, o sinal unidirecional e a falta de deteção de erro inerente. A revisão DMX512-A de 2004 adicionou pacote de informações do sistema, um *Session Initiation Protocol (SIP)*. Este pacote pode ser intercalado com pacotes nulos. Uma característica dos SIPs é que eles permitem a verificação de *checksum* para os dados *DMX Null*. Este recurso é dirigido para grandes instalações, onde são transmitidas, em baixa frequência, informações sobre parâmetros do controlador intercalados com dados de iluminação normais. No entanto, os SIPs raramente foram implementados [28].

2.2 RDM

O *RDM, Remote Device Management* é um protocolo que se enquadra no padrão de dados DMX512 e que converte o protocolo DMX512 num sistema bidirecional (*half-duplex*), o que permite o *feedback* de diagnóstico dos dispositivos elétricos para o controlador, estendendo o padrão DMX512. A comunicação bidirecional permite que o controlador interroge outros dispositivos RDM e faça alterações nas suas configurações. Desta forma, este protocolo substitui todas as tarefas de configuração e ajuste que tinham de ser realizadas de forma física em cada uma das máquinas pertencentes ao sistema a controlar [29]. Neste subcapítulo é efetuada uma breve revisão da história e evolução deste protocolo, tendo em conta as suas características, especificidades e aplicações.

2.2.1 Definição

O RDM é um desenvolvimento construído a partir do protocolo de controlo de iluminação do DMX512 que permite a gestão remota integral de dispositivos, máquinas e equipamentos RDM e também DMX. Este protocolo foi oficialmente reconhecido pela normativa ANSI E1.20-2006, usando os núcleos do protocolo DMX existente e a sua configuração.

Os valores DMX normais são enviados ao longo da linha do controlador e interpretados por todos os dispositivos nesse universo DMX. Os valores RDM são enviados de outra forma, mas não constantemente. Durante uma interação RDM, o controlador pode solicitar um ou mais dispositivos para obter algumas informações que eles retornam, no entanto, estas interações diminuem até a próxima consulta. Desta forma, o RDM não consome 50% da capacidade do sinal, uma vez que durante o funcionamento normal, a parte RDM do sinal representa apenas cerca de 10% a 15% da ação. O controlador também pode enviar instruções aos dispositivos, como configuração de parâmetros, sem solicitar uma resposta contudo, para que o sistema funcione, as partes individuais devem ser identificáveis. O RDM usa o UID, *Unique Identifier*, número de identificação único de cada dispositivo, uma configuração de *hardware* inalterável e não coincide com o endereço de início DMX. Para determinar o UID de cada dispositivo RDM na linha, o controlador usa um processo "one off" denominado de "discovery".

O processo real de descoberta do UID, é efetuado com recurso ao mapeamento de cada dispositivo pertencente à rede, para que o controlador possua categoricamente o UID de todos os equipamentos RDM [29]. O UID consiste num ID de fabricante e respetivo número de série.

O protocolo RDM possibilita outras operações, tais como:

- ✓ Identificação e configuração dos dispositivos constituintes da rede;
- ✓ Configuração de endereços de outras máquinas e dispositivos DMX;
- ✓ Supervisão e configuração do modo de funcionamento e fixação;
- ✓ Solicitação de mensagens de estado das máquinas RDM;
- ✓ Monitorização dos sensores e obtenção remota de códigos de falha;
- ✓ Relatórios de *status* de dispositivos elétricos ou outros dispositivos conectados;
- ✓ Configuração de ajuste de parâmetros de desempenho e de *display*;
- ✓ Monitorização e controlo de atributos como inversão de movimento, entre outros;
- ✓ Compatibilidade com o *hardware* DMX existente;
- ✓ Entre outras operações mais específicas desenvolvidas pelos fabricantes das máquinas.

2.2.2 História

O RDM foi desenvolvido em 2006, e demorou algum tempo até que o seu uso se tornasse amplo. A norma E1.11-2004, uma revisão do padrão DMX512-A, inicialmente estabeleceu as bases para o protocolo RDM através da definição de *Enhanced Functionality*.

Existe atualmente vários controladores dedicados de iluminação *mainstream* que suportam RDM, bem como uma crescente lista de dispositivos RDM, como *scanners*, *dimmers* e *moving heads*. Os produtos de distribuição de dados, incluindo *links* DMX / RDM sem fio, estão agora disponíveis. O benefício avançado deste sistema depende das configurações físicas e circunstanciais do *layout* da aplicação, das capacidades do elemento ou elementos de controlo e respetivas máquinas utilizadas e seus utilizadores.

Embora os fabricantes de iluminação profissional estejam a apostar exponencialmente no desenvolvimento de equipamentos baseados no RDM, a implementação atual é lenta e, não raras vezes, não suportada pelo mercado. Comopositor, o RDM tem novos estudos e desenvolvimentos em integração de redes, *média* e a redefinição integral do próprio controlo de iluminação. Detalhes do padrão e os respetivos testes podem ser encontrados no ANSI E1.20 - 2006 [30].

2.2.3 Funcionamento

O protocolo RDM e a sua camada física foram projetados para serem compatíveis com equipamentos DMX. Por norma todos os recetores DMX512 são compatíveis em sistemas mistos com um controlador RDM e equipamentos habilitados ao RDM. Os recetores DMX e os elementos RDM podem ser usados por um controlador DMX normal, do ponto de vista do utilizador, o *layout* do sistema é muito semelhante a um sistema DMX, uma vez que os *drivers* da saída DMX estão sempre habilitados.

O protocolo RDM foi projetado para que nunca haja colisão de dados, este fenómeno só pode estar presente quando o sistema se encontra em modo “*discovery*”. Para garantir essa falta de colisões, ao mesmo tempo que possibilita a implementação em diferentes plataformas, há momentos em que todos os *drivers* de linha devem ser desativados.

A investigação desenvolvida pelo ANSI E1.20 – 2006, mostrou que muitos *drivers* EIA-485 projetados para operar a 5 volts apresentam resultados positivos nos testes necessários, no entanto este fenómeno não é tão claro para os *drivers* projetados para operar a 3,3 volts.

A especificação DMX512, como já foi visto, requer que os pacotes DMX comecem com o código de início, normalmente 0x00. Ao usar o código de início 0xCC, os pacotes RDM podem ser inseridos com segurança entre os pacotes de dados DMX sem que haja incompatibilidade com quaisquer dispositivos DMX antigos não habilitados ao RDM. A necessidade de acoplar toda a comunicação RDM nos pinos 2 e 3 suscitou preocupações ao nível da colisão de dados [30]. O padrão RDM aborda este problema, garantindo que, em todos os casos (exceto em modo “*discovery*”), apenas um dispositivo está autorizado a transmitir em qualquer momento. Então, somente o controlador pode iniciar uma troca de dados RDM, os elementos a controlar só respondem caso seja pedido por este.

A comunicação RDM pode ser dividida em três tipos:

- ✓ *Discovery* - Descoberta;
- ✓ *Unicast Communication* - Comunicação direcional;
- ✓ *Broadcast Communication* - Comunicação de difusão.

Discovery

O modo “*discovery*” é a única situação em que as colisões de dados podem ocorrer. O controlador irá transmitir um comando de descoberta para todos os dispositivos e aguardar uma resposta. Se houver mais de um dispositivo conectado, as respostas simultâneas provavelmente resultarão numa colisão de dados, e o controlador não receberá uma resposta corretamente formatada. O controlador irá melhorar a sua pesquisa para um menor alcance de UID de acordo com um padrão de pesquisa binária. Assim que o controlador receciona uma resposta correta, silencia o dispositivo recetor e continua a pesquisar outros dispositivos. Uma vez que todos os dispositivos foram silenciados (nenhuma resposta é recebida para comandos de “*discovery*”), o processo é concluído e o controlador já possui uma lista dos dispositivos conectados [30].

Unicast Communication

A comunicação geral com um dispositivo específico ocorre num padrão de pedido-resposta. O controlador envia a solicitação ao dispositivo, abordando-o pelo UID do mesmo. Quando o pedido for enviado, o controlador “destrói” o controlo da linha DMX por um período de tempo determinado, de modo a que o dispositivo possa transmitir a sua resposta. A comunicação *unicast* é a única maneira pela qual os dados podem ser recuperados de um dispositivo (diferente do UID que pode ser obtido usando o mecanismo de *discovery* supramencionado). Se o dispositivo não responder dentro de um determinado período de tempo, o controlador pode assumir que a comunicação falhou e pode efetuar uma nova tentativa [30] [29].

Broadcast Communication

Para enviar instruções de forma célere para vários dispositivos elétricos, o RDM permite a comunicação de transmissão simultânea. Tal, permite que o controlador envie instruções para todos os dispositivos ou para todos os dispositivos do mesmo fabricante. Como, mais de um dispositivo pode estar a receber a mensagem, as respostas não são permitidas na comunicação de transmissão, exceto, claro, durante o processo de descoberta [30].

2.2.4 Topologia da rede

Os produtos DMX512 não suportam inerentemente o protocolo RDM. Somente os produtos RDM ‘*ready*’ ou ‘habilitados’ são capazes de tirar partido de todas as funcionalidades do padrão RDM. O padrão RDM requer apenas um subconjunto limitado das funções disponíveis no protocolo RDM, isto significa que alguns equipamentos que suportam o RDM podem não suportar todos os recursos desejados, o que obriga o utilizador a verificar junto do fabricante quais e como são aplicados os recursos de uma determinada máquina.

O mesmo ocorre com a unidade de controlo, onde existe uma ampla gama de controladores disponíveis, que podem detetar dispositivos e modificar as suas configurações, através de mesas de controlo de iluminação completas que integram a funcionalidade RDM na sua interface de controlo. Para tal este dispositivo não tem de ser uma mesa de iluminação principal, pode ser um controlador externo aplicado à saída ou em paralelo com a mesa de iluminação ou um simples computador. É importante referir que é essencial o uso de conversores de fins de linha e/ou divisores DMX compatíveis com RDM para poder separar o sistema DMX do subconjunto de máquinas RDM, estas *gateways* permitem a comunicação de duas vias de retorno ao controlador. A funcionalidade RDM varia de acordo com o produto, mas existe sempre uma interface para o modo de *discovery* dos dispositivos RDM [30].

2.2.5 Estado da arte

A utilização do protocolo RDM é cada vez mais uma opção na construção e desenvolvimento de cenários de iluminação em eventos onde as dimensões, a quantidade de máquinas e a mobilidade são essenciais nos principais procedimentos. Este protocolo procura mudar de forma radical a maneira como os *designers* de iluminação e luminotécnicos realizam a construção do seu *layout*, para além de aumentar a celeridade das operações de manutenção.

Para a verificação de parâmetros do protocolo, existem no mercado dispositivos de teste de comunicação DMX e RDM, que funcionam como ferramentas de trabalho para realizar pequenas verificações e alterações a uma determinada máquina ou até a uma rede completa conforme o dispositivo utilizado, sem a requisição de montagem e ligação da mesa de controlo RDM. A introdução destas ferramentas de teste aumentou consideravelmente a capacidade de projetar e avaliar controladores RDM e os seus recetores. Algumas empresas também apresentam dispositivos de injeção do protocolo RDM, que se colocam entre o controlador DMX e os recetores de forma a injetarem pacotes RDM no fluxo de dados DMX.

Cross-compatibility

Tal como acontece com qualquer protocolo relativamente novo, alguns problemas de compatibilidade entre fabricantes estão a ser detetados e tendem a ser amplificados. Para superar estes problemas, a comunidade DMX implementou várias ações, entre as quais, a mais expressiva é o fórum *RDM Protocol Developer* que pretende estabelecer a ponte entre os utilizadores e fabricantes de modo a obter um esclarecimento contínuo sobre determinadas ocorrências ou conflitos e a uniformidade do protocolo. O RDM depende fortemente de alguns pressupostos que podem afetar sua compatibilidade com outras tecnologias de iluminação.

A comunicação RDM depende da resposta dos dispositivos num determinado período de tempo. Num ambiente apenas para DMX, este facto, não é um problema, pois o atraso entre o dispositivo e o controlador provavelmente será muito curto.

No entanto se o DMX estiver assente num meio intermediário (como uma rede TCP / IP (*Ethernet*) ou interface sem fios), tal situação poderá causar alguns problemas. Em geral, se o fabricante possuir controlo sobre a interface intermediária, é possível reencaminhar as respostas RDM à medida que elas vão sendo recebidas, juntamente com um sistema *proxy* para o processo de *discovery* com o intuito de fornecer a ilusão da comunicação RDM.

Caso o fabricante não possua controlo sobre a implementação da interface intermediária (como acontece com a utilização de uma rede *Ethernet*), é praticamente impossível para o recetor enviar mensagens RDM de resposta para um controlador RDM baseado em DMX. No entanto, é possível manter a comunicação RDM com dispositivos baseados em DMX e um controlador baseado em *Ethernet*. Tal acontece, porque os controladores de iluminação já estão a evoluir na direção de serem baseados apenas em *Ethernet*. Ou seja, provavelmente, a comunicação RDM e DMX assente em *Ethernet*, será convertida por um dispositivo de saída de conversão *Ethernet-to-DMX* [30].

2.3 Art-Net

O *Artistic Network (Art-Net)* é um protocolo de comunicação livre, de transmissão de dados do protocolo DMX512-A e do protocolo RDM sobre o Protocolo de *Internet, User Datagram Protocol (UDP)*. Este é utilizado para a comunicação entre controladores dedicados DMX ou computadores com *software* dedicado, e ou entre estes e algumas máquinas ou dispositivos DMX mais complexos e inteligentes de iluminação profissional.

2.3.1 Definição

O *Art-Net* é uma implementação simples do protocolo DMX512-A sobre o UDP, na qual as informações de controlo da iluminação são transmitidas em pacotes IP, normalmente numa rede de área local privada, como *Ethernet*. Este permite que vários universos DMX sejam transportados através de um único cabo Cat5 ou superior, usando a tecnologia *Ethernet*. As funções suportadas incluem transmissão e receção de dados de iluminação, funções de gestão e parametrização de encaminhamento e funções de encaminhamento inteligente [31].

2.3.2 História

O *Art-Net* foi desenvolvido e patenteado pela *Artistic Licence Holdings Ltd*, em 1988, com o intuito de superar a restrição de canais do protocolo DMX, mantendo a estrutura do mesmo. Este protocolo sofreu três revisões, ao longo dos anos, no entanto estas versões são compatíveis entre si. Na versão original, designada *Art-Net I*, o protocolo estabelece uma comunicação *broadcast* em todas as suas transações, permitindo desta forma eliminar qualquer necessidade de configuração da rede por parte do utilizador e possibilitar o uso amplo de transmissões para aumentar o número de universos, até um limite de aproximadamente 40. No entanto, o desenvolvimento exponencial das máquinas e a sua requisição por maior necessidade de canais dedicados, obrigou a uma revisão do protocolo. Para resolver este problema, o *Art-Net II*, apresentado em 2006, aplica os mesmos conceitos do antecessor, mas implementa um algoritmo de identificação de percurso, ou seja, o controlador determina quais os nós que detêm o respetivo universo. Dessa forma, o controlador pode efetuar uma transmissão *unicast* para o respetivo nó, reduzindo a ocupação da largura de banda e aumentando o limite de universos para 256.

Este limite rapidamente foi esgotado, e a necessidade de o ampliar, levou a uma segunda revisão do protocolo. O *Art-Net III*, lançado em 2011, aborda os problemas na gestão de grandes números de universos, e através da implementação de um conceito designado *Binding*, que consiste na implementação de *gateways*, redefinindo o limite máximo até 32.768 universos.

Em 2016, o *Art-net IV*, a última versão deste protocolo, foi apresentado e supera o problema *multi-homing*, implementando um novo método de gestão de *gateways* que suportam múltiplas portas DMX. Este conceito, permite que uma *gateway* (ou qualquer produto *Art-Net*) suporte mais de 1000 portas DMX, tudo num único endereço IP. O conceito adicionado é 100% compatível com versões anteriores da *Art-Net* [31] [28].

2.3.3 Pacote de dados

O endereçamento é normalmente efetuado consoante o nó de comunicação, e muitas vezes bloqueado pelo endereço de máquina e um código *Original Equipment Manufacturer* (OEM) alocado ao fabricante. As redes de comunicação podem ser configuradas através do *Dynamic Host Configuration Protocol* (DHCP) ou endereços fixos inerentes ao *Internet Protocol* (IP) configurados e utilizar pacotes *unicast* para maior eficiência de rede. O protocolo pode abordar 32.768 universos DMX, cada um de 512 canais, limitado pela largura de banda. O endereçamento fixo pode ser problemático em redes com outros requisitos de endereçamento. O pacote de dados do protocolo Art-Net difere totalmente do pacote de dados DMX, uma vez que este é agora transmitido sobre o protocolo UDP.

Ou seja, a transmissão de dados DMX é agora efetuada com recurso a uma porta fixa UDP, normalmente a porta 0x1936 (6454 decimal). Abaixo, na Tabela 2.6 é representada a constituição de um pacote de dados Art-Net DMX.

Offset (Bytes)	0	1	2	3
0	'A'	'r'	't'	'_'
4	'N'	'e'	't'	0
8	Opcode ArtNet (0x5000)		Versão Protocolo	
12	Sequência	Porta Física	Universo	
16	Tamanho (2 to 512)		Data	Data
20	Data... (2 to 512)			

Tabela 2.6 – Formato do pacote de dados Art-Net DMX [31].

Os primeiros 8 *bytes* considerados, o cabeçalho do pacote, são invariáveis e identificam o tipo de pacote, ou seja, identificam o ArtNet.

Os *bytes* 8 e 9 representam o *opcode*, registos que indicam ao destinatário qual o pacote de dados DMX específico conforme o universo, respetivamente designados como '*OpcodeLo*' e '*OpcodeHi*'. Os dois *bytes* consequentes '*ProtVerHi*' e '*ProtVerLo*' indicam a versão do protocolo utilizado.

A sequência é um número compreendido entre 1 e 255, que permite ao destinatário reordenar os pacotes de dados enviados, de forma a endereçar a entrega fora de ordem (é utilizado o valor '0' para desativar este recurso).

O *byte* identificado como porta física indica o universo físico original da proveniência dos dados, se necessário. Os *bytes* 14 e 15 identificam o endereço de destino. Enquanto os *bytes* 16 e 17, representam respetivamente, o *byte* mais e o *byte* menos significativo do tamanho do *array* de dados. Os *bytes* consequentes representam os valores entre 0 e 255 dos canais ocupados.

Conceptualmente, um pacote de dados é transmitido para todos os nós constituintes da rede, no entanto só os dispositivos corretamente endereçados conforme o universo recebem os dados que lhe dizem respeito [31].

2.3.4 Estado da arte

O Art-Net 4 oferece uma nova e única característica que facilita a gestão da fonte de dados do *gateway*. Isto permite que os utilizadores escolham Art-Net como a ferramenta ideal para o mapeamento e gestão do protocolo RDM, enquanto usa *Streaming-ACN* (ANSI E1.31) para o controlo direto e em tempo real. No entanto, ao contrário do Art-Net, o *sACN* não suporta gestão de rede ou gestão do RDM, o que pode apresentar desvantagens.

Para tal o Art-Net 4 permite que outros protocolos sejam usados de forma complementar, além de permitir que os designers e luminotécnicos usufruam de um novo recurso de *software* que permite que todas as portas DMX sejam atribuídas a um universo totalmente independente. O Art-Net 4 já possui suporte integrado para a *VLC* (*Visible Light Communication*), uma tecnologia que provavelmente terá um enorme impacto nos próximos anos [31].

2.4 Rádio Frequência

A Rádio Frequência (RF) é um recurso tecnológico das telecomunicações utilizado para permitir a comunicação bidirecional por intermédio de *transceivers* (transmissores-recetores) de dados e informações previamente codificadas implementadas num sinal eletromagnético que se propaga através do espaço físico material e imaterial.

A Figura 2.12 diferencia o tipo de frequência conforme a largura de banda que ocupa ao longo do espectro eletromagnético.

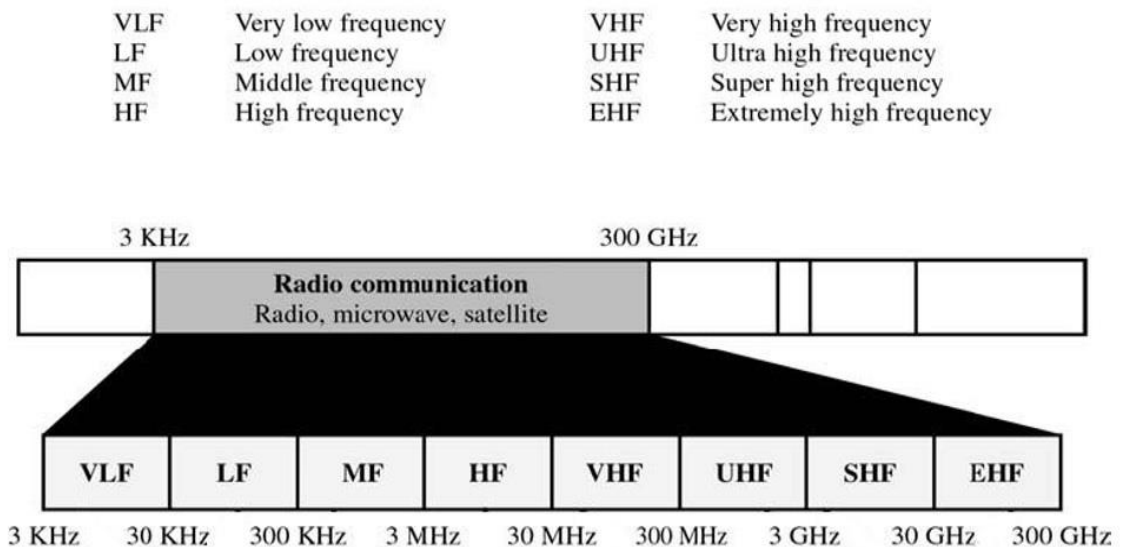


Figura 2.12 - Diagrama geral do espectro eletromagnético.

2.4.1 Definição

A Radiofrequência (RF) é a faixa do espectro de frequência que abrange aproximadamente 10 kHz a 300 GHz e que corresponde à frequência das ondas de rádio. As ondas de rádio são conhecidas por ondas de radiofrequência ou, simplesmente, radiofrequência. Estas ondas consistem em campos eletromagnéticos ativos que permitem o envio de sinal de um ponto para outro, sem a necessidade de fios.

A RF é um sinal que se propaga, normalmente por um condutor, habitualmente de cobre, e são irradiados no ar através de uma antena. Uma antena converte um sinal do meio elétrico cablado num sinal *wireless* (sem fio) e vice-versa. Os sinais irradiados no ar livre, em forma de ondas eletromagnéticas, propagam-se em todas as direções [32].

➤ Bandas de frequências ISM

As bandas de frequência *Industrial, Scientific and Medical* (ISM), são dedicadas a serviços industriais, científicos e médicos e impõe certas restrições técnicas na potência dos transmissores e na frequência que utilizam. Algumas destas são livres e permitem aos utilizadores e organizações o seu uso. Na Tabela 2.7 é representada uma lista de algumas das bandas ISM mais recorrentes.

Gama de frequências		Frequência	Aplicações
13.553 MHz	13.567 MHz	13.56 MHz	Serviços móveis
26.957 MHz	27.283 MHz	27.12 MHz	Serviços móveis e rádio CB
40.66 MHz	40.7 MHz	40.68 MHz	Serviços móveis e satélite
433.05 MHz	434.79 MHz	433.92 MHz	Serviços amadores e radiolocalização
2.4 GHz	2.5GHz	2.45 GHz	Serviços móveis, radiolocalização, amador e satélite
5.725 GHz	5.875 GHz	5.8 GHz	Satélite, radiolocalização, serviços móveis e satélite

Tabela 2.7 – Algumas das bandas ISM mais usuais segundo a ANACOM [33].

2.4.2 Estado da arte

Ao longo dos anos, a radiofrequência tem vindo a ser um dos recursos mais utilizados e explorados no desenvolvimento do setor das telecomunicações. O desenvolvimento deste setor passa pela criação e otimização de protocolos de comunicação e desenvolvimentos de dispositivos que permitam a implementação destes de forma a maximizar a sua funcionalidade, fiabilidade e segurança.

No entanto este recurso tecnológico, amplamente utilizado e explorado, começa a apresentar uma enorme saturação que obriga ao estudo e desenvolvimento de novas técnicas e protocolos de comunicação. Esta saturação de usufruto e a sua exploração massiva, levou a que nos últimos anos fossem investigados e estudados os efeitos potenciais da exposição à radiofrequência e aos campos magnéticos e de micro-ondas. Embora existam evidências de efeitos biológicos relacionados com a exposição aguda (aquecimento), os resultados da pesquisa sobre os efeitos relacionados à exposição crônica de baixo nível emitiram resultados contraditórios. Algumas revisões do estado da arte, apontam para a exigência de uma maior identificação e classificação dos tipos de comunicação, protocolos e respetivas larguras de bandas utilizadas, efetuando uma avaliação crítica do aproveitamento e respetivo desempenho do recurso tecnológico.

As limitações já reconhecidas da utilização de determinadas larguras de banda, potenciam o estudo e desenvolvimento de novas tecnologias dentro e fora da faixa de espectro da radiofrequência, entre as quais se destacam o enorme desenvolvimento das comunicações óticas sem fios, que apesar de já terem sido utilizadas muito antes da comunicação rádio, se encontravam ofuscadas pela preferência de transmissão rádio. Tal tem vindo a acontecer, pelo elevado custo temporal e económico no desenvolvimento de novas técnicas e tecnologias RF, e estes serem controlados por grandes entidades com enormes recursos tecnológicos e científicos. Deste modo, a radiofrequência tem sido ao longo destes últimos anos a tecnologia de excelência nas mais diversas áreas para efetuar a transmissão de dados entre dois ou mais pontos, devendo-se esta preferência à simplicidade de implementação, aos custos de infraestrutura, flexibilidade, fiabilidade e velocidades de transmissão das redes, comparativamente a outras tecnologias.

2.5 Ethernet

A *Ethernet* foi apresentada em 1975, dois anos após o início do seu desenvolvimento, e ainda hoje em dia, é uma das arquiteturas de comunicação mais utilizadas na comunicação de dados. Este protocolo de transmissão de dados, permite a criação de redes locais ou *Local Area Networks* (LAN's) e apresenta uma elevada utilização devido às suas inúmeras vantagens. Entre as quais se destacam a simplicidade, fiabilidade, flexibilidade, o custo reduzido de implementação e a elevada velocidade de comunicação. Este padrão, encontra-se ao abrigo da norma IEEE 802.3 que por sua vez especifica a camada física e a subcamada *Media Access Control* (MAC), definindo o protocolo *Carrier Sense Multiple Access with Collision Detection* (CSMA/CD) [34]. O CSMA/CD consiste no tipo de acesso à rede, onde o dispositivo que pretende aceder verifica se a mesma se encontra disponível. Se verificar a ocupação da rede, o equipamento efetua uma espera durante um tempo aleatório antes de repetir a verificação. Caso dois equipamentos iniciem a comunicação em simultâneo, será detetada a colisão, procedendo ambos à interrupção da comunicação e novamente à espera para retransmitir [35].

Deste modo, o padrão *Ethernet* implementa as duas primeiras camadas do modelo OSI, que consiste no modelo de referência para efetuar a comunicação entre computadores. Este modelo define, em 7 camadas, a estrutura de tarefas envolvidas na comunicação entre equipamentos, conforme representado na Figura 2.13.

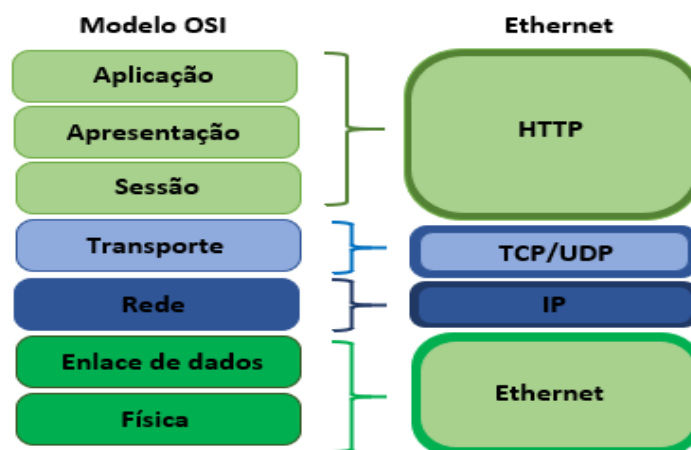


Figura 2.13 - Relação do protocolo *Ethernet* com o modelo OSI. Adaptado de [36].

A camada de enlace de dados (*Layer 2*) é responsável pelo “encapsulamento” de dados e pela gestão, da receção e envio das mesmas, controlando o fluxo de informação, detetando erros provenientes da camada física e estabelecendo um protocolo de comunicação com os sistemas diretamente conectados [37].

Na camada física são definidas as características físicas essenciais como a codificação e dados, tempos, níveis de tensão, entre outros. O encapsulamento dos dados resulta na seguinte estrutura presente na Figura 2.14:

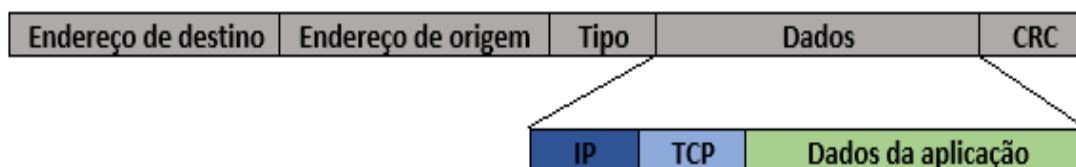


Figura 2.14 - Formato do pacote *Ethernet* na camada de ligação.

2.5.1 Internet Protocol (IP)

O Protocolo *Internet* encontra-se na terceira camada do modelo OSI, a camada de rede, e implementa o encaminhamento de pacotes recorrendo às informações de endereçamento e de controlo que possui. Conjuntamente com o TCP, este protocolo representa a base dos protocolos *Internet*.

O IP é um protocolo sem conexão e garantia de confiabilidade da integridade da entrega de pacotes. No entanto, protocolos de nível superior utilizam o IP para assegurar de que os pacotes são entregues ao endereço apropriado. Este fornece a fragmentação e montagem dos pacotes de dados de modo a permitir o envio de pacotes com tamanhos máximos diversos [38].

2.5.2 TCP/UDP

Os protocolos, *Transmission Control Protocol* (TCP) e o *User Datagram Protocol* (UDP) pertencem à quarta camada do modelo OSI (camada de transporte) e definem, em conjunção com a porta e IP do equipamento, como é transmitida a informação na rede.

A principal diferença entre estes dois protocolos é que o TCP, ao invés do UDP, garante que existe a entrega de todos os pacotes de dados, sendo por isso vantajoso na partilha de informação em sistemas críticos. Apesar do protocolo UDP não fornecer controlo de fluxo ou verificação de erros, é mais eficiente pois permite transportar uma grande quantidade de dados de forma célere.

2.5.3 Wi-Fi

Wi-Fi é a abreviatura de “*Wireless Fidelity*” e serve de designação a produtos que respeitam o conjunto de normas 802.11 criado pelo *Institute of Electrical and Electronic Engineers* (IEEE). O *Wi-Fi*, marca registrada da *Wi-Fi Alliance*, é uma tecnologia para redes de área local sem fios de suporte a dispositivos baseados nos padrões IEEE 802.11.

O *Wi-Fi* usa com frequência as bandas de rádio *Super High Frequency* (SHF) ISM de 2,4 GHz e 5,8 GHz.

Na Tabela 2.8, são referenciados os protocolos inerentes à camada física 802.11.

Tabela 2.8 - Protocolos inerentes à tecnologia Wi-Fi.

Protocolo	Publicação	Frequência (GHz)	Largura de Banda Máxima (MHz)	Velocidade máxima de transferência	Alcance (m)	
					Indoor	Outdoor
802.11	Junho 1997	2.4	22	0.24 MB/s	20	100
802.11a	Setembro 1999	5/3,7	20	6.44 MB/s	35/-	120/5000
802.11b	Setembro 1999	2.4	22	1.31 MB/s	35	140
802.11g	Junho 2003	2,4	20	17.88 MB/s	38	140
802.11n	Outubro 2009	2,4/5	40	103.32 MB/s	70	250
802.11ac	Dezembro 2013	5	160	6.44 MB/s	-	-
802.11ad	Dezembro 2012	60	2160	823.97 MB/s	-	-

A camada física 802.11 define diversos padrões de transmissão e codificação para comunicações sem fios sendo os mais comuns: *FHSS* (*Frequency Hopping Spread Spectrum*), *DSSS* (*Direct Sequence Spread Spectrum*) e *OFDM* (*Orthogonal Frequency Division Multiplexing*).

Entre outras características do projeto IEEE 802.11 pode-se citar a título de exemplo:

- Suporte de diversos canais;
- Sobreposição de diversas redes na mesma área do canal;
- Robustez em relação a interferência;
- Providência de mecanismos para evitar estações perdidas (*hidden nodes*);
- Oferecer privacidade e controlo no acesso ao meio.

De salientar, ainda, que este protocolo utiliza uma função de coordenação distribuída, denominada *Distributed Coordination Function* (*DCF*). Este modo, também conhecido como acesso com contenção, utiliza o protocolo abordado CSMA/CA no processo de deteção e prevenção de colisões, uma vez que estas são difíceis de detetar em ambientes sem fios [39].

2.6 SPI

Outro tipo de comunicação frequente na utilização da maioria dos microcontroladores é o *Serial Peripheral Interface* (SPI), este tipo de comunicação série síncrono possui uma topologia mestre – escravo de 4 sinais, sendo utilizado exponencialmente em sistemas embebidos. As aplicações comuns desta tecnologia são a comunicação entre o microcontrolador e dispositivos periféricos como sensores, cartões de memória, ecrãs, entre outros. Este permite a implementação de comunicações bidirecionais sendo o sinal *Master Out Slave In* (MOSI) referente à comunicação do mestre para o escravo e o sinal *Master In Slave Out* (MISO) o inverso. Neste protocolo a seleção do escravo com o qual o mestre da rede pretende comunicar é efetuada pelo sinal *Slave Select* (SS) sendo o mestre também responsável por gerar o sinal de relógio necessário para o sincronismo da comunicação (ver Figura 2.15).

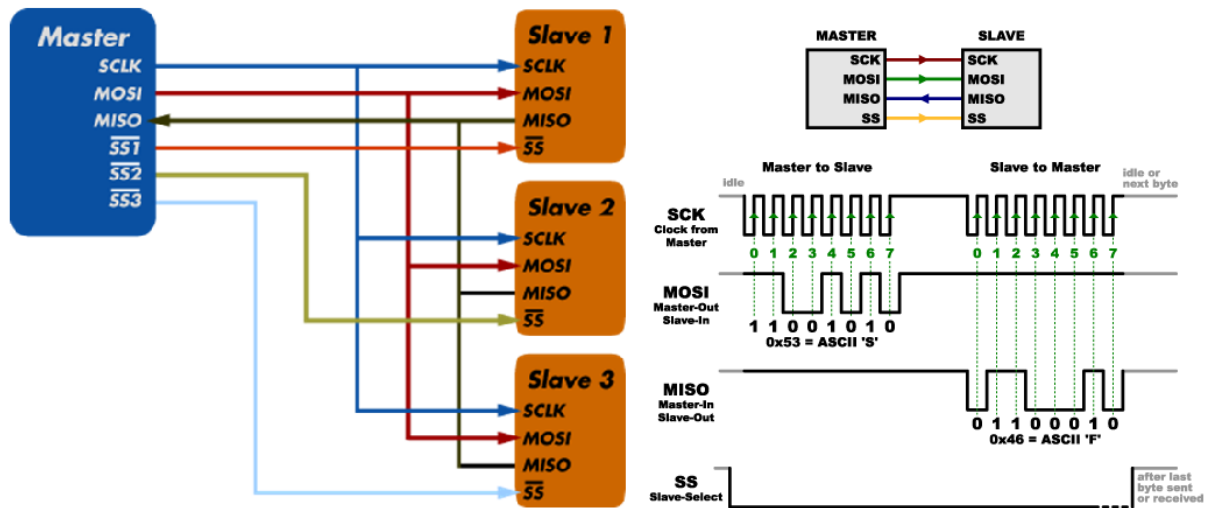


Figura 2.15 - Exemplo da topologia e comunicação da rede SPI. Adaptado de [40] e [41].

2.7 UART

O *Universal Asynchronous Receiver/Transmitter* (UART) consiste num padrão de comunicação série bidirecional assíncrono entre dois ou mais dispositivos, e encontra-se presente na maior parte dos microcontroladores. Para tal, cada dispositivo necessita efetuar a conversão de paralelo para série dos dados a enviar (*Tx*) e efetuar o processo inverso na receção (*Rx*). Esta conversão consiste em enviar a informação pretendida num *bit* de cada vez a um determinado débito pré-estabelecido entre o emissor e o recetor. O facto de ser uma comunicação assíncrona indica que não existe um sinal de relógio independente do sinal de dados, deste modo, para que haja sincronismo e que a mensagem seja interpretada pelo recetor são enviados dados adicionais no início de cada transmissão com o intuito de permitir o sincronismo por parte do recetor. Para tal, cada dispositivo deve possuir um relógio interno, e as velocidades devem ser configuradas de modo a serem iguais em ambos os dispositivos. Do mesmo modo, o fim da mensagem e o *bit* de paridade, podem ser adicionados no fim do pacote, sendo o fim da mensagem obrigatório [42].

Na Figura 2.16 encontra-se o exemplo de um pacote de dados série.

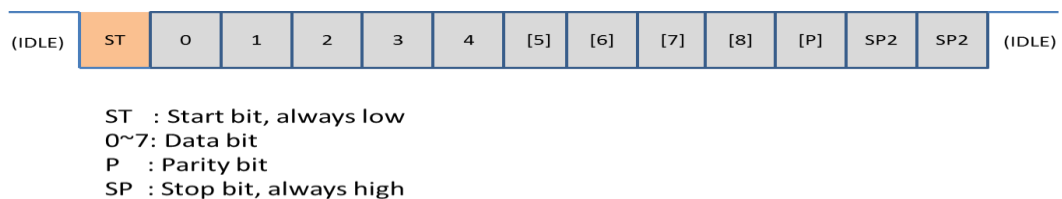


Figura 2.16 - Exemplo de um pacote de dados série. Adaptado de [43].

Este protocolo possui diversas vantagens, uma das mais significativas é a capacidade de adaptação deste tipo de portas de comunicação em microcontroladores para estabelecer uma comunicação direta e simples com computadores. Esta comunicação pode ser efetuada, emulando no computador uma porta RS-232 via *Universal Serial Bus* (USB). Para tal, basta converter, a porta série (UART) para o *standard* USB, recorrendo a conversores USB – UART (Figura 2.17) sendo interpretada pelo computador como uma porta série *standard* RS-232 [44].

Tal deve-se ao facto do protocolo RS-232 ter por base a comunicação UART idêntica à existente nos microcontroladores, diferindo apenas nas tensões de funcionamento [42].



Figura 2.17 - Conversor UART – USB.

2.8 Sumário

Neste capítulo foram referidas as tecnologias e equipamentos de maior relevo tendo em conta o projeto de investigação elaborado no âmbito da Comunicação DMX sem fios. Como tal, este capítulo, teve como intuito a introdução dos conceitos teóricos e a apresentação das diversas tecnologias disponíveis no mercado.

3 HARDWARE E SOFTWARE EM SISTEMAS EMBEBIDOS

Durante o desenvolvimento deste projeto, foi claro que a seleção, utilização e a capacidade de gestão de diversos equipamentos e tecnologias, iria ser a tarefa mais exigente e demorada. Por norma, estes sistemas dependem do desenvolvimento de *hardware* e *software* aprimorado ao objetivo. O estudo e o desenvolvimento destes programas ou ferramentas são indispensáveis ao avanço de projetos, permitindo um progresso eficiente a todos os níveis. Estas ferramentas de desenvolvimento podem ser dedicadas à programação e simulação dos microcontroladores, à simulação dos circuitos eletrónicos, ao desenvolvimento de placas de circuito impresso (PCIs), entre outras.

O sistema embebido a desenvolver deve requerer de diversos subsistemas e componentes de forma a permitir receber, processar e reencaminhar uma comunicação DMX. Esta deve progredir do meio físico típico para um meio sem fios, e garantir ao utilizador uma utilização fácil, versátil e otimizada do sistema. Na sequência do exposto, foi elaborado o diagrama simplificado demonstrado na Figura 3.1.

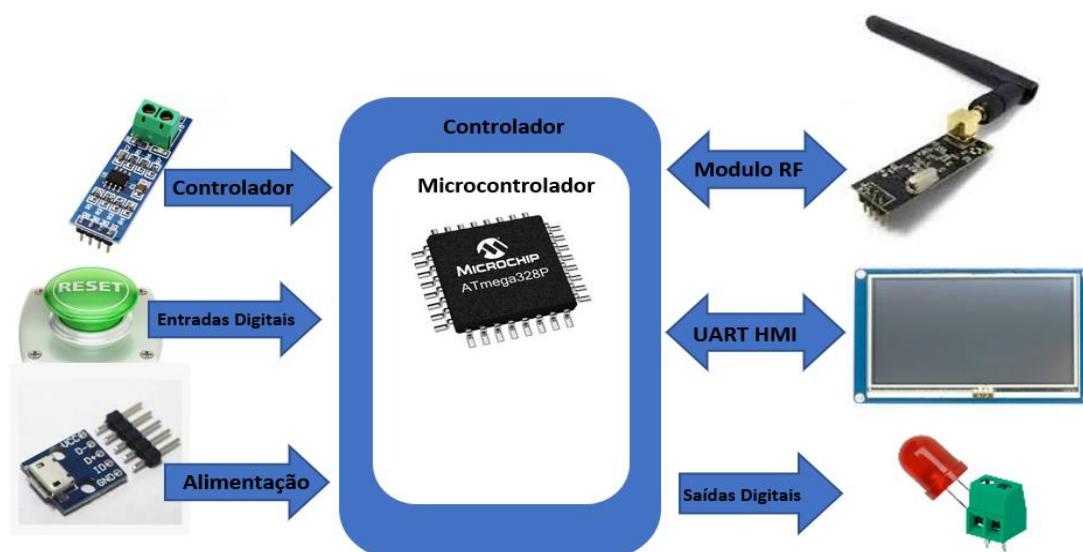


Figura 3.1 - Diagrama de blocos do sistema embebido a desenvolver.

Este capítulo efetua uma descrição do *hardware* selecionado, dos sistemas embebidos e das ferramentas de desenvolvimento utilizadas na realização deste projeto.

3.1 Sistemas embebidos

Um sistema embebido é um sistema dedicado para realizar uma determinada tarefa que pode conter ou não, um microprocessador. Este tipo de sistema procede à realização de um conjunto de tarefas predefinidas, geralmente com requisitos específicos. Já que o sistema é dedicado a tarefas específicas, através de engenharia, é possível otimizar o projeto, reduzindo variáveis como o tamanho, recursos computacionais e, conseqüentemente, o tempo de desenvolvimento e o custo do produto. Na vida quotidiana, a interação com este tipo de sistemas é constante e, não raras vezes, passam despercebidos.

Estes sistemas encontram-se presentes em relógios, *gadgets*, automóveis, *smartphones*, consolas de jogos, entre muitos outros. Apresentam na sua constituição dois elementos principais sendo estes o *hardware* e o *firmware* [45]. O *hardware* é elemento composto por todos os componentes físicos necessários para o funcionamento do sistema englobando, os circuitos integrados, componentes discretos, placas de circuito impresso, alimentação e todos os restantes constituintes materiais. O *firmware* consiste no conjunto de instruções e de dados que permite a realização, por parte do microcontrolador, das tarefas a desempenhar pelo sistema, distinguindo-se do termo *software* pelo facto de desempenhar operações de mais baixo nível, ou seja, diretamente com o *hardware*.

As telecomunicações, comunicações sem fios, fazem cada vez mais parte dos sistemas embebidos, sendo que estes permitem a transferência de dados e informações em curtas ou longas distâncias, conforme a tecnologia e o *hardware* utilizado, sem recorrerem à utilização de cabos. Nos dias de hoje algumas das tecnologias mais comuns de comunicações sem fios são o *Long Term Evolution (LTE)*, *LTE-Advanced*, *Wi-Fi*, *Bluetooth*, entre outras. O mundo contemporâneo utiliza as comunicações sem fios numa grande diversidade de dispositivos e contextos. Entre estes usos podemos citar as redes sem fios, os telemóveis e *smartphones*, *tablets* e *surfaces*, dispositivos GPS, controladores e atuadores dedicados, a televisão por satélite artificial, entre outros.

Estes sistemas apresentam diversas vantagens, como:

- ✓ Qualquer dado e informação pode ser transmitida rapidamente e com alta velocidade;
- ✓ Manutenção e instalação de baixo custo;
- ✓ Possibilidade de acesso em qualquer lugar, desde que tenha sinal sem fios;
- ✓ Maior grau de mobilidade durante a comunicação;
- ✓ Capacidade de interligação de um maior número de dispositivos;
- ✓ Versatilidade da comunicação.

3.2 Microcontroladores

Um microcontrolador consiste num circuito eletrónico que para além das funções existentes num microprocessador inclui outras funções e periféricos (memória, funções de entrada e saída, entre outras). É constituído por um núcleo de processamento, memórias, temporizadores e entradas e saídas. Estas entradas e saídas (*IO's*) podem ser digitais, analógicas e, podem ainda, disponibilizar outras funções específicas como, interrupções, ou portas de comunicação. Estes dispositivos programáveis foram desenvolvidos para aplicação em sistemas embebidos, estando presentes na maioria dos equipamentos eletrónicos [45]. Ao nível do mercado, encontram-se disponíveis com uma grande diversidade de especificações e preços.

A placa de desenvolvimento Arduino Uno, foi a placa utilizada para desenvolver os primeiros esboços deste projeto, tal como apresentado na Figura 3.2. O microcontrolador *ATMEGA328P-PU* é o componente principal da placa Arduino Uno, um dispositivo de 8 *bits* da família AVR da Atmel, com arquitetura RISC e encapsulamento DIP28.

Este microcontrolador possui 32 KB de memória *Flash*, dos quais, 512 *Bytes* são utilizados pelo *bootloader*, 2 KB de memória RAM e 1 KB de EEPROM.

Esta possui 4 canais de *Analog to Digital Conversor* ADC de 10 bits, 5 pinos PWM, 12 pinos de entrada/saídas digitais, entre os quais, os terminais de uma porta UART. Esta requer de um *bootloader* de 16MHz, valor do cristal externo, interligado internamente, aos pinos 9 e 10 do microcontrolador *Atmel*, mas pode operar a uma frequência de até 20 MHz (ver Figura 3.2).

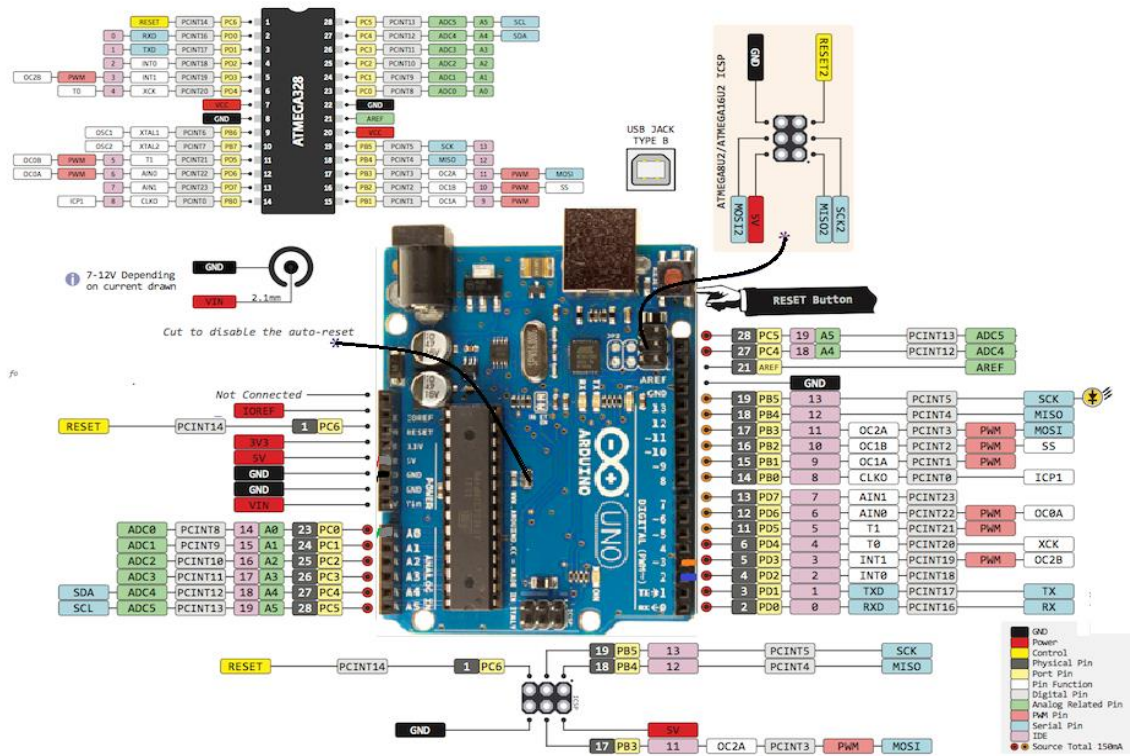


Figura 3.2 - Placa de desenvolvimento Arduino Uno. Adaptado de [46].

Numa segunda fase do projeto, foi utilizado o microcontrolador ATmega328P-AU, de forma a minimizar o tamanho da placa PCB criada. A placa Arduino Pro Mini possui todos os atributos do microcontrolador anterior, à exceção do tipo de encapsulamento (TQFP32) e as suas dimensões reduzidas, como pode ser visível na Figura 3.3 abaixo apresentada.

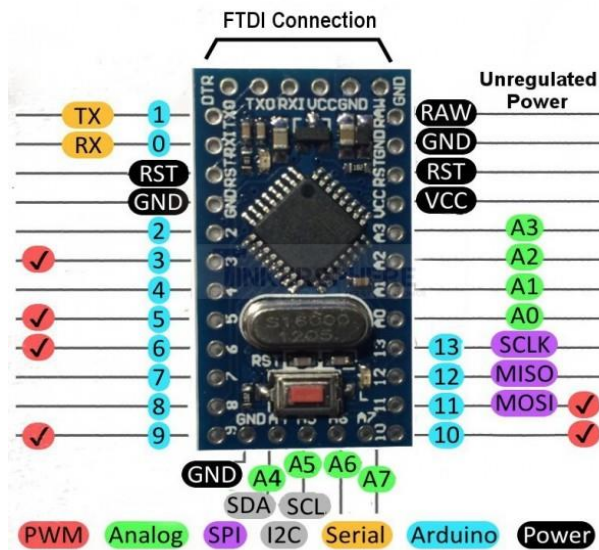


Figura 3.3 - Placa de desenvolvimento Arduino Pro Mini [46].

Na Tabela 3.1, encontram-se expostas as especificações mais relevantes do microcontrolador utilizado na realização deste projeto.

Tabela 3.1 - Especificações do microcontrolador ATmega328P [47].

Característica	Descrição
Referência	ATmega328P
Fabricante	Atmel
Arquitetura	8-bit AVR
Memória <i>flash</i>	32KB
Memória RAM	2KB
Número de pinos	32
Número de I/O's	23
Número de ADC's	6 (10-bit)
Frequência de relógio máxima	20MHz
Tipos de <i>interface</i>	I2C, SPI, UART
Url do fabricante	www.atmel.com

3.3 Módulo de rádio frequência nRF24L01

O nRF24L01 é um módulo *transceiver* de custo muito reduzido fornecido com um protocolo de banda base embecida (*ShockBurst*), adequado para aplicações *wireless*. Produzido pela *Nordic Semiconductor*, este módulo vem equipado com uma antena embutida que opera numa gama de frequências ISM de 2.400 GHz a 2.525GHz, com uma velocidade máxima de operação de 2 Mbps, um alcance que pode chegar aos 10 metros em ambiente interiores e 50 metros em ambientes exteriores, respetivamente. O módulo vem equipado com 8 pinos de interligação, tal como demonstrado na Figura 3.4.

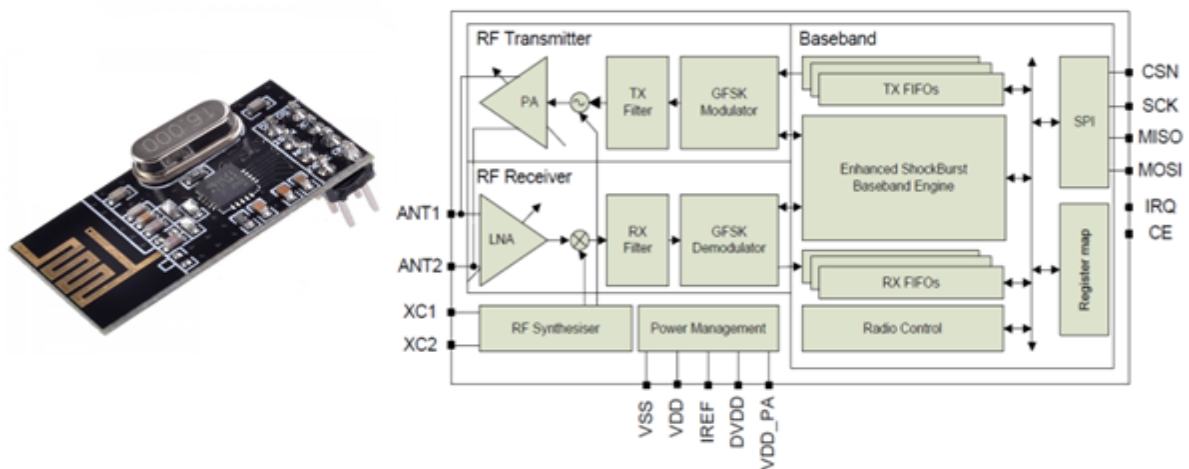


Figura 3.4 - Módulo nRF24L01 e respetivo diagrama de blocos [48].

Na sequência do exposto, este módulo, que é controlado pelo integrado NRF24L01 da *Nordic*, caracteriza-se pelo baixo consumo de energia e pela alta velocidade de comunicação, que pode chegar a 2 Mbps.

Devido ao uso da interface SPI, é possível interligar este módulo com a maioria dos microcontroladores disponíveis atualmente. Uma outra grande vantagem, é a capacidade deste módulo, poder atuar como emissor, recetor ou *transceiver*, dependendo apenas da configuração de *software*.

A sua tensão de alimentação pode variar entre 1,9 V a 3,6 V e os pinos de sinal podem trabalhar normalmente com nível de sinal de 5 V, utiliza modulação GFSK, capacidade anti interferência, verificação de erros, comunicação multiponto de 126 canais e controlo de fluxo.

Uma das características mais interessantes do nRF24L01 é o suporte designado por *Enhanced Shockburst*, este permite a criação de, até 6 *links* bidirecionais, com um tratamento em *hardware* para envio de confirmações de receção (*Acknowledge*, ACK), e *Cyclic Redundancy Check* (CRC) e retransmissão automática em caso de falha [48]. Na Tabela 3.2 é representado o formato do pacote de dados do módulo nRF24L01.

Tabela 3.2 - Formato do pacote de dados *Enhanced Shockburst* [48].

Preâmbulo	Endereço	Controlo do pacote de dados	Pacote	CRC
1 <i>byte</i>	3 a 5 <i>bytes</i>	9 <i>bits</i>	0 a 32 <i>bytes</i>	1 a 2 <i>bytes</i>

3.3.1 Propriedades elétricas e características do módulo

Este módulo, apresentado na Figura 3.5, possui reguladores de tensão internos que asseguram um alto *Power Supply Rejection Rádio (PSRR)* e uma ampla tensão de alimentação.

Também na Figura 3.5, é possível, identificar a distribuição dos pinos de ligação do módulo no lado esquerdo, e visualizar a antena embutida no lado direito.

O tipo de ficha do módulo impede a aplicação direta na *protoboard*, portanto é recomendada a utilização de um adaptador para *protoboard* ou a utilização de outro tipo de ligação, como o uso de *jumpers* macho-fêmea [48].



Figura 3.5 - Módulo nRF24L01 e respetiva legenda da identificação de pinos. Adaptado de [49].

Em seguida apresenta-se a Tabela 3.3 com as principais características do módulo nRF24L01.

Tabela 3.3 - Especificações do módulo radiofrequência nRF24L01 [48].

Característica	Descrição
Referência	nRF24L01
Fabricante	<i>NORDIC SEMICONDUCTOR</i>
Tensão de alimentação	1,9 a 3,6 V
Corrente de alimentação	1 μ A
Número de pinos	8 unidades
Velocidade de operação	2 Mbps
Modulação	GFSK
Comunicação	Multiponto 125 canais
Dimensões	29 x 15 x 5 mm
Frequência de relógio	2,4 GHz
Taxa de Transmissão	+7 Db
Sensibilidade de Recepção	\leq -90dB
Tipos de <i>interface</i>	SPI
Url do fabricante	http://www.nordicsemi.com/

3.3.2 Estado da arte

Ao nível dos métodos de manipulação de pacotes de dados, o módulo nRF24L01 apresenta duas funcionalidades dedicadas:

- *ShockBurst*™ (compatível com nRF2401, nRF24E1, nRF2402 e nRF24E2, a 1 Mbps);
- *Enhanced ShockBurst*™.

O *ShockBurst* permite usufruir de um alto débito de transmissão e recepção de dados, uma vez que este é o responsável pela manipulação de pacotes de dados, desta forma, o nRF24L01 não requer, nem depende de qualquer recurso imposto pelo controlador. Ou seja, esta funcionalidade permite o envio e recepção de pacotes de dados de forma mais rápida e eficiente, sem a necessidade de um alto desempenho por parte do microcontrolador (MCU), para a gestão de pacotes de dados.

Ao efetuar todo o processamento de sinal *on-chip*, o nRF24L01 oferece ao microcontrolador uma interface SPI simples e compatível com a velocidade de processamento adequada à maioria dos microcontroladores. Tal facto, permite a maximização da velocidade de processamento de dados no *link* RF, permitindo a gestão e redução do consumo médio do sistema. Esta funcionalidade permite em modo de recepção, a notificação do MCU por parte do IRQ, quando um endereço válido e um pacote útil são recebidos, respetivamente.

No entanto, em modo de transmissão, o *ShockBurst* gera automaticamente o preâmbulo e o CRC, enquanto o IRQ notifica o microcontrolador de que a transmissão está concluída [48].

Desta forma, esta funcionalidade diminui a demanda de recursos de memória do microcontrolador, permitindo uma aquisição de menor custo e a redução do tempo de desenvolvimento de programação. O nRF24L01 possui um *buffer FIFO* de recepção de três níveis (compartilhado entre 6 *pipes*) e um *buffer FIFO* de transmissão de três níveis. Desta forma, o microcontrolador pode ter acesso aos *buffers FIFO* a qualquer momento, em *power down mode*, em *standby mode* e durante a transmissão de pacotes de dados.

O *Enhanced ShockBurst*™ é um método de manipulação de pacotes com funcionalidade bidirecional, fácil e eficiente. Esta função é responsável pela gestão dos *pipelines* de comunicação e pela deteção e validação de dados, recorrendo CRC.

Esta funcionalidade vai permitir ao nRF24L01 lidar com o reconhecimento de pacotes recebidos e retransmissões de pacotes perdidos, sem envolvimento do microcontrolador, caso a função *Acknowledge* esteja ativa [48]. Uma otimização deste módulo é o módulo nRF2401+, este admite extra *pipelines*, extra *buffer*, e ainda uma função de auto retransmissão. Este módulo disponibiliza um amplificador de potência e um circuito LNA interno, que permite obter ainda maior alcance na comunicação. Estas melhorias, em conjunto, com a implementação de uma antena externa de 2.4G (2DB), com uma capacidade de transmissão de 250Kbps em ambiente aberto, permitem obter um alcance máximo compreendido entre 800 a 1000 metros de distância [50].

Na Figura 3.6, é possível identificar o módulo nRF24L01+ e respetiva antena, e verificar que as alterações ao nível de *hardware* se centram no suporte complementar a esta.

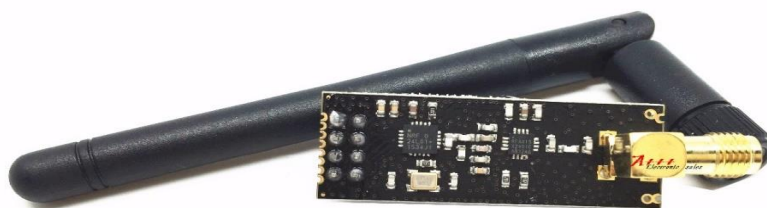


Figura 3.6 - Módulo nRF24L01+ e respetiva antena [50].

De salientar que ambos os modelos do módulo nRF24L01 possuem um pino de interrupção, *Interrupt Request* (IRQ). O pino IRQ é ativado quando os registos TX_DS IRQ, RX_DR IRQ ou MAX_RT IRQ se encontram ao nível lógico '1' no registo STATUS [48] [50].

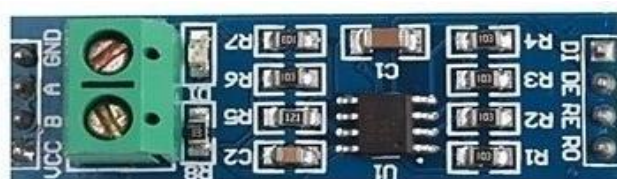
Em seguida apresenta-se a Tabela 3.4 com as principais características do módulo nRF24L01+.

Tabela 3.4 - Especificações do módulo de radiofrequência nRF24L01+ [50].

Característica	Descrição
Referência	nRF24L01+
Fabricante	<i>NORDIC SEMICONDUCTOR</i>
Tensão de alimentação	3 a 3,6 V
Corrente de alimentação mínima	4,2 μ A
Corrente de pico em modo emissor	115 mA
Corrente de pico em modo recetor	45 mA
Figura de Ruído do LNA	2.6 Db
Ganho do LNA	10 Db
Ganho do PA	20 Db
Sensibilidade no modo recepção a 250 Kbps	104 dBm
Ganho de pico da antena	2 Dbi
Potência máxima de saída	+ 20 dBm
Número de pinos	8 unidades
Velocidade de operação	2 Mbps
Modulação	GFSK
Comunicação	Multiponto 126 canais
Dimensões	46 x 17 x 12 mm
Frequência de relógio	2,4 GHz
Taxa de Transmissão	+7 Db
Sensibilidade de Recepção	\leq -90Db
Tipos de <i>interface</i>	SPI
Url do fabricante	http://www.nordicsemi.com/

3.4 Módulos de RS-485

O módulo *RS-485* da *LC Technology*, representado na Figura 3.7, funciona com base no *chip MAX485*, sendo que este é caracterizado por ser um *transceiver* de baixa potência e com uma velocidade de resposta limitada, podendo alcançar uma velocidade máxima de transmissão de apenas 2.5 Mbps. O módulo é utilizado essencialmente para realizar comunicações *RS-485* uma vez que, interligado a um microcontrolador, permite converter os sinais série TTL em *RS-485* e admite a implementação de várias unidades no mesmo barramento, (*Multi-Drop*) [51].

Figura 3.7 - Módulo *RS-485* da *LC Technology* [52].

Como já foi referido anteriormente o *RS-485* é um protocolo simples, robusto e, ao contrário de outras tecnologias proprietárias, a sua utilização é gratuita. Por estas razões, muitos sensores e dispositivos atuadores adotaram a sua utilização como uma forma de comunicação, o que ainda prevalece na indústria.

A ligação dos módulos *RS-485* é simples, uma vez que se trata de um *shield*. As principais preocupações devem centrar-se na ligação dos pinos de alimentação do módulo e dos condutores A e B do par entrançado do barramento *RS-485*.

Este módulo permite a configuração como transmissor ou recetor, através da utilização dos pinos RE (*Receptor Enable*) e DE (*Driver Enable*), respetivamente. Caso estes pinos estejam ligados ao pino Vcc, o módulo irá atuar como emissor, ligados ao pino GND atuará como recetor. Finalmente, tem de se proceder à ligação, respetivamente, para o módulo de entrada de dados DI (*Digital Input*) no caso de servir como o emissor, ou módulo de RO (*Receptor Output*) de dados de saída no caso de servir como recetor [53].

Portanto, no caso de utilização da porta UART, o esquema de ligação do módulo em modo emissor é apresentado na Figura 3.8.



Figura 3.8 - Esquema de ligação do módulo RS485 em modo emissor [53].

Ao invés o circuito do modo de funcionamento como recetor é apresentado na Figura 3.9.



Figura 3.9 - Esquema de ligação do módulo RS-485 em modo recetor [53].

Na Tabela 3.5, encontram-se descritas as especificações do módulo RS-485.

Tabela 3.5 - Especificações do módulo RS-485 [51].

Característica	Descrição
Referência	LC RS-485
Fabricante	LC TECHNOLOGY
Tensão de alimentação	5 V
Corrente de alimentação	120uA~500Ma
Consumo medio	830Mw
Número de pinos	8+2
Dimensões	44 x 14 x 15 mm
Velocidade de operação	≤10 Mbps
Tipos de interface	SPI
Url do fabricante	http://www.chinalctech.com

3.5 Human Machine Interface (HMI)

O *HMI* Nextion consiste num *display* LCD tátil que fornece uma solução de controlo e visualização a qualquer microcontrolador ou dispositivo que possua uma porta UART, uma vez que este possui uma interface de 4 pinos para estabelecer comunicação com qualquer mestre série *TTL*.

Este dispositivo é principalmente indicado para aplicações *IoT* (*Internet of things*) e apresenta-se como uma das melhores soluções de mercado, uma vez que se trata de um dispositivo inteligente e conta com uma plataforma de programação (Figura 3.10) [54].

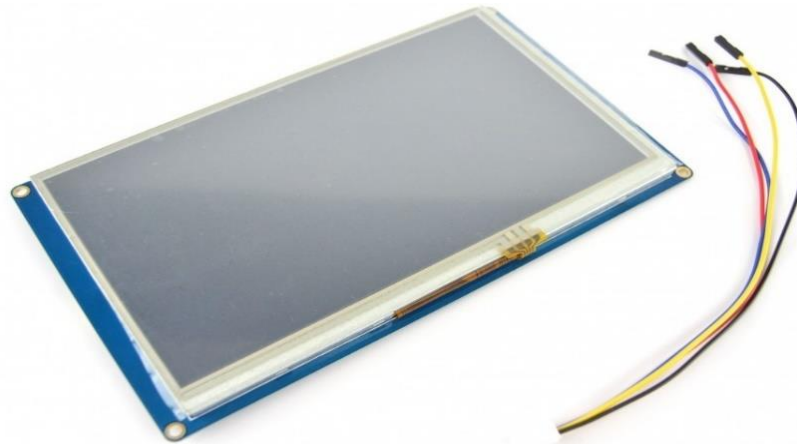


Figura 3.10 - *HMI* Nextion 7 polegadas [54].

O *HMI* Nextion é um equipamento composto por uma parte de *hardware* e uma parte complementar de *software*, sendo que para a programação do ambiente gráfico deste, é disponibilizado um editor de acesso gratuito (*Nextion Editor*). A descrição deste *software* e a sua programação podem ser consultados adiante.

Este dispositivo permite o desenvolvimento de ambiente gráfico e a sua programação é aplicada diretamente no controlador da aplicação, o que faz com que este seja um dispositivo bastante versátil, e capaz de ser inserido em qualquer sistema embebido de forma de veras acessível [54].

Este, possui uma enorme diferença comparativamente com outras interfaces visto efetuar o armazenamento e processamento de toda a informação, minimizando a utilização de recursos por parte do microcontrolador.

Deste modo, a troca de informação entre estes dispositivos é reduzida apenas comandos e troca de variáveis. Esta interface é efetuada recorrendo à porta UART presente no HMI que dispõe dos pinos de transmissão (TX) e receção (RX) assim como os pinos de alimentação (5V e GND).

Além disto, encontra-se também presente uma *slot* para cartões de memória microSD, para armazenamento de dados e atualização de *firmware*.

Na Tabela 3.6 encontram-se definidas as características gerais do HMI.

Tabela 3.6 - Especificações do HMI [54].

Característica	Descrição
Referência	NX8048T070_011R
Fabricante	NEXTION
Tamanho do <i>Display</i>	7.0"
Resolução do <i>Display</i>	800x480 <i>pixel</i>
Área Visual	95.04 × 53.86mm
Cor	RGB 64K, 65536 cores, 16 bits 565, 5R-6G-5B
Brilho ajustável	0 ~ 230 nit, com intervalo de ajuste de 1%
Capacidade <i>multi-touch</i>	Sim
Tipo de <i>touch screen</i>	Resistivo
Controlador dedicado	Não
Tensão de alimentação	5 V DC
Corrente de alimentação	2,0A
Corrente de pico (Brilho 100%)	510 mA
Corrente de mínima	15 mA
Consumo medio	250mA a 5V
Número de pinos	4
Dimensões	181 x 108x 09 mm
Peso	268g
Velocidade de operação	≤115.2 Mbps
Memória <i>Flash</i>	16 MB
Memória <i>RAM</i>	3584 <i>Bytes</i>
Memória expansiva	Micro SD com capacidade máxima de 32GB
Tipos de <i>interface</i>	UART, SPI
Url do fabricante	https://nextion.itead.cc/

3.6 Desenvolvimento de placas de circuito impresso

Uma placa de circuito impresso consiste numa placa isolante, regularmente de fenolite, fibra de vidro, fibra de poliéster ou outro tipo de polímero, que possuem, uma ou mais camadas finas de cobre isoladas entre si, de forma a implementar o circuito elétrico projetado, onde são soldados os diversos componentes, constituindo as ligações entre eles.

Os circuitos impressos são usados em quase todos os produtos eletrônicos e requerem conhecimento e esforço adicional no *design* para estabelecer o circuito, não obstante a sua produção e montagem podem ser automatizadas.

O desenvolvimento de uma placa de circuito impresso começa pela elaboração do respetivo esquema elétrico, tendo em conta os componentes que se pretendem usar, isto é, devem ser utilizados os modelos que possuam as mesmas dimensões dos componentes reais a utilizar, que no caso dos circuitos integrados corresponde ao tipo de encapsulamento.

No desenvolvimento deste projeto foram utilizados os recursos do *software* de desenho assistido de placas de circuito impresso, *EAGLE (Easy Applicable Graphical Layout Editor)* [55].

O *software EAGLE* criado pela empresa *Cadsoft*, e recentemente adquirido pela *AutoDesk*, é uma ferramenta de simples interação e manuseamento, que permite um desenvolvimento acessível e rápido de PCB's. Este *software* de desenvolvimento disponibiliza uma ferramenta de criação automática das pistas e possui o modelo de inúmeros componentes disponíveis através da utilização das suas bibliotecas.

Na página oficial, é possível também encontrar manuais, tutoriais e outras bibliotecas para os mais variados componentes e módulos, bem como, uma versão gratuita desta aplicação, que permite o desenvolvimento de placas com dimensões até 100x80mm, requerendo apenas o registo no *site* [55]. Para um correto desenvolvimento de placas de circuito impresso é essencial ter em consideração algumas regras presentes no *Standard IPC-2221*, sendo as mais relevantes descritas neste subcapítulo.

Na Figura 3.11 é possível observar os diversos constituintes do circuito, encontrando-se cada um dos nós associados a uma referência. A atribuição destas referências permite a interligação dos diversos componentes de cada um dos nós do circuito, simplificando o desenvolvimento e a perceção do circuito elétrico. Depois da criação e verificação do esquema elétrico da placa, procede-se ao desenvolvimento do desenho da mesma, sendo esta concebida já com os respetivos componentes e ligações elétricas [55].

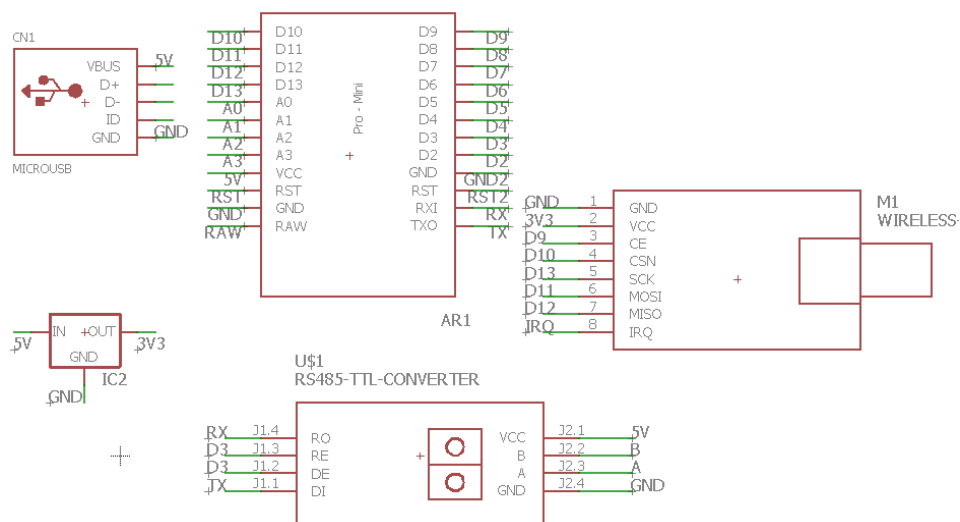


Figura 3.11 - Esquema *Eagle* da placa PCB desenvolvida para o protótipo DMX sem fios.

O posicionamento dos componentes deve ser efetuado de forma a garantir o afastamento dos circuitos de potência dos restantes, com o intuito de minimizar as interferências. Outro aspeto importante a ter em consideração na distribuição dos componentes é a passagem das respetivas pistas devendo ser, sempre que possível, colocados os componentes a interligar a distâncias reduzidas, com o objetivo de tornar possível uma passagem direta das mesmas.

Na sequência do exposto são descritos de seguida os aspetos mais importantes a ter em conta no desenvolvimento eficiente de PCI's [56] [57] .

- Evitar ângulos retos no desenho das pistas de forma a evitar a ocorrência de reflexões, minimizando o comprimento e a sua capacitância;
- Garantir algum espaço na distribuição dos componentes, de forma a permitir uma melhor dissipação térmica e simplificação na montagem;
- Colocar os condensadores de desacoplamento nas alimentações e nos circuitos integrados, seguindo as suas especificações técnicas;
- Desenhar zonas de restrição de vias e pistas, permitindo que o roteador automático implemente estas limitações e, subsequentemente que detete e identifique falhas;
- Afastar as pistas que contenham sinais sensíveis a interferências eletromagnéticas, tais como os sinais analógicos das pistas de potência;
- Proceder ao posicionamento dos conectores para interligação com o exterior na periferia da placa, de modo a facilitar as respetivas ligações;
- Consultar as especificações dos fabricantes dos circuitos integrados e verificar as recomendações em termos de posicionamento, dissipação de calor, parâmetros de alimentação e necessidades de circuitos externos auxiliares;
- Verificar a largura necessária para as pistas que requeiram a passagem de correntes mais significativas ou, em alternativa, optar pelo reforço posterior das mesmas;
- Proceder a uma verificação completa da placa projetada, sobretudo no que diz respeito ao afastamento entre pistas e às zonas para soldagem dos diversos componentes.

3.7 Programação dos microcontroladores

A programação de um microcontrolador consiste na criação de uma estrutura lógica, geralmente em linguagem C/C++, das instruções que permitem ao mesmo desempenhar as tarefas a que se propõe. De seguida, é necessário efetuar a compilação e a transferência do programa desenvolvido, do compilador para a memória *Flash* do microcontrolador.

O compilador consiste numa aplicação que fornece um ambiente de desenvolvimento, teste e depuração de programas, onde é gerado um ficheiro que contém as instruções a serem executadas pelo microcontrolador.

A transferência do programa para o microcontrolador é usualmente feita recorrendo a um *software* específico. Este varia de acordo com o fabricante e o modelo a programar. Por fim, o microcontrolador ficará a executar as tarefas definidas, permitindo, de acordo com o restante *hardware*, desempenhar as funções a que se destina [58].

Com o intuito de facilitar o desenvolvimento destas aplicações, encontram-se disponíveis ambientes de desenvolvimento integrados, ou IDE, que disponibilizam todas as ferramentas necessárias nas várias etapas.

Estas aplicações, além de permitirem o desenvolvimento, a verificação de erros, a compilação e transferência do código, disponibilizam ainda meios para testar o funcionamento do programa em tempo real [59]. Na sequência do exposto será efetuada uma breve descrição ao IDE utilizado no presente projeto. O Arduíno consiste numa plataforma de desenvolvimento de protótipos eletrónicos, *Open-Source* (Código Aberto).

Possui uma estrutura de elementos periféricos que permite um desenvolvimento simplificado e célere quer ao nível de *hardware*, quer ao nível de *software*.

Uma placa Arduino é composta por um controlador, neste caso, um microcontrolador *Atmel AVR*, que suporta I/O digitais e analógicas, uma interface serial ou USB, para estabelecer comunicação com o dispositivo de programação e controlo, utilizando uma linguagem de programação padrão (bastante próxima da linguagem C/C++).

A maioria das placas desenvolvidas por este fabricante inclui o conversor UART-USB, necessário para efetuar o envio do programa e para implementar as trocas de informação entre o computador e o microcontrolador. Na Figura 3.12 é possível observar o ambiente de desenvolvimento Arduino, que possui uma *interface* acessível ao utilizador, encontrando-se, todas as ferramentas necessárias. Para as aplicações mais específicas, encontram-se disponíveis inúmeras bibliotecas com os respetivos exemplos de aplicação.

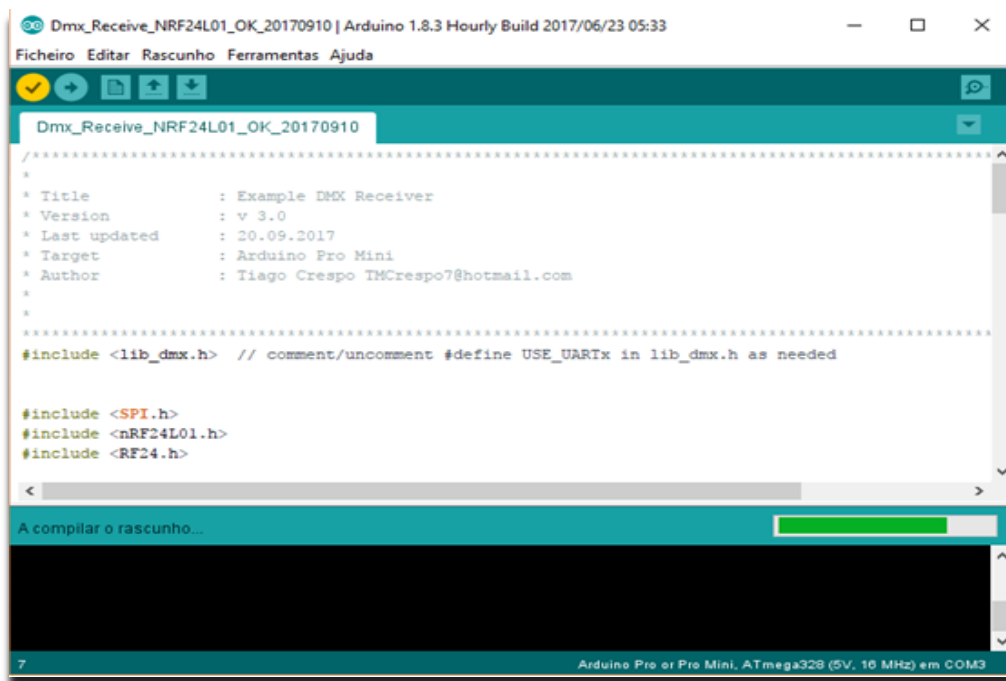


Figura 3.12 - IDE Arduino.

3.8 Programação do HMI

A programação HMI surge da necessidade de construção de uma interface “amigável” com o utilizador, sendo que é fundamental no desenvolvimento de um sistema e na sua utilização, uma vez que, estabelece um canal de comunicação entre o homem e a máquina ou o computador, no qual são feitas as interações visando atingir um objetivo comum. Por norma estas interfaces fazem parte de um sistema automático ou computacional e determinam como os utilizadores podem operar e controlar o sistema.

A importância dos *HMI*s é evidente, destacando-se alguns aspetos que reforçam ainda mais esta importância, tais como:

- Disseminação do uso em sistemas e equipamentos como *smartphones* e *tablets*;
- Aumento da complexidade dos sistemas;
- Preocupação com a qualidade do *software* (conforme Norma ISO/IEC 9126-1).

Permite ainda ao operador, em determinadas circunstâncias, ir além da operação da máquina e observar o estado do equipamento, bem como interferir no processo. Os principais objetivos dos dispositivos HMI são a otimização no controlo e supervisão de sistemas configuráveis e dependentes de ação/interação, seguros e funcionais, garantindo utilidade e efetividade. Para tal este dispositivo apresenta como apoio um *software* de suporte, programação e configuração.

Nextion *Editor* é um *software* de construção e desenvolvimento de ambiente visuais gráficos, aplicável a dispositivos do tipo GUI, (*Graphical User Interface*), como displays, TFT e *écrans* táteis. Este editor *WYSIWYG* permite, adicionar diretamente componentes das suas bibliotecas, tais como botões, caixas de texto, barras de progresso, *sliders*, painéis de instrumentos, entre outros, para enriquecer o seu *design* de interface. Com este *software* é possível desenvolver todo o ambiente gráfico em múltiplas páginas, inserir fotografias e vídeos, criar e editar gráficos, visualizar dados, sinais e inserir funções. Esta ferramenta conta ainda com uma função de *drag-and-drop*, que garante ao utilizador uma melhor e mais fácil interação e uma redução do tempo na sua programação [54]. Na Figura 3.13 é apresentado o ambiente de desenvolvimento do HMI.

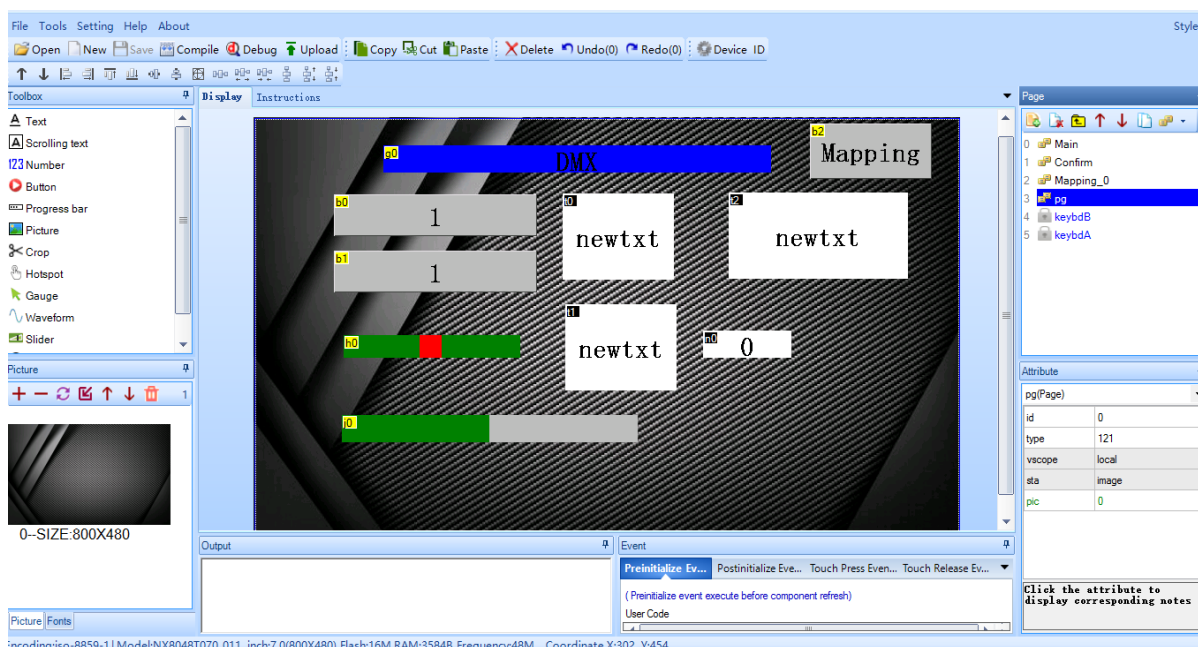


Figura 3.13 - Ambiente Nextion *Editor*.

3.9 Sumário

Neste capítulo foram apresentados os sistemas embebidos e os microcontroladores, tendo sido referidos os microcontroladores utilizados atendendo às suas especificações mais relevantes. Foram ainda expostas as principais ferramentas de desenvolvimento utilizadas, indispensáveis à criação destes sistemas de forma rápida e eficaz, ajudando a minimizar eventuais falhas, quer na fase de desenvolvimento quer no funcionamento a longo prazo. Também neste capítulo foi apresentado o HMI, as suas características e as vantagens na sua utilização.

4 DESENVOLVIMENTO

O presente capítulo descreve, etapa a etapa, todo o processo de desenvolvimento do presente projeto de investigação incluindo, não só o *hardware* necessário, mas também a programação dos sistemas desenvolvidos. Assim sendo, cada subcapítulo inicia-se com uma descrição geral das metodologias usadas, à qual se segue uma explicação detalhada em termos de *hardware* e de *software*.

4.1 Fundamento da seleção da tecnologia e metodologia adotadas

No estudo e desenvolvimento deste projeto, a correlação qualidade e o custo do seu desenvolvimento, sempre foram as principais linhas orientadoras, dado que, este foi desenvolvido na íntegra, de forma individual, por meios próprios.

No que concerne ao tipo de comunicação sem fios utilizada optou-se pela comunicação radiofrequência em detrimento de outras existentes. Esta escolha derivou do facto de se tratar de um tipo de comunicação com inúmeras vantagens: custos reduzidos, transmissão célere de dados e informações, elevada versatilidade, permitindo a comunicação a longas distâncias, conforme o *hardware* utilizado.

Uma vez definido o tipo de comunicação a implementar, foi efetuado o estudo da tecnologia a utilizar, dentro deste, foram abordadas tecnologias como o *Wi-fi*, o *Bluetooth*, o *ZigBee*, entre outras. De acordo com a aplicação pretendida, foi ainda efetuada uma análise detalhada dos módulos dedicados de radiofrequência, o nRF24L01 ou o módulo MRF24J40MA.

Face ao exposto, foi efetuada uma breve introdução às tecnologias e dispositivos considerados para o desenvolvimento deste projeto. Nesta irão ser apresentados os dispositivos conforme a tecnologia que possuem e as suas características.

Considera-se pertinente também apresentar os protocolos dos dispositivos, porque se muitos usam apenas uma comunicação série simples, outros comunicam via I2C, SPI, comandos AT ou outros. Desta forma pode-se verificar quais as comunicações mais indicadas face aos requisitos enumerados.

4.1.1 *Wi-Fi*

Nesta tecnologia, foi ponderada a utilização do módulo ESP8266 (Figura 4.1), um módulo *transceiver* baseado no *System On a Chip (SoC)*, ESP8266 produzido pela *Expressif Systems*.



Figura 4.1 - Módulo *Wi-Fi* ESP8266 [60].

Este *chip* implementa todas as funcionalidades do protocolo TCP / IP completo, através da utilização de microcontrolador interno dedicado. Esta é a característica chave deste módulo, que se reflete na sua versatilidade e aplicabilidade. Isto é, a utilização deste módulo permite estabelecer uma ligação direta ao *Wi-Fi* simples, anulando qualquer necessidade de processamento externo para a gestão da comunicação *Wi-Fi*, uma vez que esta lógica de controlo pode ser implementada diretamente na própria placa. As principais características deste são: o tamanho muito reduzido e o preço de comercialização, que ronda os dois euros, ou até mesmo um custo inferior, e a facilidade com que o mesmo pode ser integrado em diversas e variadas soluções, uma vez que depende apenas de uma comunicação série, *UART*. A seleção do módulo sucessor deste, o módulo *ESP32* que além do *Wi-Fi* integrado, possui um modo duplo de *Bluetooth* e emprega um microprocessador *Tensilica Xtensa LX6* em ambas as versões *dual-core* e *single-core*, também foi ponderada devida à ampla gama de recursos e potencialidades de criação de redes mistas.

Nesta tecnologia, o consumo de energia, a suscetibilidade a interferências e o alcance da rede são um problema. Foi verificado que o *ESP8266* apresenta um consumo de energia bastante elevado comparativamente com outros dispositivos, o que afeta uma futura possibilidade de implementação de baterias de alimentação no sistema desenvolvido. A banda de 2,4 GHz encontra-se amplamente ocupada e os três canais, não sobrepostos, disponíveis oferecem opções limitadas a este módulo que apresenta ainda uma cobertura restrita, dificuldades de funcionamento em ambientes com muitos obstáculos, e possibilidade de interferências com outros dispositivos IEEE 802.11 existentes, como microfones sem fios, redes de controlo de sistemas de som e imagem. A segurança, a viabilidade e a evolução despontam como preocupações imediatas em relação à adoção desta tecnologia. Em suma o módulo *ESP8266*, seria uma excelente escolha devido à elevada velocidade de transmissão (54Mbit/s +), caso não apresentasse as desvantagens acima descritas e caso esta aplicação não pretendesse afastar-se das bandas de frequência padrão excessivamente utilizadas pelo protocolo *Wi-Fi* [61] [62].

4.1.2 Bluetooth / BLE4.0

Relativamente à tecnologia *Bluetooth* esta apresenta limitações físicas e práticas, nomeadamente o curto alcance, o baixo débito, a estruturação da rede a que as comunicações podem ser efetuadas e não menos importante o custo de aquisição.

O *Bluetooth* é um dos protocolos sem fios mais utilizados nas especificações do *IoT*, especialmente com a recente introdução da extensão *Bluetooth Low Energy*, também conhecida pelos utilizadores da *Apple* como *iBeacon*.

Os benefícios do protocolo *BLE* foram notáveis aquando do lançamento do *Bluetooth 4.0*, proporcionando o desenvolvimento de placas com um baixo consumo de energia, o que auxilia na elaboração de placas alimentadas por bateria com um tempo de trabalho superior a 1-2 anos, em alguns casos.

Outra característica, já implementada em diversas aplicações (por exemplo, *smartphones*), é a possibilidade de incorporar o protocolo de gestão diretamente no nível do *kernel* sem solicitar qualquer intervenção pelos utilizadores. Isso facilita a configuração de uma rede de dispositivos *Bluetooth*, com menor latência e maior alcance em relação ao padrão *Bluetooth*.

Voltando à visão geral dos equipamentos ponderados, considerando que se optava pela tecnologia *bluetooth* um dos módulos sujeitos a possível seleção seria o módulo *bluetooth V4.0 nRF51822 BLE*. Este é representado na Figura 4.2.

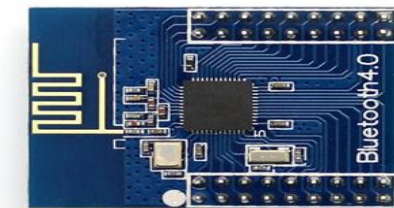


Figura 4.2 - Módulo *Bluetooth V4.0 nRF51822 BLE* [63].

Este é um módulo *SMD BLE* e é um *transceiver* multiprotocolo RF a 2.4GHz, construído envolto do processador ARM Cortex-M0 de 32bits, relativamente económico, pequeno e fácil de usar. Este apresenta as interfaces digitais de comunicação, SPI, I2C e UART [64].

Este módulo é uma alternativa vantajosa face aos módulos *Bluetooth* da família HC, amplamente conhecida e económica. Esta tecnologia apresenta uma alta velocidade de transmissão de dados e um baixo consumo médio de energia. A não utilização desta tecnologia, e consequentemente deste módulo, ou outros semelhantes, prende-se com o curto alcance e velocidade de transmissão e o facto de ser uma tecnologia consideravelmente dispendiosa.

4.1.3 Zigbee

ZigBee é uma especificação para um conjunto de protocolos de comunicação de alto nível usados para criar redes de área pessoal (*WPAN*) construídas a partir de rádios digitais pequenos e de baixa potência. O *ZigBee* é baseado no padrão IEEE 802.15.4. Embora o seu baixo consumo de energia limite as distâncias de transmissão para 10-100 metros, os dispositivos *ZigBee* podem transmitir dados em longas distâncias, transmitindo dados através de uma rede de dispositivos intermediários para alcançar distâncias superiores. O *ZigBee* é tipicamente usado em aplicativos de baixa taxa de dados que requerem uma longa vida útil da bateria e redes seguras (uma vez que é compatível com criptografia de 128 *bits*): automação residencial, cuidados de saúde, aplicativos de controlo industrial com curto alcance e baixa taxa de *bits*.

Na Figura 4.3 é possível observar dois modelos do módulo ZigBee.



Figura 4.3 - Módulos *ZigBee* [65].

Apesar desta tecnologia apresentar inúmeras vantagens no que respeita ao desenvolvimento deste projeto, entre as quais a facilidade de criação da rede e o desenvolvimento de programação.

O baixo débito de dados (250kbit/s), o custo elevado e o facto de possuir um alcance bastante inferior a outro tipo de tecnologias foram determinantes na exclusão da mesma para a elaboração do presente projeto.

4.1.4 Outros *Transceivers* Rádio Frequência

Entre as restantes tecnologias existentes, algumas não têm aplicabilidade na conceção do projeto a desenvolver, como por exemplo a tecnologia *GSM* ou mesmo *GPS*, no entanto, existe ainda vários *transceivers* rádio frequência prioritários, que permitem a comunicação de dados a distintas frequências, nomeadamente, alguns que operam na banda de 2,4 GHz, enquanto outros usam outras bandas.

Estes podem conectar-se via UART, SPI ou até I2C e possuem muitos recursos sofisticados, como modos de baixa potência (baixo consumo), múltiplos canais, interrupção de canal, ajuste dinâmico de frequência, CRC, retransmissão, entre outros. O alcance destes pode variar, aproximadamente, entre 10 e os 1000 metros.

Dentro desta tecnologia, e além do módulo nRF24L01, foi ainda ponderado o módulo MRF24J40MA.

Entre todos os *SoC* e soluções disponíveis no mercado, um dos mais completos e de melhor desempenho é o *Microchip MRF24J40MA*, um módulo *transceiver* de rádio baseado no padrão IEEE 802.15.4 que funciona a 2,4 GHz. O *MRF24J40MA* possui uma antena PCB integrada, circuitos de correspondência e também suporta os protocolos como: *ZigBee*[™], *MiWi*[™] e *MiWi P2P*[™]. O módulo *MRF24J40MA* conecta-se a centenas de microcontroladores PIC® através de uma interface SPI de 4 fios e é uma solução ideal para redes de sensores sem fios, domótica, automação de edifícios e aplicações de consumo [66]. Na Figura 4.4 encontra-se representado o módulo MRF24J40MA.



Figura 4.4 - Módulo *MRF24J40MA*. Adaptado de [66].

Este módulo possui inúmeras vantagens no que respeita ao desenvolvimento deste projeto, entre as quais a facilidade de criação da rede, no desenvolvimento do projeto e o modo *TurboBust* que permite velocidade de até 2 Mbps. No entanto, o custo elevado e o facto de possuir um alcance bastante inferior a outro tipo de tecnologias foram determinantes na exclusão da mesma para a elaboração do presente projeto.

4.1.5 Módulo selecionado para a elaboração do projeto: nRF24L01

Embora algumas das tecnologias sem fios e equipamentos atrás descritos apresentem características bastante favoráveis, é importante entender onde se encontram quando comparadas entre si.

Para tal foi construída a Tabela 4.1 que permite comparar as diferentes tecnologias conforme as suas características.

Tabela 4.1 - Tabela de comparação das diferentes tecnologias

Tecnologia	BLE	Wi-Fi	MRF24J40MA	nRF24L01
Módulo	nRF5182	ESP8266	MRF24J40MA	nRF24L01
Camada física	802.15.1	802.11	Suporta 802.15.4	-
Complexidade	Media	Alta	Baixa	Media
Canais	7	14	64000	126
Banda de Frequência	2.4 GHz	2.4/5 GHz	2.4 GHz	2.4 GHz
Modulação	<i>GFSK</i>	<i>OFDM/DSSS</i>	<i>DSSS</i>	<i>GFSK</i>
Alcance aproximado	<35m	<100m	<400m (PTP)	<520m
	<i>Mesh</i>	<i>Mesh</i>	<i>Mesh</i>	<i>Mesh</i>
Velocidade de	0,25/1/2 Mbps	11/54/150 Mbps	0,250 Mbps	0,25/1/2Mbps
Corrente de pico em	<13 mA (RX)	56 mA (RX)	19 mA (RX) 23	13.5 mA (RX)
Corrente de pico em	<10 μ A	<20 μ A	<2 μ A	<26 μ A
Corrente de	10 mA	80mA	23mA	15mA
		WEP/TKIP		-
Tamanho	20.8x17.0x0.8mm	24.8x14.3x0.8mm	29.9x17.8x0.8mm	38x16.46x0.8mm
Preço medio	5 Euros	5 Euros	8 Euros	3 Euros

De referir, que o módulo ZigBee S2 S2C, descrito no subcapítulo 4.1.3, não constou na presente tabela de comparação por apresentar um muito baixo débito para a aplicação pretendida, para além de apresentar um custo bastante elevado. Como pode ser consultado no Capítulo 3 o módulo nRF24L01 da *Nordic* foi a solução selecionada para o desenvolvimento das duas fases do presente projeto. A sua seleção derivou do facto de apresentar inúmeras vantagens face às alternativas disponíveis no mercado, destacando-se: fiabilidade na comunicação, menor probabilidade de ocorrência de interferências, custo reduzido, alcance favorável, baixo consumo energético, versatilidade na criação da rede de comunicação e possuir canais fora do padrão 802.01.

Tal como referido anteriormente, o módulo nRF24L01+ possui 7 canais de comunicação o que habilita a um maior controlo de transmissão e receção de dados e à possibilidade de distribuição de canais em modos de funcionamento de retransmissão. A parametrização cuidada deste módulo permite não só uma otimização no alcance, como também melhorias em termos de versatilidade. Com o intuito de no mínimo igualar a taxa de débito do DMX, tem de ser garantida uma velocidade de pelo menos 250kbps, se não forem transmitidos dados adicionais. Como para além da informação DMX têm de ser enviados dados de controlo e ser ainda garantida uma margem de retransmissões entre os diversos módulos, foi definido um limite mínimo de velocidade de pelo menos 1Mbit/s.

A seleção do método de comunicação, emissor, retransmissor e recetores individuais, adveio do facto de apresentar vantagens, uma vez que desta forma pode ser aumentada a distância entre o palco e a régie, através da colocação do retransmissor, e permitir ainda a aplicação de vários retransmissores, garantindo uma maior flexibilidade do sistema. Ou seja, possibilita que a ligação entre a régie e o palco possa ser efetuada por cabo ou por transmissão para o retransmissor, com recurso à parametrização da comunicação através do HMI.

Em relação à seleção do microcontrolador optou-se pelo Arduino, maioritariamente por ser um equipamento de custo reduzido e de fácil implementação, por já ser uma placa de desenvolvimento bastante completa. Apresenta ainda um IDE de fácil programação e desenvolvimento e uma respetiva ampla rede de partilha de conhecimento e bibliotecas de suporte. A placa de desenvolvimento Arduino Mini Pro, bastante compacta, completa e versátil, estabelece uma ótima relação qualidade preço, diminui gastos de desenvolvimento e apresenta uma grande robustez e fiabilidade. Esta placa de desenvolvimento preenche favoravelmente todas as requisições para a nossa aplicação.

Como visto, anteriormente, a opção pelo módulo *RS-485* centrou-se no seu custo e facilidade de utilização. Por último, a seleção da porta micro USB, centrou-se na sua versatilidade, fiabilidade e robustez. Uma vez que se trata de uma ficha de alimentação amplamente utilizada e a disponibilidade de transformadores, compactos e fiáveis, com recurso a esta ficha é constante.

4.2 Controlador DMX sem fios principal (Régie) – 1ª fase

O objetivo deste controlador é permitir a implementação de um sistema de controlo sem fios de comunicação bilateral, junto do elemento controlador dedicado DMX512. Este sistema é responsável pela receção do sinal DMX, proveniente do controlador de iluminação, pela organização dos dados recebidos e posterior envio, via rádio, para os restantes dispositivos da rede. Como referido, este controlador é responsável pelo envio de toda a comunicação DMX, da zona técnica para a zona cénica.

4.2.1 Diagrama geral de funcionamento

De acordo com a aplicação pretendida, foram analisadas as especificações necessárias para o desenvolvimento e implementação de um sistema de transmissão sem fios e controlo, de modo a garantir uma elevada fiabilidade na comunicação, resultando na seguinte listagem:

- Dispor de interface físico que permita e suporte a receção do sinal DMX512;
- Permitir o processamento e manipulação dos dados DMX recebidos, a fim de possibilitar a sua posterior transmissão para a rede sem fios;
- Incluir um *transceiver* rádio de modo a permitir uma comunicação bidirecional;
- Dispor de *interfaces* de comunicação de forma a estabelecer a sua interligação com elementos periféricos do sistema;
- Projeção do circuito de alimentação para permitir futura implementação de baterias;
- Contemplar a possibilidade de desenvolvimento de duas variantes deste módulo, uma sem interface com o utilizador e outra com interface gráfica (HMI), que permita ao utilizador a configuração da rede;

Na Figura 4.5 encontra-se o diagrama de *interfaces* que o controlador terá de possuir, de modo a cumprir os requisitos estabelecidos.

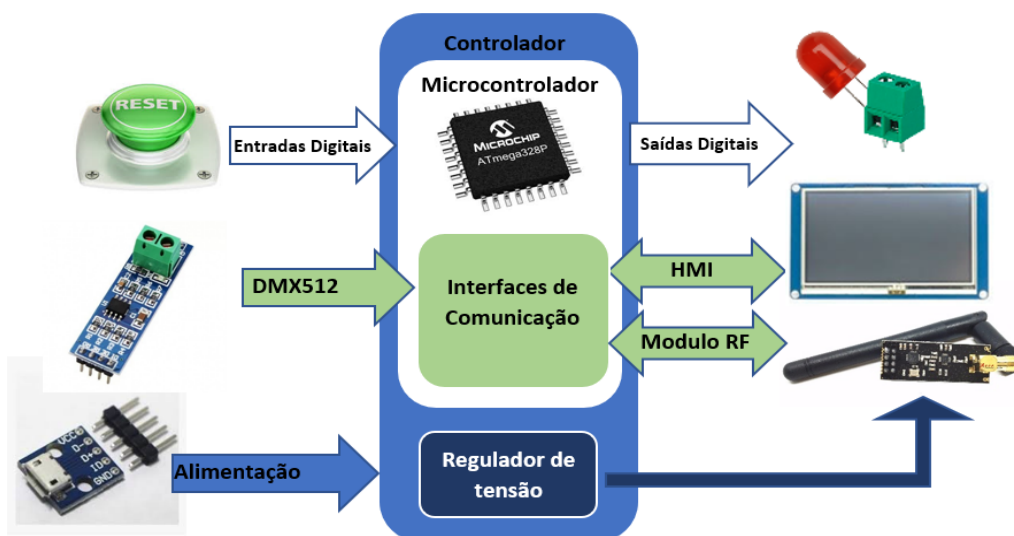


Figura 4.5 - Diagrama de *interfaces* do controlador sem fios DMX principal.

4.2.2 Hardware do controlador da transmissão DMX

Como supracitado, em termos de *hardware*, numa primeira fase, foi utilizada a placa de desenvolvimento *Arduíno Uno*, um módulo de comunicação RS-485, um módulo de comunicação RF, nRF24L01. Deste modo, foi iniciado todo o desenvolvimento e interligação de todos os componentes periféricos necessários ao desenvolvimento da *interface*, através da utilização de uma *bradboard*. Na Figura 4.6 é possível observar o *layout* de desenvolvimento inicial, encontrando-se visíveis os principais componentes físicos utilizados, descritos no capítulo anterior.

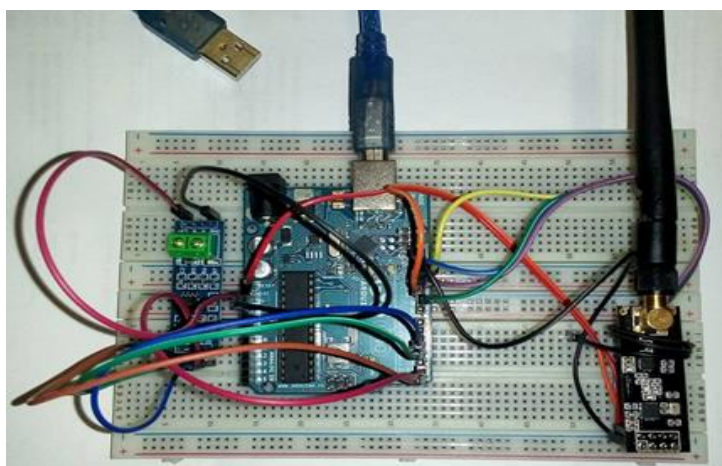


Figura 4.6 - Imagem real do primeiro *layout* desenvolvido.

Após a realização dos primeiros testes de funcionamento, foi efetuada uma reestruturação ao nível do *hardware*, de modo a otimizar o sistema a desenvolver. Como tal, esta reestruturação culminou no desenvolvimento de um módulo mais compacto e prático para a realização de testes e o estudo de otimizações futuras. A construção deste protótipo teve início após a verificação de funcionamento da integração de todos os elementos constituintes do módulo, enquanto sistema.

4.2.2.2 Módulo nRF24L01

Atendendo ao facto da fonte de alimentação ser de 5 V e deste dispositivo ser alimentado a 3,3V foi necessária a aplicação de um regulador de tensão. Para tal, foi considerada a opção de implementação de um regulador de tensão linear ou comutado. Devido ao baixo consumo, à reduzida diferença de potencial e às dimensões optou-se pelo regulador linear 78M33C da Texas Instruments, que permite uma corrente à saída de até 500 mA. Esta solução permite a simplificação do desenvolvimento da placa de circuito impresso, uma vez que este pode ser colocado, em altura, sobre a placa. Relativamente à interface com o microcontrolador, não foi necessária a colocação de divisores de tensão, uma vez que estes são tolerantes a 5V. Para a interligação do módulo RF com o microcontrolador foi efetuada a ligação dos pinos de alimentação, *Vcc* e *GND*, respetivamente à saída de 3,3 V do regulador de tensão e *GND* do microcontrolador. Para estabelecer a interface SPI, a ligação dos pinos de *CE*, *CS*, *SCK* e o sinal *MOSI* e *MISO*, referente à comunicação do mestre para o escravo, é efetuada recorrendo às saídas digitais 9, 10, 13, 11 e 12, respetivamente.

4.2.2.3 Módulo RS-485

Este módulo desempenha a função de conversão de um sinal série com referencial à massa para um sinal série diferencial. Assim, não existe qualquer alteração entre a trama de dados recebidos e enviados. Desta forma o sinal do par diferencial do protocolo DMX512, ligado de acordo com o sinal elétrico aos pinos A e B, e os pinos ‘RO’ e ‘IO’ ligados ao porto série nativo do controlador, nomeadamente, na saída digital ‘0’ (RX) e na saída digital ‘1’ (TX).

4.2.2.4 Placa desenvolvida

Este controlador foi desenvolvido com recurso a uma *protoboard*. No desenvolvimento da placa foram seguidos os seguintes passos:

- ✓ Inclusão dos *headers* para a fixação do microcontrolador e do módulo NRF24L01+;
- ✓ Implementação direta do módulo RS-485;
- ✓ Aplicação direta de uma ficha de alimentação micro *USB*;
- ✓ Aplicação de filtros de alimentação e regulador de tensão dedicado ao módulo sem fios.

Na Figura 4.9, é possível visualizar o esquema de ligações efetuado.

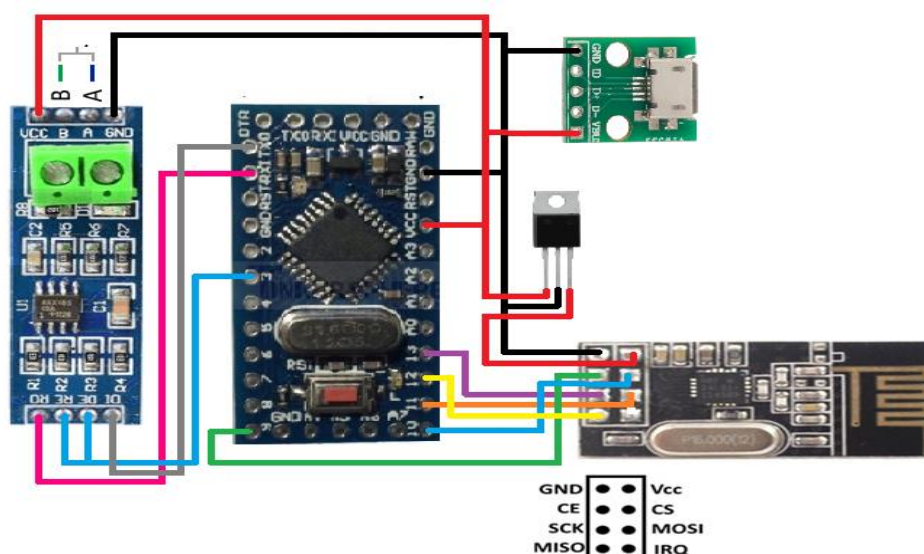


Figura 4.9 - Esquema de ligações do controlador principal.

Apesar do protótipo desenvolvido ter sido criado sob uma *protoboard*, foi desenvolvido de igual modo, um projeto *Eagle* com o intuito de a qualquer altura este poder ser impresso e adquirido.

Na Figura 4.10 é possível verificar o esquema de ligações desenvolvido para a placa de circuito impresso.

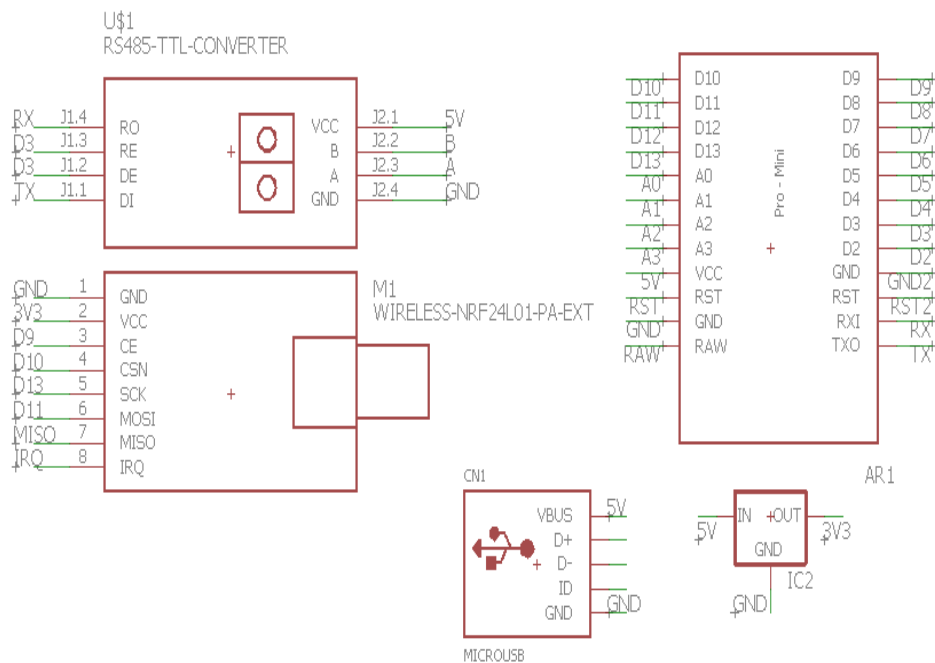


Figura 4.10 - Esquema de ligações *Eagle* do controlador principal.

O respetivo desenho da placa desenvolvida pode ser observado no Anexo II, pág. 95.

4.2.2.5 HMI

Como visto anteriormente, o *HMI* selecionado foi o Nextion NX8048T070, um *HMI* genérico de sete polegadas, com propriedades multicolor e *multitouch*. Este equipamento permite a sua interligação série através de dois pinos de comunicação série e dois para efetuar a sua alimentação.

A utilização do *HMI* prende-se com a necessidade de criação de uma interface com o utilizador do sistema de forma a permitir que este realize determinadas funções. Entre as quais é objetivada uma função de mapeamento remoto dos dispositivos a controlar.

Este dispositivo permite o desenvolvimento de um amplo leque de funções, se culminado com a programação do microcontrolador.

4.2.3 Programação desenvolvida

Atendendo às especificações pretendidas, foi efetuada a programação do microcontrolador de modo a providenciar o englobamento de todos os elementos periféricos e efetuar, em função dos mesmos, o controlo das comunicações do sistema.

A estrutura lógica resultante encontra-se no diagrama presente na Figura 4.11.



Figura 4.11 - Diagrama de funcionamento do controlador sem fios DMX512.

No que diz respeito à programação deste controlador, encontra-se implementada a atualização constante da matriz de dados DMX em memória. Esta atualização é efetuada aquando da receção de dados via série. De modo a minimizar a ocorrência de falhas de comunicação, o envio destes dados via rádio realiza-se também em permanência, ou seja, sempre que o microcontrolador executa um ciclo, efetua uma transmissão.

Para garantir a receção correta de todos os dados da porta série independentemente da tarefa que o microcontrolador esteja a efetuar, encontra-se implementada a receção destes dados por interrupção. Isto é, a utilização da única porta UART nativa do microcontrolador foi, inequivocamente, aplicada ao controlo do protocolo DMX512, uma vez que esta comunicação deve ser prioritária em relação a outras. Isto porque, representa um sistema crítico no controlo de iluminação profissional e exige um alto débito para o bom funcionamento do módulo conversor RS-485. O canal selecionado determina a frequência central de funcionamento do nRF24L01. Cada canal ocupa uma largura de banda inferior a 1MHz à velocidade de 250kbps e de 1Mbps e inferior 2MHz a 2Mbps. A fim de evitar a sobreposição de canais a 2Mbps, deve ser garantida uma largura de banda livre de pelo menos 2MHz.

Como visto anteriormente (Ver subcapítulo 3.3, pág.37), a largura de banda destes módulos, varia entre 2.400 e 2.525 GHz e atendendo a que a resolução da seleção de canais é de 1MHz, é possível comprovar a existência dos 126 canais. Deste modo, o canal “0” possui uma frequência central de 2.400 GHz e o canal 126 de 2.525 GHz.

No que diz respeito à seleção do canal de funcionamento do nRF24L01, foi efetuada a análise da gama de frequências menos propícias a interferências. Tendo em consideração a vasta gama de equipamentos que operam via *Wi-fi* (802.11), foi considerada a ocupação das frequências correspondentes aos 14 canais deste protocolo que se encontram padronizados de acordo com a Tabela 4.2.

Tabela 4.2 - Lista de canais do padrão 802.11

Nº do canal	Baixa frequência	Alta Frequência
1	2401	2423
2	2406	2428
3	2411	2433
4	2416	2438
5	2421	2443
6	2426	2448
7	2431	2453
8	2436	2458
9	2441	2463
10	2446	2468
11	2451	2473
12	2456	2478
13	2461	2483
14	2473	2495

Após a análise da Tabela 4.2 e, atendendo à necessidade no afastamento destas gamas de frequências a fim de evitar eventuais interferências causadas por esta tecnologia, optou-se pela seleção do canal 110, cuja frequência de 2.510GHz, garante este distanciamento.

A frequência do canal RF é definida pelo registo RF_CH do nRF24L01 e calculada através da Equação 4.1.

Equação 4.1-Equação de cálculo da frequência do canal RF

$$F_0 = 2400 + RF_CH \text{ [MHz]}$$

Para além do exposto foi ponderada a utilização do sistema de deteção de erros. O módulo nRF24L01 possui um mecanismo de CRC que permite ser configurado a um ou dois Bytes.

Este sistema foi implementado, no desenvolvimento do projeto uma vez que este, possibilita ao recetor a validação da receção dos diversos pacotes de dados.

Este sistema de controlo minimiza ainda a propagação de eventuais erros de pacotes de dados para a rede DMX, ou seja, se o teste de CRC falhar o pacote de dados será descartado.

O pacote de dados também é descartado, quando a ferramenta *Enhanced Shockburst* identifica uma incoerência entre o número de identificação do *pipe* de comunicação do pacote de dados recebido e o número de identificação do *pipe* de receção.

No entanto, a função *Acknowledgment* não foi utilizada, mesmo tendo sido ponderada entre os dois módulos principais, pelo facto de se possuir uma comunicação de alto débito e esta requerer atualização de pacotes de forma consecutiva. Como visto anteriormente, o protocolo DMX, também, não utiliza esta função, uma vez que não funciona por se tratar de uma comunicação multiponto e poder contribuir para atrasos e falhas de comunicação. Para a gestão dos módulos nRF24L01 foram utilizadas três bibliotecas, nomeadamente as bibliotecas “SPI.h”, “nRF24L01.h” e “RF24.h”. A primeira, permite a comunicação do Arduino, configurado como mestre da rede, com dispositivos escravos, via *SPI*.

A segunda livreria, específica o *driver* do respetivo dispositivo escravo, enquanto a biblioteca “RF24.h oferece maior facilidade na gestão da comunicação rádio e maior controlo na gestão de pacotes de dados.

No entanto, cada trama de dados do módulo nRF24L01, possui um tamanho máximo de 32 *bytes*, o que resulta insuficientemente no envio dos 512 canais (*bytes*) da trama DMX. Para solucionar esta limitação, foi necessário implementar uma fragmentação do *array* de dados DMX, de forma a obter 32 pacotes de 16 *bytes*, cada. Além desta desfragmentação foi necessário adicionar no início de cada pacote a transmitir, um *byte* para o número identificador de mensagem que identifica o número do pacote DMX a que se refere este subpacote (*ID_Message*), e outro *byte*, sucessivo a este, para o número de identificação do pacote que se refere ao número do pacote da subdivisão (*ID_Package*).

No controlo de dados DMX, foi utilizada uma biblioteca da plataforma Arduino. Esta biblioteca exigiu distintas alterações na sua fonte, uma vez que se encontra construída para o controlo de quatro universos DMX com recurso à placa de desenvolvimento Arduino Mega, que possui quatro portos UART nativos. Esta é designada “lib_dmx.h” e permite maior facilidade no desenvolvimento e gestão dos dados DMX, no entanto, a limitação de um porto UART nativo do Arduino Mini Pro e a impossibilidade de utilização de portos série com recurso a bibliotecas de criação de portos *Software Serial*, obrigaram à realização de reconfigurações da biblioteca.

Como visto anteriormente, o *HMI* Nextion é uma ótima escolha para criação de interfaces com os dispositivos Arduino, uma vez que a maior parte das suas funcionalidades e processos são executados de forma autónoma no HMI. Ou seja, o microcontrolador apenas é responsável pelo envio de comandos para o HMI ou pela receção de notificações de eventos, como por exemplo a mudança de estado de um botão. Desta forma, todo o desenvolvimento e *design* do *layout* da GUI é feito num ambiente de desenvolvimento Nextion *Editor*.

A título de exemplo, o microcontrolador pode alterar o estado de uma variável presente no *Display*, enviando, via série, o valor da variável que pretende alterar, o ID da página em que se encontra e o ID do objeto a alterar. Desta forma, a biblioteca disponibilizada para Arduino (*Nextion.h*) permite a criação de objetos associados a cada componente presente no HMI, disponibilizando funções de leitura e escrita, dependendo do tipo de componente.

No entanto, como visto anteriormente, o HMI selecionado para a criação da interface com o utilizador, apenas pode estabelecer comunicação série. Desta forma, é requerido outro porto de comunicação UART, para estabelecer comunicação com este. Como o microcontrolador selecionado apenas disponibiliza um porto série nativo por *hardware*, é exigida a utilização de uma biblioteca de criação deste tipo de porta de comunicação, ao nível de *software*.

A biblioteca *SoftwareSerial* foi desenvolvida para a plataforma Arduino, de forma a permitir a comunicação série, sem capacidade de interrupção, através da utilização de qualquer porta digital do microcontrolador, com o intuito de replicar a funcionalidade da porta física com recurso a *hardware*. Esta biblioteca permite criar e possuir em funcionamento simultâneo, diversas portas série por *software*, com velocidades de até 115.200 bps. No entanto, o controlador apenas pode receber dados de um porto de cada vez, devido ao facto da leitura deste ser efetuada por *pulling*. Portanto para a intercomunicação com o *HMI* foi utilizada esta biblioteca, com recurso aos portos digitais 4 e 5.

Para carregar o projeto no HMI, é necessário desenvolver a programação da aplicação na ferramenta de desenvolvimento, compila-la e garantir a inexistência de erros e enviar o ficheiro da compilação para o cartão SD. Depois é apenas necessário colocar o cartão no HMI e iniciá-lo, para que este proceda a atualização do projeto desenvolvido.

Neste caso, a aplicação desenvolvida, permite executar o mapeamento dos terminais sem fios, de forma a possibilitar o “endereçoamento virtual” remoto dos recetores DMX. Para tal, os microcontroladores são programados de forma a poder receber uma função específica, aquando do mapeamento, que permite ao utilizador mapeá-los para que o microcontrolador principal efetue o desfasamento da trama e proceda à entrega, de forma correta, dos pacotes de dados DMX.

Ou seja, desde que haja conhecimento do respetivo endereço de cada máquina, podendo este, desta forma, ser igual em todas elas, é possível efetuar o desfasamento da trama conforme mapeamento efetuado. Este processo é muito semelhante à função de *discovery* do protocolo RDM, aplicado nos microcontroladores utilizados. A interface HMI, vai permitir ao utilizador comutar o parâmetro de retransmissão, de forma a poder funcionar sem o módulo retransmissor caso este, seja dispensável, bem como programar o número de retransmissores a utilizar, entre outros. Na Figura 4.12 é possível visualizar o ambiente gráfico da página de inicialização do HMI.



Figura 4.12 - Ambiente gráfico HMI.

Nesta página é possível iniciar o mapeamento das máquinas através do botão de “Start Mapping”. Este habilita a mudança de página da aplicação de forma a poder ser iniciada a função de mapeamento. Esta mudança de página requer de confirmação, com recurso a uma janela de *popup*, como pode ser visível na Figura 4.13

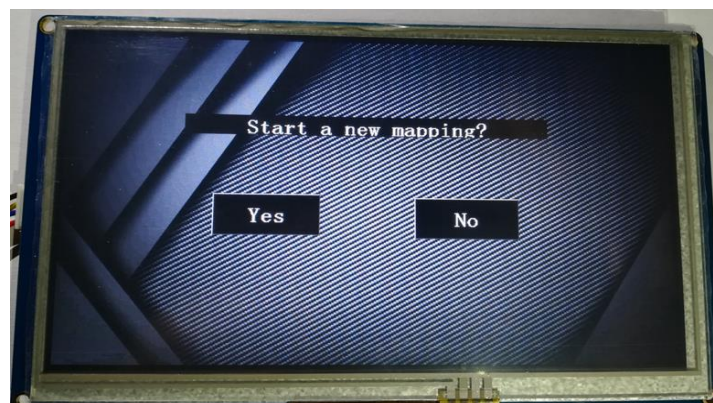


Figura 4.13 – Janela de confirmação da realização de mapeamento

Caso a confirmação não seja executada favoravelmente, é apresentada novamente a janela inicial. Efetuada a seleção positiva da confirmação, uma nova janela é apresentada, a qual permite de maneira simples a realização do mapeamento das máquinas a controlar.

Na página de realização do mapeamento é possível desabilitar de forma integral a função de mapeamento, e desta forma, realizar funções com os módulos sem fios de forma tradicional, ou iniciar o processo individual de mapeamento. O ambiente desenvolvido para esta página pode ser observado na Figura 4.14.



Figura 4.14 - Janela de realização de mapeamento das máquinas a controlar.

Para iniciar o mapeamento é necessário inserir o endereço conhecido da máquina em questão, que, como foi visto, pode ser o mesmo em todas as máquinas, desde que seja efetivamente o parametrizado, previamente, no dispositivo. E inserir o endereço do canal DMX definido aquando a realização de *Patch* no controlador dedicado DMX.

A identificação da máquina é realizada ainda de uma forma bastante primária, esta pode ser conhecida através da distribuição numerada dos dispositivos de controlo, ou efetuada com recurso a uma função de pedido de oscilação aleatório dos seus parâmetros de controlo.

Nesta fase inicial, foi a maneira mais fácil encontrada para realizar este objetivo. Para habilitar esta função aleatória numa determinada máquina, é apenas necessário selecionar a máquina que se pretende.

Para tal, foram adicionados dois botões, um de decretação e outro de incrementação que possibilitam ao utilizador deslocar-se entre máquinas, de forma a visualizar qual a que se encontra em tempo real a executar esta função.

Este processo é possível, porque o controlador principal, previamente emitiu um pedido de identificação a todos os controladores remotos funcionais, ao qual de forma individual e crescente vão ser identificados e silenciados.

Uma vez que, o controlador possui a lista de todos os dispositivos encontrados, pode ser iniciado o seu mapeamento.

Depois de realizada a identificação de todos os dispositivos, a ordem de tarefas é indiferente, uma vez que o mapeamento só é efetuado quando os dados presentes no ecrã são enviados para a máquina que se encontra em movimento aleatório.

O valor dos parâmetros de endereços pode, para já variar entre 1 e 512, ou seja, o limite do controlador para tal foi habilitada a escrita nestes campos como pode ser visto na Figura 4.15.



Figura 4.15 – Inserção do endereço DMX.

No desenvolvimento da programação deste controlador, as maiores dificuldades verificaram-se na compatibilidade entre dispositivos e respetivas bibliotecas utilizadas, no processamento e sincronização da trama DMX, no algoritmo de desfragmentação da trama DMX recebida e sua organização para envio via rádio.

Além destes, um problema de relevância significativa detetado, foi a alimentação dos módulos, a qual foi prevista ser efetuada por transformadores, que não possuem uma filtragem adequada ao nível da frequência de funcionamento, despertam erros na transmissão de pacotes e até falhas na comunicação.

4.2.3.1 Interface série com o computador

Para fins de desenvolvimento foi ponderado o acesso aos pinos de programação do microcontrolador, da ficha micro *USB*, pelo facto de, por um lado, o porto de programação ser o mesmo do porto de comunicação com o RS-485 e, por outro lado, a maioria dos transformadores possuem estes pinos interligados pelo risco de interferências.

No desenvolvimento deste controlador foi efetuada uma tentativa de disponibilização direta do acesso à porta *UART* nativa do microcontrolador. Foi comprovada a impossibilidade de utilização dos pinos de comunicação da ficha micro *USB*, uma vez que aquando da alimentação os pinos D+ e D- encontram-se interligados entre si, o que inviabiliza a utilização destes no estabelecimento de comunicação para a programação do microcontrolador.

Tal facto impede de forma integral o funcionamento correto do dispositivo em modo normal. Deste modo, foi anulada a hipótese de programar o microcontrolador por este meio. Foi ainda estudada outra alternativa, também ela desconsiderada, que visava a comunicação através da utilização dos pinos livres (não ligados) da ficha de comunicação DMX. A não opção por esta segunda alternativa derivou do facto de não possuir três pinos livres nesta ficha. Recorde-se apenas que para efetuar a programação remota do microcontrolador é necessária a disponibilização de 3 pinos de controlo: o RX, o TX, bem como a realocação do botão de *reset* do microcontrolador. Desta forma, foram excluídas todas as alternativas de programação remota do microcontrolador, uma vez que estas não colmatavam a necessidade de desmontagem do invólucro do dispositivo final.

4.3 Retransmissor DMX sem fios principal (Stage) – 1ª fase

Na segunda fase do projeto este controlador sofreu um *upgrade*, ao nível de programação, de forma a reencaminhar toda a comunicação DMX para todos os recetores sem fios terminais alocados individualmente em cada máquina ou criando grupos de máquinas.

O presente controlador atua como retransmissor, não apresentando qualquer alteração ao nível de dados, funcionando, basicamente, como *gateway* da rede, ou seja, deve efetuar a receção dos dados emitidos, via rádio pelo transmissor, e proceder ao seu reencaminhamento para os recetores remotos alocados a cada recetor DMX.

Este controlador foi concebido de forma a ser versátil, ou seja, pode receber os dados via rádio e proceder ao reencaminhamento destes, usando o mesmo meio ou por cabo.

O desenvolvimento deste controlador também foi faseado, à semelhança do anteriormente descrito. Mais uma vez, este controlador, integrante da zona cénica, é responsável pela receção de toda a comunicação DMX, emitida pelo controlador emissor da zona técnica.

4.3.1 Diagrama geral de funcionamento

De igual forma, foram analisadas as especificações necessárias para que o controlador em questão permita a retransmissão do protocolo DMX, sem fios, resultando nos seguintes itens:

- Dispor de capacidade de receção sem fios de comunicação DMX512;
- Dispor de interface físico que permita e suporte a transmissão do sinal DMX512;
- Permitir o processamento e manipulação de dados DMX recebidos caso seja pertinente;
- Incluir um *transceiver* rádio de modo a permitir uma comunicação bidirecional;
- Dispor de *interfaces* de comunicação de forma a estabelecer a sua interligação com elementos periféricos do sistema;
- Projeção do circuito de alimentação para permitir futura implementação de baterias;
- Permitir a versatilidade na comunicação e a sua configuração por parte do utilizador.

Na Figura 4.16 encontra-se o diagrama de *interfaces* que o controlador a deve possuir:

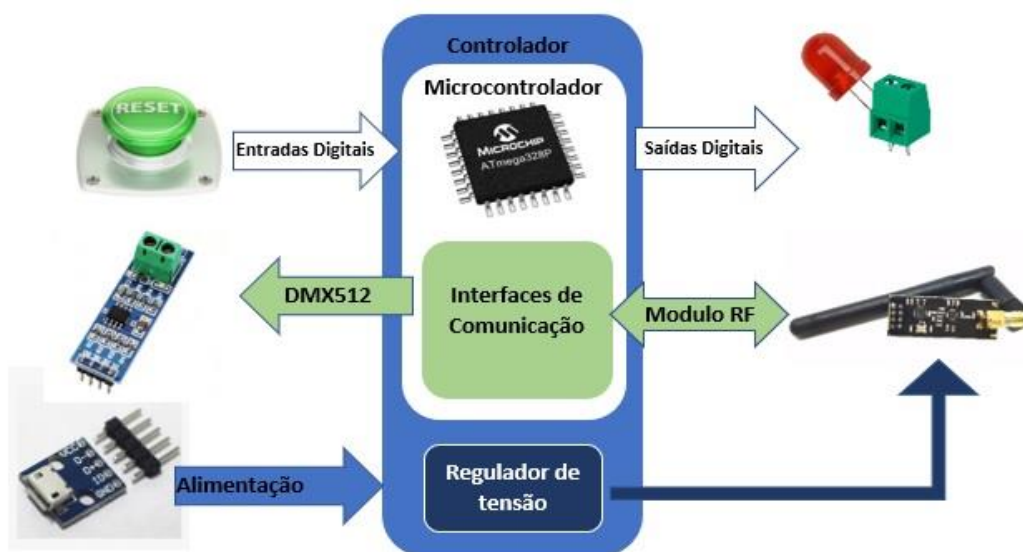


Figura 4.16 - Diagrama do recetor sem fios DMX512.

4.3.2 Alterações ou hardware do controlador de retransmissão DMX (Stage)

O controlador desenvolvido para a receção principal e retransmissão, apresenta grandes semelhanças, ao nível de *hardware*, com o controlador descrito anteriormente, exceto a substituição da tomada XLR5 fêmea por uma tomada macho e o facto de não permitir a implementação do suporte *HMI*.

Esta alteração, prende-se com disponibilização de maior versatilidade ao sistema construído, uma vez que que possibilita o reencaminhamento por cabo dos dados recebidos sem fios, desta forma podemos desprezar, caso exista necessidade o reenvio de dados via rádio e a utilização dos módulos terminais.

Esta versatilidade permite tornar o sistema redundante a falhas de comunicação ou danos e limitações ao nível de *hardware*. Contudo a este nível, este módulo segue as mesmas linhas de desenvolvimento, descritas no subcapítulo 4.2.2 (pág.57).

Na fase de testes, o protótipo desenvolvido apresentava a configuração ilustrada na Figura 4.17.

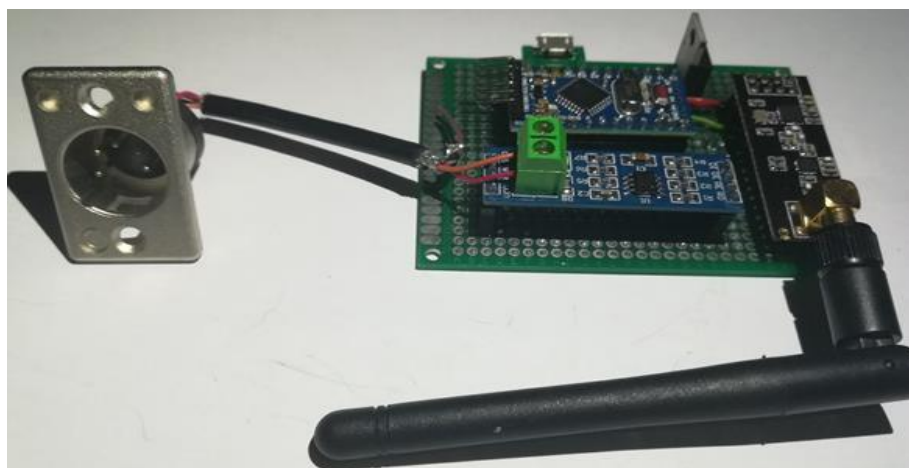


Figura 4.17 - Controlador recetor sem fios DMX512.

4.3.2.1 Placa desenvolvida

O desenvolvimento da placa deste controlador, não apresenta quaisquer diferenças relativamente à placa desenvolvida para o controlador do anterior, uma vez que não possui qualquer alteração ao nível de *hardware* fixo, nem alteração das interligações entre os diversos componentes, constituintes do sistema. Desta forma, a placa desenhada e desenvolvida para o primeiro controlador é válida também para este.

4.3.3 Programação desenvolvida

A programação do microcontrolador foi efetuada de modo a permitir a execução de diversas tarefas com alguma “simultaneidade”. Nesta aplicação é necessário garantir uma aquisição constante de dados provenientes do controlador transmissor principal e o seu envio, recorrendo à retransmissão via rádio ou à tomada de saída DMX disponibilizada para efetuar a ligação por cabos, caso sejam recorrentes alguns danos ou conflitos, entre módulos, em palco.

Portanto, as diferenças ao nível de programação centram-se na inversão da função de leitura por escrita, no módulo RS-485, esta função é gerida por *software*, pela biblioteca DMX, e a

permuta entre transmissão e receção é efectuada com recurso à alteração de estado do porto digital 3 do microcontrolador que estabelece comunicação com o pino do módulo RS-485.

Outra alteração significativa, centra-se na criação de outro *pipe* de comunicação para o reencaminhamento dos pacotes de dados recebidos, de modo a isolar as distintas comunicações. A criação deste é lógica e enquadra-se com arquitetura de construção do módulo nRF24L01, uma vez que a utilização do mesmo *pipe* de comunicação, gera conflitos de pacotes. Tal acontece, porque não existe distinção entre os pacotes recebidos pelo retransmissor ou pelo emissor.

Recorrendo à alteração da trama afim de incluir um ID do equipamento de origem da mensagem seria possível implementar esta estrutura de rede diferenciando a origem da mensagem. Na Figura 4.18 é possível observar o diagrama lógico do sistema.

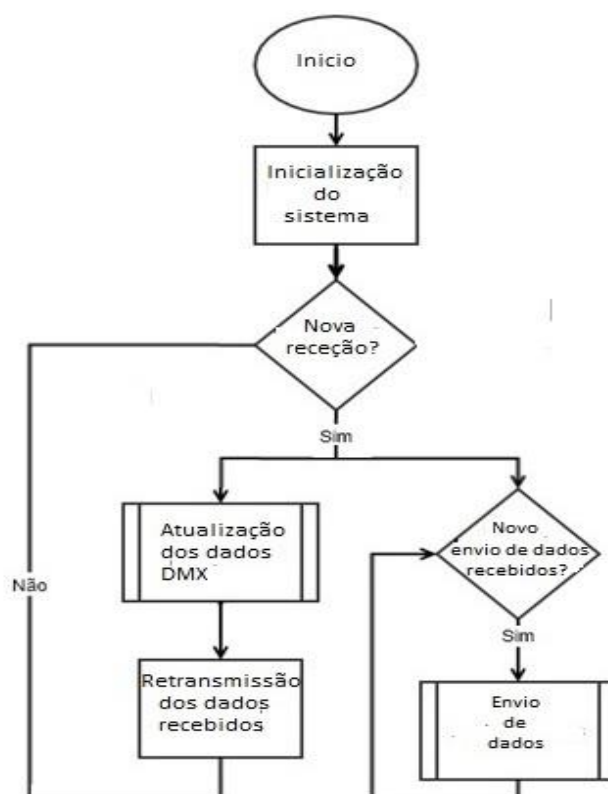


Figura 4.18 - Diagrama de funcionamento do recetor sem fios DMX512 principal.

Contudo optou-se pela implementação de *pipes* distintos, visto serem implementados ao nível de *hardware*, obtendo um melhor desempenho da rede e consequentemente do sistema. Tal acontece porque o endereço do *pipe* de emissão tem de ser validado pelo módulo rádio recetor.

Foi ainda ponderado a ramificação da retransmissão para varios *pipes*, de modo a criar subgrupos de máquinas, no entanto esta solução apenas é vantajosa quando o sistema apresenta uma grande quantidade de máquinas ou quando possuímos universos paralelos. Em contrapartida, esta ramificação aumenta drasticamente o fluxo de dados na rede, e consequentes efeitos de congestionamento da rede. Então a cada receção do fragmento da trama DMX, é actualizado o *array* do *buffer* de retransmissão do módulo rádio, sendo este fragmento alocado na respectiva posição, e efectuada a retransmissão destes fragmentos aos módulos recetores terminais.

4.4 Controlador recetores sem fios individuais DMX512 – 2ª fase

Este controlador é o elemento terminal da rede de comunicação DMX sem fios desenvolvida e tem como objetivo a receção dos pacotes de dados DMX referentes à máquina à qual está alocado, ou ao conjunto de máquinas que se sucedem a esta.

Importa ainda referir que este controlador foi sujeito a uma miniaturização aquando o desenvolvimento da placa de circuito impresso, no intuito de ser colocado no interior de uma estrutura metálica cilíndrica ou uma caixa com o objetivo de aumentar a resistência do equipamento face ao elevado manuseamento a que este é sujeito.

4.4.1 Diagrama geral de funcionamento

Para proceder ao desenvolvimento do sistema, foi efetuada a listagem das especificações pretendidas, resultando nos itens seguintes:

- Dispor de interface físico, conector XLR5, que permita a receção do sinal DMX512;
- Dispor de capacidade de receção sem fios de comunicação DMX512;
- Suportar a implementação de um módulo rádio, e dotar este de capacidade de receção;
- Permitir o processamento e manipulação dos dados DMX recebidos, via rádio;
- Incluir um *transceiver* rádio de modo a permitir uma comunicação bidirecional;
- Dispor de *interfaces* de comunicação de forma a estabelecer a sua interligação com elementos periféricos do sistema;
- Projeção do circuito de alimentação para permitir a futura implementação de baterias.

Na Figura 4.19, encontra-se o diagrama de *interfaces* que o controlador terá de possuir, de modo a cumprir os requisitos estabelecidos.

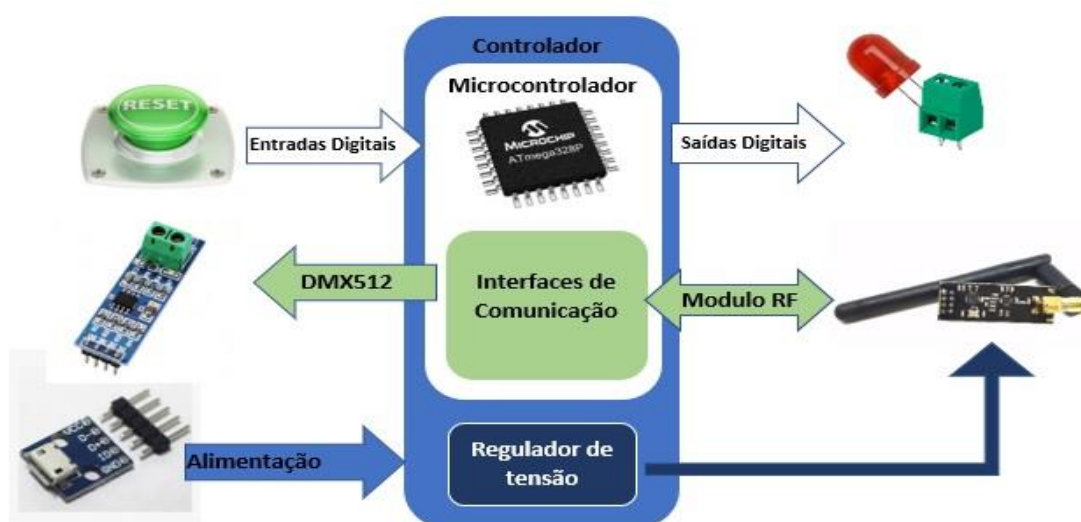


Figura 4.19 - Diagrama do controlador recetor terminal sem fios DMX.

4.4.2 Hardware do controlador de receção DMX Terminal

Como recorrente, na fase de desenvolvimento e teste foi criado um protótipo de reduzidas dimensões do módulo, com recurso a uma *proto-board*, de forma a experienciar o *layout* desenvolvido e a deteção de possíveis incoerências não previstas (ver Figura 4.20).

4.4.3 MAX 485

De referir ainda, que o módulo RS-485 foi descartado no desenvolvimento deste controlador. Foi implementado em sua substituição, um circuito equivalente deste, com recurso ao circuito integrado MAX485.

O circuito equivalente deste componente é apresentado na Figura 4.22.

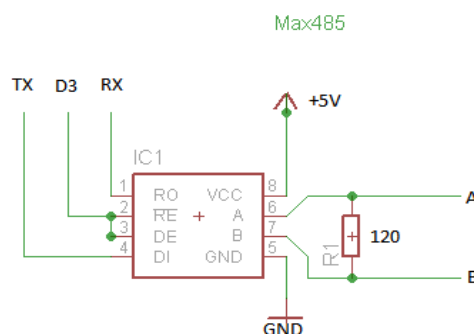


Figura 4.22 – Esquema equivalente de ligação do MAX485 [69].

4.4.4 Placa desenvolvida

De forma a permitir o aumento da resistência do controlador concebido, e seguindo a linha de desenvolvimento de alguns fabricantes, é proposto o desenvolvimento minimalista deste controlador, atendendo aos recursos e condições existentes. Para tal, foi desenvolvida uma placa PCB de dimensões muito reduzidas, cerca de 570x220mm. De modo a permitir a inserção do microcontrolador na placa, foram incluídos os terminais de ligação com os pinos necessários incluindo os terminais de alimentação possibilitando a alimentação externa da parte de controlo.

Foram ainda colocados na placa as zonas de fixação para o regulador de tensão e para o módulo nRF24L01+, ficando este ao lado do microcontrolador e ainda terminais de ligação para duas resistências, dois condensadores e o MAX485 ficando estes debaixo do microcontrolador. Na Figura 4.23 ilustra-se o desenho da placa desenvolvida.

No desenvolvimento da placa foram seguidos os seguintes passos:

- ✓ Inclusão de furação para a fixação direta do microcontrolador;
- ✓ Inclusão de furação para a fixação direta ou *header* para a o módulo NRF24L01+;
- ✓ Implementação direta do circuito integrado MAX 485 e periféricos;
- ✓ Aplicação direta de uma ficha de alimentação micro USB;
- ✓ Aplicação de filtros de alimentação geral de entrada e regulador de tensão dedicado.

Na Figura 4.23, é possível observar, à esquerda, o resultado do desenvolvimento da PCB desenvolvida e à direita o resultado final da mesma.

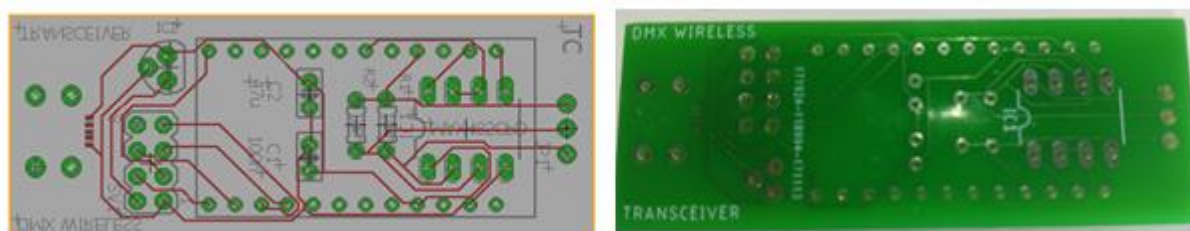


Figura 4.23 - Desenho da PCI do controlador DMX sem fios terminal e respetivo resultado final.

Por fim, foi efetuada a montagem dos componentes periféricos e o MAX485, dos *headers* de fixação do microcontrolador, da ficha de alimentação e do regulador de tensão.

Na Figura 4.24 é possível observar o desenvolvimento faseado deste controlador.



Figura 4.24 – Etapas de desenvolvimento do controlador desenvolvido

Na Figura 4.25 é possível observar o resultado do desenvolvimento.

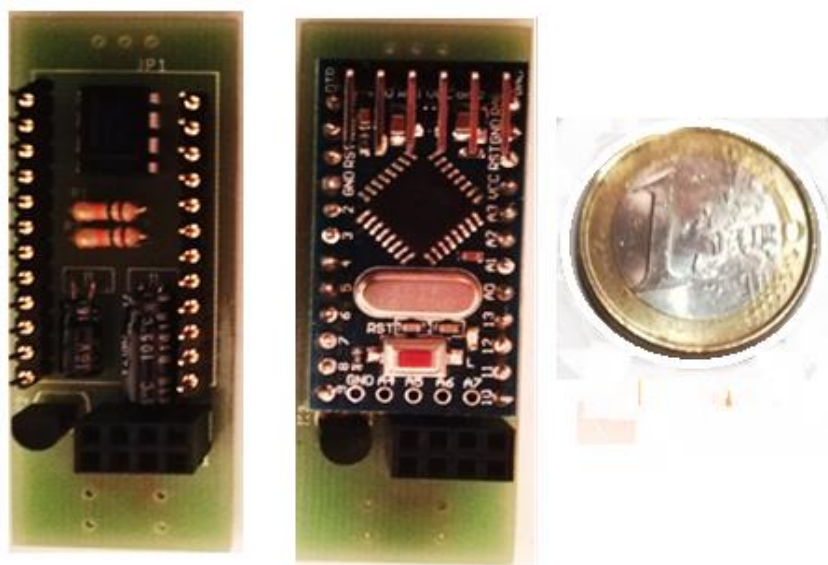


Figura 4.25 - Controlador DMX sem fios terminal.

4.4.5 Programação desenvolvida

A programação desenvolvida neste controlador centrou-se no controlo da receção dos pacotes de dados do DMX. Atendendo às especificações pretendidas, foi efetuada a programação do microcontrolador de modo a providenciar o englobamento de todos os elementos periféricos e efetuar, em função dos mesmos, o controlo das comunicações do sistema.

A estrutura lógica resultante encontra-se no diagrama presente na Figura 4.26.

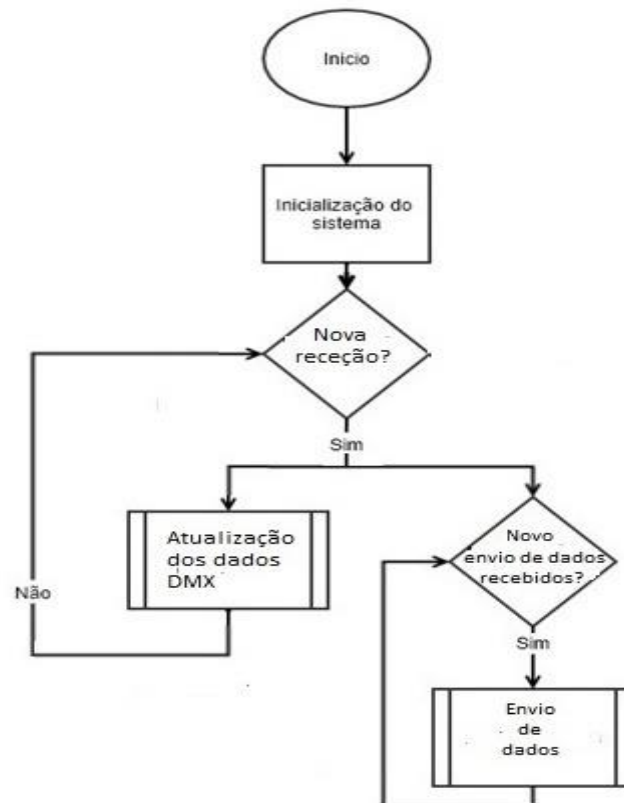


Figura 4.26 - Diagrama de funcionamento do controlador DMX sem fios terminal.

Como é possível observar no diagrama de funcionamento (Figura 4.26) é efetuada a inicialização do sistema e respetivas variáveis, uma vez detetada nova reação rádio o controlador formata os dados fragmentados recebidos e procede ao seu envio para o conversor RS-485. Neste controlador também foram adicionadas várias funções de retransmissão uma vez que estes, têm de responder ao pedido de identificação do controlador principal.

4.5 Sumário

Neste capítulo foi exposto o desenvolvimento de três sistemas embebidos gerados, referindo a perspetiva geral e o funcionamento genérico de cada dispositivo. Em cada subcapítulo foram incluídos diagramas permitindo, o enquadramento das informações neles referidas assim como a perceção visual de funcionamento. Nos subcapítulos referentes ao desenvolvimento de *hardware* foram incluídas descrições dos diversos componentes utilizados, atendendo às suas características funcionais.

5 SOFTWARE DE DESENVOLVIMENTO E CONTROLO DMX

Neste capítulo, é efetuada uma breve descrição das diversas ferramentas de controlo DMX512, nomeadamente as características do *software* de programação e as ferramentas auxiliares complementares deste.

Desta forma pretende-se demonstrar o funcionamento e o desenvolvimento exigidos para a realização dos testes dos controladores criados aplicados a um um projeto de iluminação profissional.

Também neste capítulo, será abordado um dos *softwares* utilizado no desenvolvimento deste projeto.

A título introdutório, é descrita a sua história e evolução num mercado bastante competitivo, bem como, uma breve descrição de conceitos, equipamentos e o seu funcionamento.

5.1 Notas introdutórias

Neste capítulo, é descrito de forma introdutiva o *software* utilizado e o seu funcionamento, de forma a perceber os estágios de desenvolvimento de um projeto audiovisual, explorando as diversas ferramentas presentes em diversos *softwares* influentes nesta área.

As funções de destaque disponibilizadas num *Software* de controlo DMX são as que se descrevem de seguida:

- ✓ *Interfaces* com utilizador e com os equipamentos de controlo em tempo real;
- ✓ Criação e gestão da rede de comunicação DMX e das máquinas a controlar;
- ✓ Supervisão e controlo de estados e valores de configuração dos dispositivos a controlar;
- ✓ Permitir a programação e execução de comandos sobre iluminação profissional;
- ✓ Acesso à simulação e visualização virtual da programação desenvolvida;
- ✓ Acesso a ferramentas auxiliares à programação e à estruturação da rede de comunicação;
- ✓ Otimizar o controlo, acesso e execução de recursos de programação.

Face ao exposto, devemos ter em atenção aos possíveis modos de funcionamento de determinados controladores dedicados, nomeadamente, *Program Mode*, *Live Mode* e *Time Code*.

5.1.1 *Program Mode*

Por norma, todos os controladores dedicados de iluminação profissional, integram um modo de programação, este pode ser acessível em tempo real ao nível de *software* ou em alguns casos com recurso a um seletor físico.

Este modo de funcionamento permite ao utilizador um conjunto de funções que não se encontram disponíveis noutros modos de funcionamento, uma vez que reorganiza e adapta o *hardware* e *software* à programação e desenvolvimento. Neste modo de funcionamento é possível criar, abrir e apagar um programa, ou simplesmente continuar com o programa atual.

Permite executar funções de *patch*, e de configuração da rede de comunicação, personalizar acessos e controlo, e desenvolver toda a programação de um determinado evento.

5.1.2 *Live Mode*

Este modo de funcionamento, é destinado à execução em tempo real da programação efetuada e parte do pressuposto da pré-configuração de toda a rede DMX. Por norma, este modo de funcionamento permite maior acessibilidade aos comandos programados e oferece ao utilizador todo o controlo e responsabilidade sobre esta.

5.1.3 *Time Code*

O *Time Code* é um recurso de execução de programação desenvolvida com características temporais predefinidas. Ou seja, este modo de funcionamento, permite ao utilizador desenvolver uma programação sincronizada, a nível temporal, com determinados atos, exigências e/ou efeitos.

O desenvolvimento deste tipo de programação permite a criação funções de iluminação temporizadas, com possível sincronização a áudio e/ou vídeo de forma a poderem ser ativas quando desejado. Este recurso de programação é bastante mais complexo e exige bastante tempo de desenvolvimento.

5.2 *Software de programação e controlo*

As *interfaces* dedicadas ao controlo de iluminação começaram somente com *hardware*, recorrendo a controladores dedicados. Com o decorrer dos tempos, as novas ferramentas de *software* vieram suportar, ou mesmo terminar com este conceito, fornecendo novas *interfaces* gráficas e de controlo, bem como diversas ferramentas auxiliares de visualização, simulação, edição e de programação e controlo de iluminação profissional.

5.2.1 *Hardware e software Avolites*

A Avolites Ltd é uma empresa britânica com sede em *Park Royal*, Londres, de produção de controladores dedicados ao controlo de iluminação e equipamentos de DMX512 para uso na indústria de iluminação profissional e audiovisual.

A Avolites começou como uma empresa de produção de iluminação em 1976, com o propósito de desenvolvimento e criação de *racks dimmer* e ao longo dos anos iniciou o processo de criação e desenvolvimento de diversos controladores dedicados DMX, diversos dispositivos de interface *media* e o desenvolvimento de *softwares* de controlo.

Em 2011, a empresa expandiu-se para o setor de servidores *media* aplicados a tecnologias de entretenimento, pela vinculação com a empresa *Immersive Ltd*, concebendo a Avolites Media.

Na Tabela 5.1, são referenciadas algumas características e recursos do *software* Avolites Titan;

Tabela 5.1 – Principais características de funcionamento do *software* Avolites Titan [70].



Quicksketch

Permite a criação e edição de *icons* personalizados, fáceis de identificar e procurar. Ferramenta de desenho.



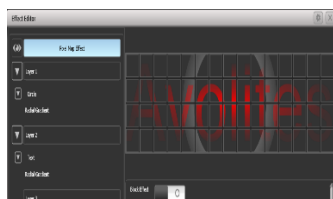
Efect Editor

Permite a criação de efeitos e a sua aplicação simultânea. Possibilidade de guardar efeitos, *Cue's*, *palettes*, macros e importação direta para outros programas.



CLIP Select

Integração e reprodução de *mídia* e vídeo, com capacidade multicamada e pré-visualização. Biblioteca de imagens e vídeos, de fácil implementação e reprodução.



Pixel Mapper tools & Pixel Mapper palettes

Quicksketch Pixel Mapper

Text in Pixel Mapper

Painel de ferramentas de mapeamento de pixels e algoritmos de controlo e animação das mesmas. Ferramenta de desenvolvimento com interface de visualização.



Cue List View

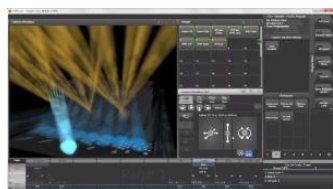
Set List

Ajuste em tempo real de visualização gráfica e sua edição. Capacidade de recorte, cópia e manipulação de objetos em tempo real.



Group Intensity Masters

Potenciômetros digitais rápidos e precisos. Capacidade de edição e alteração de parâmetros de controlo e sua organização.

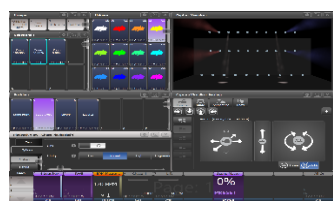


Show Import

Show Save

Show Reports

Capacidade de criação, edição, guarda e importação de programas, macros, *palettes* e efeitos desenvolvidos com facilidade e rapidez.



Keystone and Window control

Ajuste direto e personalizado do tamanho das janelas de interface e sua manipulação em tempo real.

5.2.2 Softwares para Computadores

Porém para um computador comum, a velocidade média deste tipo de porta situa-se entre 4.800 bauds e os seus múltiplos, chegando a atingir os 115.200 bauds. Já para o DMX a velocidade é de 250.000 bauds. Com esta velocidade cada fração da informação, *bit* de dados, tem um tempo de duração de apenas 4 microssegundos, o que revela bastante complexidade na adaptação de uma porta série do computador para o controlo direto do sinal DMX. Até porque hoje em dia é raro encontrar um computador com porta série (RS-232). Como referido, o meio mais simples de controlar um equipamento via DMX é através de uma mesa de controlo, na qual existem potenciômetros deslizantes, e ainda contam com uma memória de repetição, assim sendo as sequências podem ser reproduzidas automaticamente. Em face do amplo uso de computador, para executar tarefas cada vez mais básicas, não seria diferente com a iluminação profissional, no entanto como já comentado, não existindo a porta série (RS-232), utiliza-se a porta USB, com recurso a um conversor.

Existem diversos *softwares* de controlo de interface DMX para PC, como o FreeStyler DMX, o DMX control, GrandMA OnPC ou a Avolites Titan One. Estes *softwares* são muito amplos e úteis, e chegam a possuir inclusive, ferramentas de simulação de ambientes 3D, para possibilitar a criação e desenvolvimento de esboços fiéis de futuros eventos e na sua sequência de efeitos.

5.2.2.1 Avolites Titan One

O *software* Avolites Titan One permite ao utilizador possuir todas as capacidades de uma mesa de iluminação profissional, assente no PC particular. Esta é a solução mais económica do fabricante e efetua todas as tarefas associadas ao controlo de uma consola física de iluminação, uma vez que apenas se trata de uma interface USB para controlo de DMX512. Na Figura 5.1 é possível observar a *Dongle* Avolites Titan One.



Figura 5.1 – *Dongle* Avolites Titan One [70].

A *Dongle* USB Titan One permite ao utilizador controlar uma linha de DMX512 com todas as funcionalidades do *Software* Avolites Titan, aplicados num computador de sistemas *Microsoft Windows* com possibilidade de uso de ambientes *touch screen*. Este *software*, como visto anteriormente, é uma ferramenta rápida e versátil, com recursos poderosos de controlo e programação como as ferramentas *Pixelmapper*, *Quicksketch*, *Media Clip Select*, *Attribute Control-Blades* e *Shape Generator*. O *software* Avolites Titan é livre e pode ser instalado em qualquer computador, no entanto, sem a interface física, este não realiza qualquer função de controlo ou programação, com esta, funciona sem restrições ou limites de tempo de uso [70].

O *software* Avolites Titan apresenta, entre outras, as seguintes características:

- ✓ Rápido e intuitivo;
- ✓ *Software* de controlo com atualizações periódicas;
- ✓ Solução económica e fiável;
- ✓ 60 páginas de 10 *playbacks* controláveis e editáveis;
- ✓ Ferramenta de mão livre Avo 'QuickSketch' para fácil abertura e visualização;
- ✓ Uma única saída DMX512, (1 universo DMX com 512 canais);
- ✓ Grupo de ferramentas fáceis e intuitivas;
- ✓ Suporte de operação em *screen multi touch*;
- ✓ Dimensões (mm): 120 x 90 x 25;
- ✓ Peso: 120g.

5.3 Ferramentas auxiliares de trabalho

5.3.1 Avolites Titan Simulator

Avolites Titan Simulator é uma extensão de desenvolvimento gráfico do programa Avolites Titan. Este *software* permite o desenvolvimento gráfico do ambiente cénico e, consequentemente, o pré-desenvolvimento da programação do evento. Uma ferramenta mediana ao nível do desenvolvimento gráfico, mas que pode exercer um excelente desempenho aquando do uso concomitante no desenvolvimento partilhado com o *software* Avolites.

5.3.2 Avolites Titan Visualiser

O Avolites Visualiser é uma ferramenta de definição de instalações de iluminação cénica em termos de equipamentos e estruturas. Pode funcionar de forma autónoma com qualquer consola de iluminação DMX, no entanto algumas funções específicas e de melhor desempenho só são possíveis com a intercomunicação direta a controladores Avolites. O Visualiser, acima de tudo, foi projetado para ser fácil e rápido de usar, permitindo ao utilizador a edição e experimentação de ambientes cénicos desenvolvidos e a posterior realização de programação e ajuste. Esta ferramenta permite a redução no tempo de desenvolvimento e o seu custo associado.

5.3.3 Avolites Titan Remote

O Avolites Titan Remote é uma aplicação destinada a dispositivos *IOS* ou *Android* que permite o acesso e o controlo de iluminação remoto a um controlador Avolites Titan, recorrendo à configuração de uma rede *Wi-Fi*. Esta aplicação encontra-se disponível gratuitamente no *iTunes* ou na *Google Play Store* e deve ser considerada a versão correspondente à versão do controlador *Titan* utilizado. A utilização desta aplicação permite a comunicação remota com os controladores Avolites Titan Mobile ou Avolites Titan One para PC e permite, ainda, executar as seguintes funções:

- ✓ Exibir o *patch* criado e os respetivos endereços DMX e selecionar e controlar *Fixtures*;
- ✓ Modificar atributos de fixação da máquina, *Dimmer*, *Pan* ou *Tilt*;
- ✓ Criar, atualizar e aplicar *Palettes*;
- ✓ Executar *Cues*;
- ✓ Controlar a intensidade e tipo de comando;
- ✓ Criar legendas rápidas de esboço.

Um controlador Titan pode estabelecer comunicação com inúmeros dispositivos de controlo remoto mesmo quando o utilizador se encontra a realizar operações neste. Esta aplicação inclui um modo de demonstração para permitir que o utilizador experiencie a interface, sem requerer de forma prévia, uma conexão a um controlador Avolites Titan.

5.3.4 Avolites Personality Builder

O Avolites Personality Builder consiste numa aplicação de criação, edição e desenvolvimento de personalidades de máquinas inerentes ao protocolo DMX desenvolvido para ambientes *Avolites*, criada pelo fabricante em 2006 [70]. Em termos de licença, este *software*, possui diversos pacotes, de acordo com o número de variáveis que serão utilizadas no sistema. Esta aplicação é composta por três ferramentas de desenvolvimento e pela aplicação de execução do sistema.

Com este *software* é possível criar ou editar as personalidades das máquinas de forma a exercer o modo de funcionamento desejado, sempre que estas sejam coerentes com as características de *software* e *hardware* das máquinas em questão. Esta é uma ferramenta útil, no desenvolvimento de personalidades de máquinas projetadas, ou na alteração pontual ou integral de máquinas desenvolvidas por construtores desconhecidos.

No caso de não se ser detentor do manual de instruções e características, ou conhecedor do funcionamento de uma determinada máquina, pode-se verificar a distribuição unitária dos canais e com base no funcionamento destes, desenvolver uma personalidade conforme os canais ocupados.

5.4 Sumário

Neste capítulo encontra-se uma breve descrição acerca do *Software* alvo de estudo e aprendizagem, onde são referidos os principais aspetos e funcionalidades dos mesmos. Este apresenta uma enorme facilidade de programação e grande versatilidade de configuração. As ferramentas auxiliares que introduz são muito úteis e de fácil aprendizagem.

6 CONCLUSÕES

O presente projeto de investigação intitulado Comunicação Sem Fios DMX512 teve como objetivo descrever o trabalho desenvolvido durante a execução do mesmo. Assim, foram apresentados detalhadamente os principais projetos desenvolvidos atendendo às diversas etapas da sua execução.

A realização deste projeto de investigação permitiu o desenvolvimento e a consolidação de muitos dos conhecimentos adquiridos ao longo do percurso académico e profissional. Proporcionou ainda a oportunidade de adquirir novos conhecimentos nas diversas áreas abordadas durante a realização dos projetos retratados neste documento. É ainda de realçar a extrema importância de todos os conhecimentos e experiência adquirida profissionalmente, realizada de forma paralela e em parte responsável pela minha progressão na formação no ensino superior.

Permitiu ainda o contacto direto com a indústria e comércio, criando uma nova perspetiva na abordagem com colegas, clientes e outros profissionais, atendendo aos objetivos a atingir de modo a alcançar os resultados pretendidos.

Neste capítulo são apresentadas as conclusões referentes aos diversos projetos realizados, possíveis trabalhos futuros e as conclusões gerais referentes à realização deste projeto. Este capítulo encontra-se então distribuído pelos seguintes subcapítulos:

- **Subcapítulo 6.1** - Conclusões relativas ao desenvolvimento dos projetos;
- **Subcapítulo 6.2** - Trabalhos futuros;

6.1 Conclusões relativas ao desenvolvimento dos projetos

A realização dos projetos foi possível, não só através da aplicação de muitos conhecimentos adquiridos ao longo da vida académica, mas também através da aquisição de novos conhecimentos e de todo o trabalho desenvolvido. Destes conhecimentos, atendendo ao trabalho realizado, destacam-se os conhecimentos na área da eletrónica, programação e das comunicações.

Embora a análise e seleção do protocolo e respetivos equipamentos, pudesse de certa forma, ter sido mais abrangente e profunda, através da realização de testes e análise de configurações internas. O trabalho realizado constituiu uma grande etapa no desenvolvimento deste projeto e absorveu uma grande parte do tempo do seu desenvolvimento.

O controlador principal desenvolvido para suporte da comunicação DMX sem fios, cumpriu com os requisitos inicialmente definidos tendo, quer o *hardware* quer o *software* sido alvo de várias melhorias ao longo do processo de desenvolvimento. A inserção da interface com o utilizador permitiu e potencia uma melhor aplicabilidade do sistema desenvolvido e realça as vantagens que este sistema pode suportar, mas ainda requer de vastas melhorias e testes complementares. A construção do módulo retransmissor permite um maior alcance na comunicação e pode ajudar a estabelecer no futuro, várias funções de controlo e gestão, dos

dados a retransmitir. Este ajudou a atingir a meta de maiores distâncias entre módulos de comunicação.

O trabalho realizado com os módulos recetores, encontram-se ainda distante da perfeição objetivada, mas considerando o tempo de desenvolvimento, e possíveis futuras otimizações, ao nível de programação, eletrónica e estética, podem atingir um grande potencial. De referir que estes fazem parte de uma segunda fase do projeto, em que o seu término não era completamente expectável. Relativamente ao estudo e contato obtido com os distintos *softwares* e às suas ferramentas de suporte, foi absorvido uma grande quantidade de conhecimentos e a sua seleção e aprendizagem foram indispensáveis para a realização do presente projeto.

Desta forma, acredita-se que a escolha por esta metodologia, tecnologia e o rumo que foi tomado neste projeto, foi bastante aceitável e produziu resultados satisfatórios.

Em suma, e analisando de forma integral este projeto, verificou-se que apesar das dificuldades e alguns ajustes a realizar, foi obtido um excelente resultado uma vez que o sistema criado apresenta uma latência residual, um muito bom alcance e uma satisfatória fiabilidade.

6.2 Trabalhos futuros

Neste subcapítulo são referidos os aspetos dos projetos realizados que podem ser sujeitos a desenvolvimentos posteriores, de modo a melhorar a sua aplicabilidade e funcionalidade no desempenho das tarefas a que se destinam.

Relativamente ao controlador principal existe a necessidade de reestruturação do *hardware* utilizado, de forma a desenvolver de raiz uma PCB dedicada e que cumpra com todos os requisitos exigidos. Existe também a intenção de aplicar este sistema numa estrutura mecânica robusta e fácil logística, como por exemplo uma mala profissional.

Ao nível da programação do HMI é objetivada uma função de análise dinâmica da ocupação dos canais e respetivas larguras de bandas e sua comutação adaptativa, de forma a possibilitar a comutação dos mesmos e permitir uma otimização do sistema de comunicação, nomeadamente a criação de uma página de configuração e visualização do estado da rede de comunicação. Ou ainda a hipótese de criar uma página para gravar configurações e sessões de utilização. Desta forma permitir ao utilizador configurar parâmetros da rede de acordo com as condições pretendidas.

O retransmissor apresenta excelentes resultados, no entanto, é equacionado para trabalhos futuros, o estudo do desenvolvimento de uma nova placa PCB que permita a colocação de duas unidades de módulos nRF24L01 de forma a verificar a possibilidade de otimização, através da separação física do elemento recetor do elemento retransmissor, e a conseqüente separação de canais de comunicação. Contudo, foi pensado para implementação futura, a inserção de um *switch* do controlo, de forma a habilitar a inversão do módulo RS-485. Ou seja, permitir através deste, a comutação de receção ou transmissão de forma a permitir, quer o funcionamento como retransmissor com e sem fios, como permitir o seu inverso. Receber a comunicação por cabo e retransmitir esta via rádio. Desta forma otimizar de forma integral a versatilidade da comunicação desenvolvida. Ao nível de programação seria interessante perceber o funcionamento da rede com vários dispositivos retransmissores.

Ao nível global, deve ser efetuado o estudo de otimização de consumo, para perceber o funcionamento deste sistema com recurso a baterias de forma a resolver os problemas de filtragem detetados e a permitir maior versatilidade. Além disto, analisar a possibilidade de implementação de sistema de diagnóstico e estados em tempo real da ocorrência de falhas da rede e verificações de pacotes. Além de todas as otimizações mencionadas e envolvendo uma reestruturação integral do projeto desenvolvido, principalmente ao nível do controlador emissor principal, é ambicionada a expansão da rede de forma a permitir universos paralelos e o desenvolvimento mais cuidado de placas mais compactas e de melhor eficiência.

7 BIBLIOGRAFIA

- [1] A. S. S. D. Committee, “FIA-International Federation of Actors,” [Online]. Available: http://fia-actors.com/no_cache/media/news/news-detail/article/2016-study-analysis-of-the-eu-audiovisual-sector-labour-market-and-of-changing-forms-of-employment-and-work-arrangements/. [Acedido em 4 Setembro 2017].
- [2] W.-D. b. W. S. S. AB, “Live Production,” [Online]. Available: <http://www.live-production.tv/news/cultural-events/w-dmx%E2%84%A2-vangelis-concert-doha.html>. [Acedido em 20 Agosto 2017].
- [3] SIRS-E, “SIRS-E,” [Online]. Available: <https://sirs-e.com/general/wireless-dmx-applications-overview/>. [Acedido em 27 Agosto 2017].
- [4] Desconhecido, “Amazon,” [Online]. Available: <https://www.amazon.co.uk/Wireless-DMX512-Transmitter-wireless-receiver/dp/B00GSF1QJU>. [Acedido em 29 Agosto 2017].
- [5] Rortec, “Rortec,” [Online]. Available: <http://rortec.com.br/produto/transmissor-de-audio-e-video-sem-fio-dmx-512-wireless/57132>. [Acedido em 27 Agosto 2017].
- [6] USITT, “USITT,” [Online]. Available: http://www.usitt.org/standards/DMX512_FAQ.html. [Acedido em 27 Setembro 2017].
- [7] P. USITT, “Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/DMX512>. [Acedido em 27 Setembro 2017].
- [8] PLASA, “ANSI E1.11-2008,” [Online]. Available: http://www.witchesandangels.nl/public_files/documents/ANSI_E1-11_USITT_DMX512-A.pdf. [Acedido em 28 Agosto 2017].
- [9] J. Eade, *The DMX512-A Handbook Design and Implementation of DMX Enabled Products and Networks*, Entertainment Technology Press, 2013.
- [10] T. Instruments, “RS-422 and RS-485 Standards Overview and System,” [Online]. Available: <http://www.ti.com/lit/an/slla070d/slla070d.pdf>. [Acedido em 29 Setembro 2017].
- [11] SmartWorks, BB;, “RS-422 and RS-485 Applications eBook,” em *A Practical Guide to Using RS-422 and RS-485 Serial Interfaces*, 2010, pp. 8-34.

- [12] Softelectro, “Softelectro,” [Online]. Available: http://www.softelectro.ru/rs485_en.html. [Acedido em 29 Setembro 2017].
- [13] L. Bies, “Lammert Bies,” [Online]. Available: <https://www.lammertbies.nl/comm/info/RS-485.html>. [Acedido em 29 Setembro 2017].
- [14] R. Matsubara, “Maxivolt Iluminação Profissional,” [Online]. Available: http://www.iar.unicamp.br/lab/luz/ld/C%EAnica/Manuais/guia_para_iniciantes_dmx_basico.pdf. [Acedido em 7 Setembro 2017].
- [15] Paul Dexter e James L. Moody, “Concert Lighting: The Art and Business of Entertainment Lighting,” em *Concert Lighting*, Taylor & Francis, 2017.
- [16] N. Bobsby, “Practical DMX,” Entertainment Technology Press, 2005.
- [17] ESTA(2016), “ESTA,” [Online]. Available: http://tsp.esta.org/tsp/documents/docs/E1-27-1_2006R2016.pdf. [Acedido em 30 Setembro 2017].
- [18] Neutrik, “Farnell,” Neutrik, [Online]. Available: <http://cpc.farnell.com/neutrik/dmx-term-3p-male/dmx-terminator-3-pin-xlr/dp/DP30990>. [Acedido em 20 Setembro 2017].
- [19] PTRobotics, “PTRobotics,” [Online]. Available: <https://www.ptrobotics.com/modulos-dmx/2041-dmx-usb-rx-a8-8-channel-analogue-output-module.html>. [Acedido em 20 Setembro 2017].
- [20] Enttec, “ENTTEC,” [Online]. Available: <https://www.enttec.com/eu/products/controls/accessories/10m-5pin-dmx-cable-m-f/>. [Acedido em 21 Setembro 2017].
- [21] ESTA, “ESTA,” [Online]. Available: http://tsp.esta.org/tsp/documents/docs/E1-27-1_2006R2016.pdf. [Acedido em 21 Setembro 2017].
- [22] ProPlex, “StageSpot,” [Online]. Available: <https://www.stagespot.com/proplex-opto-splitter-2x8-rackmount-2x5-pin-in-8x5-pin-out-ppdds285-265746.html>. [Acedido em 22 Setembro 2017].
- [23] M. Pearce, “MicroChip,” [Online]. Available: <http://ww1.microchip.com/downloads/en/AppNotes/00001659A.pdf>. [Acedido em 30 Setembro 2017].
- [24] W. pedia, “Wikipedia,” [Online]. Available: <https://en.wikipedia.org/wiki/RS-485>. [Acedido em 21 Setembro 2017].
- [25] PicProjects, “PicProjects,” [Online]. Available: <http://picprojects.org.uk/projects/dmx/dmx688/>. [Acedido em 18 Setembro 2017].

- [26] F. Wiki, “Freestylesupport,” [Online]. Available: https://www.freestylersupport.com/wiki/dmx_basics:dmx_timing. [Acedido em 20 Setembro 2017].
- [27] OpenDMX, “OpenLightning,” [Online]. Available: <https://opendmx.net/index.php/DMX512-A>. [Acedido em 28 Setembro 2017].
- [28] A. L. H. Ltd., “Art-Net 4,” em *Specification for the Art-Net 4 Ethernet Communication Protocol*, 2017, pp. 3-7.
- [29] R. Protocol, “RDM Protocol,” [Online]. Available: <http://www.rdmprotocol.org/what-is-it/>. [Acedido em 30 Setembro 2017].
- [30] Wikipedia, “RDM Lightning,” [Online]. Available: [https://en.wikipedia.org/wiki/RDM_\(lighting\)](https://en.wikipedia.org/wiki/RDM_(lighting)).
- [31] ArtNet, “ArtNet,” [Online]. Available: <http://art-net.org.uk/>. [Acedido em 04 Outubro 2017].
- [32] S. M. S. N. K. Leonid A. Belov, “Handbook of RF, Microwave, and Millimeter-wave Components,” em *Handbook of RF, Microwave, and Millimeter-wave Components*, Londres, Artech House, 2012, p. 504.
- [33] ANACOM, “ANACOM,” [Online]. Available: https://www.anacom.pt/streaming/qnaf20092010_07042010.pdf?contentId=1019281&field=ATTACHED_FILE. [Acedido em 2017 Novembro 30].
- [34] CISCO, “Ethernet Technologies,” [Online]. Available: http://docwiki.cisco.com/wiki/Ethernet_Technologies. [Acedido em 20 Outubro 2017].
- [35] CISCO, “Carrier Sense Multi-Access/Collision Detection,” [Online]. Available: http://www.cisco.com/en/US/tech/tk389/tk214/tk125/tsd_technology_support_sub-protocol_home.html. [Acedido em 20 Outubro 2017].
- [36] IPV6, “tcpip-protocol-suite,” [Online]. Available: <https://www.ipv6.com/general/tcpip-protocol-suite/>. [Acedido em 22 Outubro 2017].
- [37] CISCO, “Internetworking Basics,” [Online]. Available: <http://www.cisco.com/cpress/cc/td/cpress/fund/ith/ith01gb.htm#xtocid166847>. [Acedido em 13 Outubro 2017].
- [38] CISCO, “Internet Protocol,” [Online]. Available: http://www.cisco.com/c/en/us/td/docs/net_mgmt/active_network_abstraction/3-7/reference/guide/ANARefGuide37/ip.html. [Acedido em 9 Setembro 2017].

- [39] Wikipedia, “DCF,” [Online]. Available: https://en.wikipedia.org/wiki/Distributed_coordination_function. [Acedido em 13 Outubro 2017].
- [40] L. d. garagem, “SPI Protocol,” [Online]. Available: <http://labdegaragem.com/profiles/blogs/tutorial-comunica-o-spi-serial-peripheral-interface-com-arduino>. [Acedido em 5 Outubro 2017].
- [41] S. fun, “SPI,” [Online]. Available: <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>. [Acedido em 9 Outubro 2017].
- [42] C. M. Gilmore, “Basic Skills in Electricity and Electronics,” em *Introduction to Microprocessors*, McGraw-Hill, 1981, pp. 250-259.
- [43] Uchar, “Unsigned Char,” [Online]. Available: <https://uchar.wordpress.com/2015/08/28/bit-banging-uart/>. [Acedido em 30 Outubro 2017].
- [44] “FT232R - USB UART IC,” FTDIChip, [Online]. Available: <http://www.ftdichip.com/Products/ICs/FT232R.htm>. [Acedido em 20 Outubro 2017].
- [45] S. Heath, “Series for Design Engineers,” em *Embedded Systems Design Second edition*, Newnes, 2003, pp. 1 - 15.
- [46] Marcusjenkins, “Arduino Pinout Diagrams,” [Online]. Available: <http://marcusjenkins.com/arduino-pinout-diagrams/>. [Acedido em 15 outubro 2017].
- [47] Atmel, “ATmega48A/PA/88A/PA/168A/PA/328/P,” [Online]. Available: http://www.atmel.com/Images/Atmel-8271-8-bit-AVR-Microcontroller-ATmega48A-48PA-88A-88PA-168A-168PA-328-328P_datasheet_Complete.pdf. [Acedido em 7 novembro 2017].
- [48] Nordic, “nRF24L01,” [Online]. Available: <https://www.nordicsemi.com/eng/Products/2.4GHz-RF/nRF24L01P>.
- [49] “NRF24L01+,” [Online]. Available: <https://arduino.stackexchange.com/questions/20371/pin-mapping-of-nrf2401l01-using-adafruits-nrf24l01-library>. [Acedido em 2017 Outubro 2017].
- [50] Nordic, “Saprkrfun,” [Online]. Available: https://www.sparkfun.com/datasheets/Components/SMD/nRF24L01Pluss_Preliminary_Product_Specification_v1_0.pdf. [Acedido em 22 Outubro 2017].
- [51] “Arduino-Info,” [Online]. Available: <https://arduino-info.wikispaces.com/RS485-Modules>. [Acedido em 30 Setembro 2017].

- [52] [Online]. Available: <http://www.msseletronica.com/detalhes/modulo-conversor-max485-rs485-para-rs232-ttl-ideal-para-arduino-pic/1097.html>. [Acedido em 30 Setembro 2017].
- [53] L. Llamas. [Online]. Available: <https://www.luisllamas.es/arduino-rs485-max485/>. [Acedido em 30 Setembro 2017].
- [54] Nextion, “Itead,” [Online]. Available: <https://www.itead.cc/nextion-nx8048t070.html>. [Acedido em 6 Novembro 2017].
- [55] Eagle, “AutoDesk,” [Online]. Available: <https://www.autodesk.com/education/free-software/eagle>.
- [56] IPC, “Generic Standard on Printed Board Design,” [Online]. Available: <http://www.ipc.org/TOC/IPC-2221B.pdf>. [Acedido em 29 outubro 2017].
- [57] D. L. Jones, “PCB Design,” em *PCB Design Totutorial*, 2004, pp. 5-25.
- [58] “How to program a microcontroller,” EngineersGarage, [Online]. Available: <http://www.engineersgarage.com/tutorials/how-to-program-a-microcontroller>. [Acedido em 20 Novembro 2017].
- [59] “What are Integrated Development Environments?,” All About Circuits, [Online]. Available: <http://www.allaboutcircuits.com/technical-articles/what-are-integrated-development-environments/>. [Acedido em 12 Novembro 2017].
- [60] M. ESP8266, “Embarcados,” [Online]. Available: <https://www.embarcados.com.br/modulo-esp8266/>.
- [61] “Espressif,” [Online]. Available: http://espressif.com/sites/default/files/documentation/0a-esp8266ex_datasheet_en.pdf. [Acedido em 4 Novembro 2017].
- [62] E. I. Team, “ESP8266,” em *ESP8266 Technical Reference*, 2017, pp. 1-20.
- [63] “Bluethooth Module,” [Online]. Available: <http://thefindom.com/communication-module.html>. [Acedido em 04 Novembro 2017].
- [64] Nordic, “Bluetooth V4.0 BLE,” [Online]. Available: http://infocenter.nordicsemi.com/pdf/nRF51822_PS_v3.3.pdf. [Acedido em 4 Novembro 2017].
- [65] Digi, “ZigBee RF Solutions,” [Online]. Available: <https://www.digi.com/products/xbee-rf-solutions/sub-1-ghz-modules>. [Acedido em 4 Novembro 2017].

- [66] Microchip. [Online]. Available: <http://www.microchip.com/wwwproducts/en/MRF24J40MA>. [Acedido em 4 Novembro 2017].
- [67] “Forum Arduino,” [Online]. Available: <https://forum.arduino.cc/index.php?topic=434270.0>. [Acedido em 19 Novembro 2017].
- [68] “Texas Instruments,” [Online]. Available: <http://www.ti.com/lit/ds/slvs582i/slvs582i.pdf>. [Acedido em 10 Novembro 2017].
- [69] “electronics.stackexchange,” [Online]. Available: <https://electronics.stackexchange.com/questions/100487/dmx-on-arduino-with-rs485>. [Acedido em 20 Outubro 2017].
- [70] A. Media, “Avolites,” [Online]. Available: <https://www.avolites.com/software/latest-version>. [Acedido em 5 Dezembro 2017].
- [71] J. C. T. R. Araújo, “Virtual Reality for Lighting Simulation in Events,” Lisboa, 2014.
- [72] R. Cadena, “Focus on Lighting Technology,” Pro Lights & Staging News, 2002, pp. 130-138.
- [73] “CastSoft,” [Online]. Available: <http://cast-soft.com/wysiwyg-lighting-design/>. [Acedido em 23 Outubro 2017].
- [74] “SaldStudios,” [Online]. Available: <http://www.saldstudios.com/de/nachrichten/karel-gott-tour-2014-pohled-pod-poklicku/>. [Acedido em 23 Outubro 2017].
- [75] L. D. F. Jönsson, “SpotLightWeb,” [Online]. Available: <https://spotlightweb.wordpress.com/2014/04/19/lighting-wysiwyg-perfects-lighting-on-multi-size-stages-at-swedish-melodifestivalen/>. [Acedido em 23 Outubro 2017].
- [76] U. Kar, “Ujjal Kar,” [Online]. Available: <http://www.dmx512-online.com/phys1.html>. [Acedido em 18 Setembro 2017].
- [77] IEFP, “OPAC.IEFP,” [Online]. Available: <http://opac.iefp.pt:8080/images/winlibimg.aspx?skey=&doc=73175&img=1181>. [Acedido em 4 Outubro 2017].
- [78] P. Hindle, “MicroWaveJornal,” [Online]. Available: <http://www.microwavejournal.com/articles/24759>. [Acedido em 30 Outubro 2017].
- [79] “Comrad,” [Online]. Available: https://www.conrad.com/ce/en/content/service_eagle_software/service_eagle_software;

jsessionId=E3A46D4C947EA0A3E86FEE280710E15D.ASTPCEN27. [Acedido em 25 Outubro 2017].

- [80] “DigChip,” [Online]. Available:
<https://www.digchip.com/datasheets/parts/datasheet/477/LP2950-50LPRE3.php>.
[Acedido em 13 Novembro 2017].

Anexo I - Programa do controlador DMX Sem Fios Principal

Neste anexo pretende-se expor, de uma forma simplificada, a estrutura do programa desenvolvido para o controlador principal.

```
/*-----*/
/*----Projeto de Investigação Comunicação Sem Fios DMX512-----*/
/*----Área de Especialização em Automação e Comunicações em Sistemas Industriais---*/
/*----Autor: Tiago Manuel de Oliveira Crespo-----dezembro 2017-----*/
/*-----*/
/*----Declaração das Bibliotecas-----*/

#include <lib_dmx.h>

#include <SPI.h>

#include <nRF24L01.h>

#include <RF24.h>

/*-----Declaração de Constantes e de Pinos-----*/

#define DMX512 (0)

#define CE_PIN 9

#define CSN_PIN 10

byte TxBuffer[520];

byte AuxTxBuffer[24];

byte idMensagem = 0;

const uint64_t pipe[2] = { 0xF0F0F0F066LL, 0xF0F0F0F0ALL}; // Declaração dos Pipes de
Comunicação

bool sendsuccess = 0;

/*-----Declaração de Objetos-----*/

RF24 radio(CE_PIN, CSN_PIN);

void SEND(){// Função Para Envio de Fragmentos DMX Via Rádio

    if(idMensagem >= 199)idMensagem = 0;//Reset ID Mensagem
```

```

        idMensagem++; //Incremento do ID Mensagem

unsigned int ponteiro = 0; //Variavel de Controlo do Número de Canais Enviados

byte idPacote = 0; //Variavel de Controlo do ID do pacote Atual

while(ponteiro < 512){ //Executa Enquanto Não Enviar os 512 Canais

    for(int i=2; i<18; i++){ //Colocação dos Dados DMX no Buffer de Transmissão

        AuxTxBuffer[i]=TxBuffer[ponteiro];

        ponteiro++;

    }

    AuxTxBuffer[0]=idMensagem; //Colocação do ID Mensagem no Buffer de Transmissão

    AuxTxBuffer[1]=idPacote; //Colocação do ID Pacote no Buffer de Transmissão

    idPacote++;

    sendsuccess = radio.write( AuxTxBuffer, 18 ); //Envio dos 18 Bytes em Buffer via Rádio

}

}

void setup()

{

    ArduinoDmx0.set_control_pin(3); // Arduino output pin for MAX485 input/output
control (connect to MAX485 pins 2-3)

    ArduinoDmx0.set_rx_address(1); // set rx0 dmx start address

    ArduinoDmx0.set_rx_channels(512); // number of rx channels

    ArduinoDmx0.init_rx(DMX512); // starts universe 0 as rx, NEW Parameter DMX
mode

    radio.begin();

    radio.setPALevel( RF24_PA_HIGH ); //MIN=-18dBm, LOW=-12dBm, MED=-6dBm,
and HIGH=0dBm ,RF24_PA_MAX

    radio.setDataRate(RF24_2MBPS); //RF24_250KBPS for 250kbs, RF24_1MBPS for
1Mbps, or RF24_2MBPS for 2Mbps

    radio.setCRCLength(RF24_CRC_8); //RF24_CRC_8 for 8-bit or RF24_CRC_16 for
16-bit

    radio.setAutoAck(false);

```

```
    radio.setRetries(1,1);

    radio.setChannel(110);

    radio.openWritingPipe(pipe[1]); //Declaração do Pipe 0 Para Transmissão com
    Endereço "1"

    delay(500);

} //end setup()

void loop()

{

    for(int i=0; i<512; i++){ //Copiar os Dados DMX Recebidos Para o Array de
    Transmissão

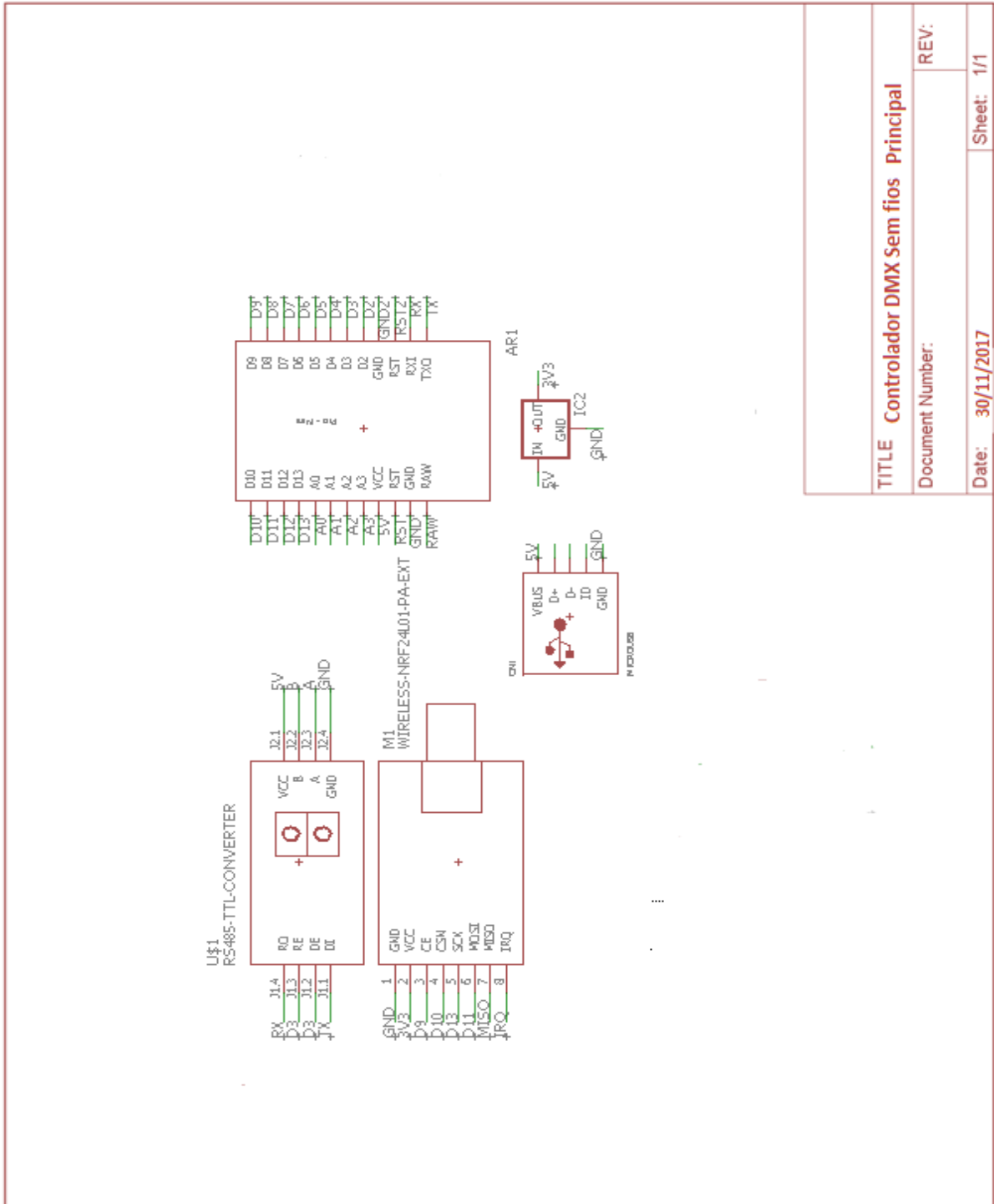
    TxBuffer[i] = ArduinoDmx0.RxBuffer[i];

    }

    SEND();

} //end loop()
```


Anexo II - Esquema elétrico do controlador DMX sem fios principal



TITLE Controlador DMX Sem fios Principal	
Document Number:	REV:
Date: 30/11/2017	Sheet: 1/1

Anexo III - Programa do controlador DMX sem fios Secundário

Neste anexo pretende-se expor, de uma forma simplificada, a estrutura do programa desenvolvido para o controlador do retransmissor.

```

/*-----*/
/*----Projeto de Investigação Comunicação Sem Fios DMX512-----*/
/*----Área de Especialização em Automação e Comunicações em Sistemas Industriais----*/
/*----Autor: Tiago Manuel de Oliveira Crespo-----Dezembro 2017-----*/
/*-----*/

/*-----( Import needed libraries )-----*/
#include <lib_dmx.h> // comment/uncomment #define USE_UARTx in lib_dmx.h as needed

#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

/*-----( Declare Constants and Pin Numbers )-----*/
#define DMX512 (0) // (250 kbaud - 2 to 512 channels) Standard USITT DMX-512
#define CE_PIN 9
#define CSN_PIN 10

// NOTE: the "LL" at the end of the constant is "LongLong" type
const uint64_t pipe[2] = { 0xF0F0F0F066LL, 0xF0F0F0F0ALL }; // Define the receive pipe

/*-----( Declare objects )-----*/
RF24 radio(CE_PIN, CSN_PIN); // Create a Radio

/*-----( Declare Variables )-----*/
byte RxBuffer[512];
byte AuxRxBuffer[18];
byte ID_Pacote_Act = 0;
byte ID_Mensagem_Act = 0;
byte ID_Mensagem = 0;

bool RxBufferReady = 0;

```

```

bool sendsuccess = 0;

bool done = false;

void setup() /***** SETUP: RUNS ONCE *****/
{
    ArduinoDmx0.set_control_pin(3); // Arduino output pin for MAX485 input/output
    control (connect to MAX485-1 pins 2-3)

    ArduinoDmx0.set_tx_address(1); // set rx1 start address

    ArduinoDmx0.set_tx_channels(512); // 2 to 2048!! channels in DMX1000K (512 in
    standard mode) See lib_dmx.h *** new *** EXPERIMENTAL

    // New parameter needed: DMX Mode

    ArduinoDmx0.init_tx(DMX512); // starts universe 1 as rx, standard DMX 512 - See
    lib_dmx.h, now support for DMX faster modes (DMX 1000K)

    radio.begin();

    radio.setPALevel( RF24_PA_HIGH ) ; //MIN=-18dBm, LOW=-12dBm, MED=-
    6dBm, and HIGH=0dBm

    radio.setDataRate(RF24_2MBPS);//RF24_250KBPS for 250kbs, RF24_1MBPS for
    1Mbps, or RF24_2MBPS for 2Mbps

    radio.setCRCLength(RF24_CRC_8);//RF24_CRC_8 for 8-bit or RF24_CRC_16 for
    16-bit

    radio.setAutoAck(false);

    radio.setRetries(1,1);

    radio.setChannel(108);

    radio.openWritingPipe(pipe[0]);

    radio.openReadingPipe(1,pipe[1]);

    radio.startListening();

}/--(end setup )---

void loop() /***** LOOP: RUNS CONSTANTLY *****/
{
    if ( radio.available()
    {
        while (radio.available())
        {
            radio.read( AuxRxBuffer, 18 );
        }
    }
}

```


Anexo IV - Programa do controlador de DMX sem fios terminal

Neste anexo pretende-se expor, de uma forma simplificada, a estrutura do programa desenvolvido para o controlador terminal.

```

/*-----*/
/*----Projeto de Investigação Comunicação Sem Fios DMX512-----*/
/*----Área de Especialização em Automação e Comunicações em Sistemas Industriais----*/
/*----Autor: Tiago Manuel de Oliveira Crespo-----Dezembro 2017-----*/
/*-----*/

/*-----Declaração das Bibliotecas-----*/
#include <lib_dmx.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>

/*-----Declaração de Constantes e de Pinos-----*/
#define DMX512 (0)
#define CE_PIN 9
#define CSN_PIN 10

const uint64_t pipe[2] = { 0xF0F0F0F066LL, 0xF0F0F0F0ALL}; // Declaração dos Pipes de
Comunicação

/*-----Declaração de Objetos-----*/
RF24 radio(CE_PIN, CSN_PIN); // Declaração da Comunicação Rádio

/*-----Declaração de Variáveis-----*/
byte RxBuffer[512] ;// Buffer de Dados DMX
byte AuxRxBuffer[18] ;// Buffer dos dados de receção Rádio
byte ID_Pacote_Act = 0; // Identificador do ID do Pacote Atual
byte ID_Mensagem_Act = 0; // Identificador do ID da Mensagem Atual
bool RxBufferReady = 0; // Flag de Nova Receção Via Rádio

void setup() /****** SETUP*****/
{

```

```

    ArduinoDmx0.set_control_pin(3); // Atribuição da Saída 3 Como pino de Controlo de
    Receção/Transmissão do MAX485

    ArduinoDmx0.set_tx_address(1); // Configuração do Primeiro Endereço DMX
    ArduinoDmx0.set_tx_channels(512); // Configuração do Numero de Canais DMX
    ArduinoDmx0.init_tx(DMX512); // Inicialização da Comunicação DMX

    radio.begin();// Inicialização da Comunicação Rádio

    radio.setPALevel( RF24_PA_HIGH ) ; //MIN=-18dBm, LOW=-12dBm, MED=-6dBm
    e HIGH=0dBm - Configuração da Potência de Transmissão Rádio

    radio.setDataRate(RF24_2MBPS);//RF24_250KBPS para 250kbs, RF24_1MBPS para
    1Mbps e RF24_2MBPS para 2Mbps - Configuração da Velocidade de Transmissão
    Rádio

    radio.setCRCLength(RF24_CRC_8);//RF24_CRC_8 para 8-bit e RF24_CRC_16 para
    16-bit - Configuração do CRC

    radio.setAutoAck(false);//Configuração do Auto Acknowledge

    radio.setRetries(1,1);//Configuração do Numero de Retransmissões e do Timeout em
    caso de Falha

    radio.setChannel(110);//Configuração do Canal de Funcionamento Rádio

    radio.openReadingPipe(1,pipe[0]);//Declaração do Pipe 1 para Leitura com endereço
    "0"

    radio.startListening();//Inicialização da Leitura
}

//--Fim setup---

void loop() /******LOOP*****/
{
    if ( radio.available() ) //Verificação da Existência de Dados Rádio Para Receção
    {
        while (radio.available())
        {
            radio.read( AuxRxBuffer, 18 ); //Leitura dos 18 Bytes para o Buffer de Receção
        }

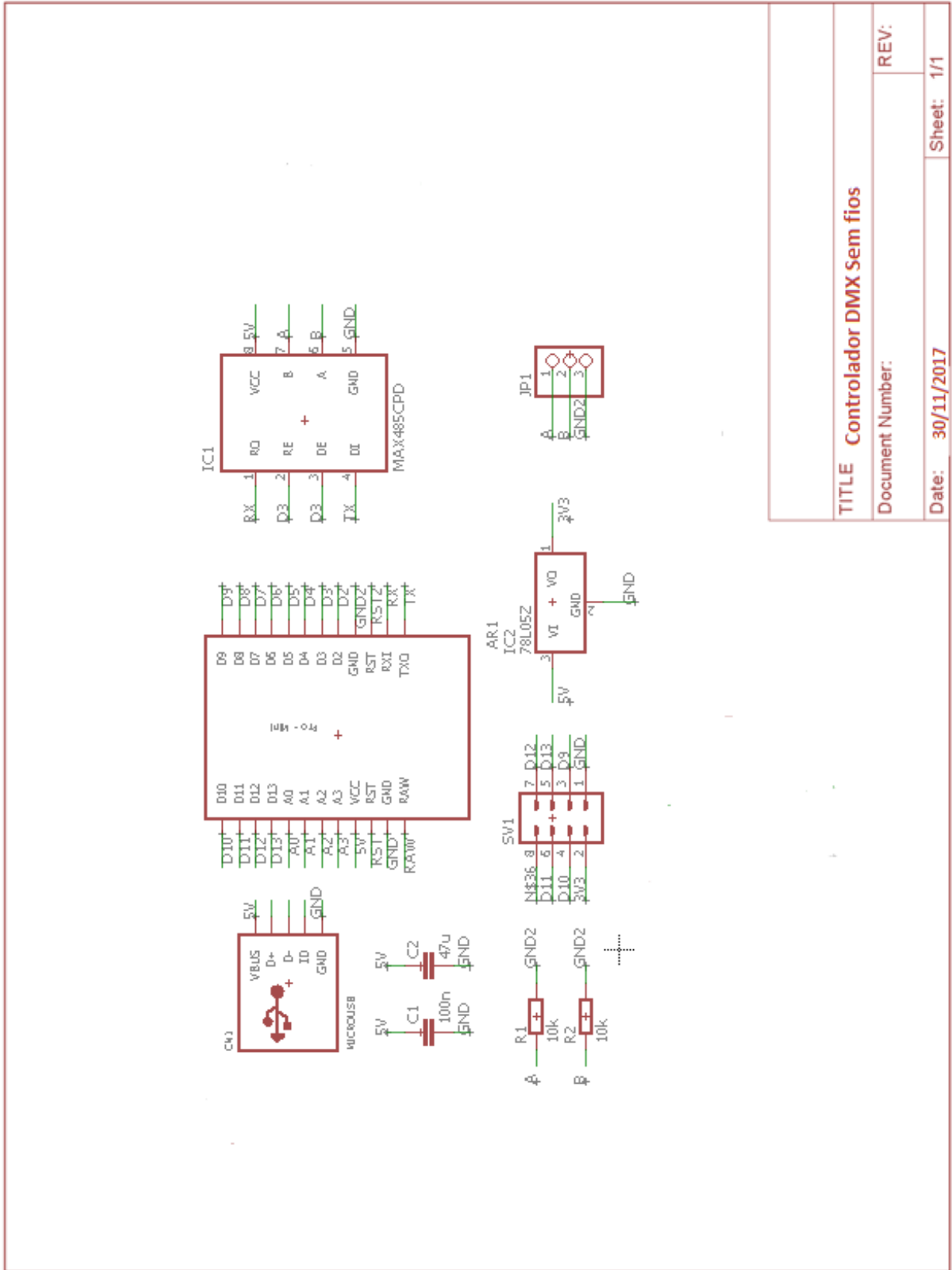
        for(int j=0; j < 16; j++){ //Atualização do Array de Dados DMX
            RxBuffer[(AuxRxBuffer[1] * 16)+j]=AuxRxBuffer[j+2]; //Colocação dos Dados
            Recebidos na Respetiva Posição do Array DMX
        }

        ID_Pacote_Act = AuxRxBuffer[1];
    }
}

```

```
    RxBufferReady = 1;//Ativação da Flag Para Efetuar a Atualização da Saída DMX
}
if(RxBufferReady){
    for(int k=0; k<512; k++){//Atualização do Array de Saida DMX
        ArduinoDmx0.TxBuffer[k]=RxBuffer[k];
    }
    RxBufferReady = false;
}
}//--Fim loop---
```


Anexo V - Esquema elétrico do controlador DMX sem fios terminal



TITLE Controlador DMX Sem fios

Document Number:

REV:

Date: 30/11/2017

Sheet: 1/1

Anexo VI - Programa do mapeamento controlador de DMX sem fios terminal

Neste anexo pretende-se expor, de uma forma simplificada, a estrutura do programa desenvolvido para o mapeamento HMI do controlador principal.

```
/*-----*/
/*----Projeto de Investigação Comunicação Sem Fios DMX512-----*/
/*----Área de Especialização em Automação e Comunicações em Sistemas Industriais---*/
/*----Autor: Tiago Manuel de Oliveira Crespo-----Dezembro 2017-----*/
/*-----*/

#include <lib_dmx.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include "Nextion.h"
#include <SoftwareSerial.h>
SoftwareSerial SoftySerial(4, 5); // RX, TX

#define DMX512 (0) // (250 kbaud - 2 to 512 channels) Standard USITT DMX-512
#define CE_PIN 9
#define CSN_PIN 10
byte TxBuffer[520];
byte AuxTxBuffer[24];
byte AuxRxBuffer[24];
byte BufferModulos[250];
int NumeroModulos = 0;
byte idMensagem = 0;
byte ID_L = 0;
byte ID_H = 0;
const uint64_t pipe[4] = { 0xF0F0F0F066LL, 0xF0F0F0F0ALL, 0xEEFAFDFDEELL,
0xEEFDFAF50DFLL}; // Define the transmit pipe
bool sendsuccess = 0;
bool init_mapping = 0;
bool confirm_init_mapping = 0;
```

```
int time_init_mapping = 30000;
unsigned long time_millis_init_Mapping = 0;
bool Mapping_Step_1 = 0;
int ModuloAtual = 0;
int Endereco_Maquina = 0;
int Endereco_Mesa = 0;
RF24 radio(CE_PIN, CSN_PIN);
#####HMI
void InitMapping(void *ptr);
void InitEnderecamento(void *ptr);
void FimEnderecamento(void *ptr);
void MaquinaAnterior(void *ptr);
void MaquinaSeguinte(void *ptr);
void GravarConfigMaquina(void *ptr);
void DesabilitarEnderecamento(void *ptr);
NexButton b0 = NexButton(2, 2, "b0"); //InitMapping
NexButton b1 = NexButton(2, 3, "b1");//InitEnderecamento
NexButton b2 = NexButton(2, 4, "b2");//FimEnderecamento
NexButton b3 = NexButton(2, 5, "b3");//MaquinaAnterior
NexButton b4 = NexButton(2, 6, "b4");//MaquinaSeguinte
NexButton b5 = NexButton(2, 7, "b5");//GravarConfigMaquina
NexButton b6 = NexButton(2, 8, "b6");//Desabilitar Modo Endereçamento
NexNumber n0 = NexNumber(2, 9, "n0");//ModuloAtual
NexNumber n1 = NexNumber(2, 10, "n1");//Endereco_Maquina
NexNumber n2 = NexNumber(2, 11, "n2");//Endereco_Mesa
NexTouch *nex_listen_list[] =
{
    &b0,
    &b1,
    &b2,
    &b3,
    &b4,
    &b5,
```

```
&b6,  
&n0,  
&n1,  
&n2,  
NULL  
};  
void InitMapping(void *ptr)  
{  
    init_mapping = 1;  
    time_millis_init_Mapping = millis();  
    for(int i=0; i<250; i++){  
        BufferModulos[i]=0;  
    }  
    NumeroModulos = 0;  
}  
void InitEnderecamento(void *ptr)  
{  
    Mapping_Step_1 = 1;  
}  
void FimEnderecamento(void *ptr)  
{  
    Mapping_Step_1 = 0;  
}  
void MaquinaSeguinte(void *ptr)  
{  
    if(ModuloAtual + 1 < NumeroModulos)ModuloAtual++;  
    else ModuloAtual = 0;  
    Endereco_Maquina = Endereco_Maquina | BufferModulos[(ModuloAtual * 10)+ 3] <<  
8;//Byte Mais significativo  
    Endereco_Maquina = Endereco_Maquina | BufferModulos[(ModuloAtual * 10)+ 2];//Byte  
Menos significativo  
    Endereco_Mesa = Endereco_Mesa | BufferModulos[(ModuloAtual * 10)+ 5] << 8;//Byte  
Mais significativo
```

```
Endereco_Mesa = Endereco_Mesa | BufferModulos[(ModuloAtual * 10)+ 4]; //Byte Menos
significativo
```

```

n0.setValue(ModuloAtual);
n1.setValue(Endereco_Maquina);
n2.setValue(Endereco_Mesa);
}
void MaquinaAnterior(void *ptr)
{
    if(ModuloAtual > 0)ModuloAtual--;
    else ModuloAtual = NumeroModulos - 1;
    Endereco_Maquina = Endereco_Maquina | BufferModulos[(ModuloAtual * 10)+ 3] <<
8; //Byte Mais significativo
    Endereco_Maquina = Endereco_Maquina | BufferModulos[(ModuloAtual * 10)+ 2]; //Byte
Menos significativo
    Endereco_Mesa = Endereco_Mesa | BufferModulos[(ModuloAtual * 10)+ 5] << 8; //Byte
Mais significativo
    Endereco_Mesa = Endereco_Mesa | BufferModulos[(ModuloAtual * 10)+ 4]; //Byte Menos
significativo

n0.setValue(ModuloAtual);
n1.setValue(Endereco_Maquina);
n2.setValue(Endereco_Mesa);
}
void GravarConfigMaquina (void *ptr)
{
    uint32_t Aux_Endereco_Maquina;
    uint32_t Aux_Endereco_Mesa;
    ID_L = BufferModulos[ModuloAtual * 10];
    ID_H = BufferModulos[(ModuloAtual * 10) + 1];
    n1.getValue(&Aux_Endereco_Maquina);
    n2.getValue(&Aux_Endereco_Mesa);
    Endereco_Maquina = Aux_Endereco_Maquina;
    Endereco_Mesa = Aux_Endereco_Mesa;
}

```

```
BufferModulos[(ModuloAtual * 10)+ 3] = (Endereco_Maquina >> 8) & 0xFF;//Byte Mais significativo
```

```
BufferModulos[(ModuloAtual * 10)+ 2] = Endereco_Maquina & 0xFF;//Byte Menos significativo
```

```
BufferModulos[(ModuloAtual * 10)+ 5] = (Endereco_Mesa >> 8) & 0xFF;//Byte Mais significativo
```

```
BufferModulos[(ModuloAtual * 10)+ 4] = Endereco_Mesa & 0xFF;//Byte Menos significativo
```

```
radio.stopListening();
radio.openWritingPipe(pipe[0]);
AuxTxBuffer[0] = 204;
AuxTxBuffer[1] = 101;
AuxTxBuffer[2] = ID_L;
AuxTxBuffer[3] = ID_H;
AuxTxBuffer[4] = BufferModulos[(ModuloAtual * 10)+ 2];
AuxTxBuffer[5] = BufferModulos[(ModuloAtual * 10)+ 3];
AuxTxBuffer[6] = BufferModulos[(ModuloAtual * 10)+ 4];
AuxTxBuffer[7] = BufferModulos[(ModuloAtual * 10)+ 5];
for(int i=8; i<18; i++){
AuxTxBuffer[i]=0;
}
sendsuccess = radio.write( AuxTxBuffer, 18 );
radio.startListening();
}
void DesabilitarEnderecamento(void *ptr)
{
radio.stopListening();
radio.openWritingPipe(pipe[0]);
AuxTxBuffer[0] = 205;
AuxTxBuffer[1] = 101;
AuxTxBuffer[2] = 255;
AuxTxBuffer[3] = 255;
for(int i=4; i<18; i++){
```

```
AuxTxBuffer[i]=0;
}
sendsuccess = radio.write( AuxTxBuffer, 18 );
radio.startListening();
}
//#####HMI
void SEND(){
    radio.stopListening();
    radio.openWritingPipe(pipe[0]);
    if(idMensagem >= 199)idMensagem = 0;
    idMensagem++;
    unsigned int ponteiro = 0;
    byte idPacote = 0;
    while(ponteiro < 512){
        for(int i=2; i<18; i++){
            AuxTxBuffer[i]=TxBuffer[ponteiro];
            ponteiro++;
        }
        AuxTxBuffer[0]=idMensagem;
        AuxTxBuffer[1]=idPacote;
        idPacote++;
        sendsuccess = radio.write( AuxTxBuffer, 18 );
        radio.startListening();
    }
}
void Receive(void)
if ( radio.available() )
{
    while (radio.available())
    {
        // Fetch the data payload
        radio.read( AuxRxBuffer, 18 );
    }
}
```

```
    }
    //*****Adição novo modulo
    if(AuxRxBuffer[0] == 211 && AuxRxBuffer[1] == 100){
        bool Modulo_ja_inserido = 0;
        for(int i=0; i<250; i = i + 10){
            if(BufferModulos[i] == AuxRxBuffer[2] && BufferModulos[i + 1] ==
            AuxRxBuffer[3]) Modulo_ja_inserido = 1;
        }
        if(!Modulo_ja_inserido){
            BufferModulos[NumeroModulos * 10] = AuxRxBuffer[2];// ID_L
            BufferModulos[(NumeroModulos * 10) + 1] = AuxRxBuffer[3];// ID_H
            NumeroModulos++;
        }
        radio.stopListening();
        radio.openWritingPipe(pipe[0]);
        AuxTxBuffer[0] = 202;
        AuxTxBuffer[1] = 100;
        AuxTxBuffer[2] = AuxRxBuffer[2];
        AuxTxBuffer[3] = AuxRxBuffer[3];
        for(int i=4; i<18; i++){
            AuxTxBuffer[i]=0;
        }
        sendsuccess = radio.write( AuxTxBuffer, 18 );
        radio.startListening();
    }
    //*****Fim Adição novo modulo
}

void Detect_Modules(void){
    radio.stopListening();
    radio.openWritingPipe(pipe[0]);
    AuxTxBuffer[0] = 201;
    AuxTxBuffer[1] = 100;
    for(int i=2; i<18; i++){
```

```
AuxTxBuffer[i]=0;
}
sendsuccess = radio.write( AuxTxBuffer, 18 )
radio.startListening();
}
void setup()
{
  ArduinoDmx0.set_control_pin(3); // Arduino output pin for MAX485 input/output
  control (connect to MAX485 pins 2-3)
  ArduinoDmx0.set_rx_address(1); // set rx0 dmx start address
  ArduinoDmx0.set_rx_channels(512); // number of rx channels
  ArduinoDmx0.init_rx(DMX512); // starts universe 0 as rx, NEW Parameter DMX
  mode
  radio.begin();
  radio.setPALevel( RF24_PA_HIGH) ;//MIN=-18dBm, LOW=-12dBm, MED=-6dBm,
  and HIGH=0dBm
  radio.setDataRate(RF24_2MBPS);//RF24_250KBPS for 250kbs, RF24_1MBPS for
  1Mbps, or RF24_2MBPS for 2Mbps
  radio.setCRCLength(RF24_CRC_8);//RF24_CRC_8 for 8-bit or RF24_CRC_16 for
  16-bit
  radio.setAutoAck(false);
  radio.setRetries(1,1);
  radio.setChannel(110);
  radio.openWritingPipe(pipe[0]);//*****1
  radio.openReadingPipe(1,pipe[3]);
  radio.startListening();
  b0.attachPop(InitMapping);
  b1.attachPop(InitEnderecamento);
  b2.attachPop(FimEnderecamento);
  b3.attachPop(MaquinaAnterior);
  b4.attachPop(MaquinaSeguinte);
  b5.attachPop(GravarConfigMaquina);
  b6.attachPop(DesabilitarEnderecamento);
  delay(500);
}
```

```
} //end setup()
void loop()
{
    if(!init_mapping && !Mapping_Step_1){
        for(int i=0; i<512; i++){
            //TxBuffer[i]=i;
            TxBuffer[i] = ArduinoDmx0.RxBuffer[i];
        }
        SEND();
    }
    Receive();
    if(init_mapping && (millis() - time_millis_init_Mapping < time_init_mapping)){
        Detect_Modules();
    }
    if(millis() - time_millis_init_Mapping > time_init_mapping){
        init_mapping = 0;
    }
} //end loop()
```


Anexo VII - Esquema elétrico do controlador de DMX sem fios terminal

Neste anexo pretende-se expor, de uma forma simplificada, a estrutura do programa desenvolvido para o mapeamento HMI para o controlador recetor terminal

```
/*-----*/
/*----Projeto de Investigação Comunicação Sem Fios DMX512-----*/
/*----Área de Especialização em Automação e Comunicações em Sistemas Industriais---*/
/*----Autor: Tiago Manuel de Oliveira Crespo-----Dezembro 2017-----*/
/*-----*/

#include <lib_dmx.h>
#include <SPI.h>
#include <nRF24L01.h>
#include <RF24.h>
#include <EEPROM.h>

#define DMX512 (0) // (250 kbaud - 2 to 512 channels) Standard USITT DMX-512
#define CE_PIN 9
#define CSN_PIN 10

const uint64_t pipe[4] = { 0xF0F0F0F066LL, 0xF0F0F0F0ALL, 0xEEFAFDDEELL,
0xEEFDFAF50DFLL};

RF24 radio(CE_PIN, CSN_PIN);

byte RxBuffer[512] ;
byte AuxRxBuffer[18] ;
byte TXMapping[18] ;
byte ID_Pacote_Act = 0;
byte ID_Mensagem_Act = 0;
byte ID_Mensagem = 0;

bool RxBufferReady = 0;
bool sendsuccess = 0;
```

```
bool done = 0;
bool Mapping_step_0 = 0;//Indicação de que está presente na rede
bool Mapping_step_1 = 0;//Locate para identificação do módulo
bool Mapping_step_2 = 0;//Atribuição dos endereços de funcionamento
bool Modo_Dinamico_ON = 0; //Modulo a funcionar com endereçamento dinâmico de
máquinas
int ID_L = 0;
int ID_H = 0;
int Endereco_Maquina = 0;
int Endereco_Mesa = 0;
int Interval_Mapping_step_0 = 0;
unsigned long Init_Mapping_step_0 = 0;
int Interval_Mapping_step_1 = 2000;
unsigned long Init_Mapping_step_1 = 0;
void setup() /****** SETUP: RUNS ONCE *****/
{
    ArduinoDmx0.set_control_pin(3); // Arduino output pin for MAX485 input/output control
(connect to MAX485-1 pins 2-3)
    ArduinoDmx0.set_tx_address(1); // set rx1 start address
    ArduinoDmx0.set_tx_channels(512); // 2 to 2048!! channels in DMX1000K (512 in standard
mode) See lib_dmx.h *** new *** EXPERIMENTAL
    // New parameter needed: DMX Mode
    ArduinoDmx0.init_tx(DMX512); // starts universe 1 as rx, standard DMX 512 - See
lib_dmx.h, now support for DMX faster modes (DMX 1000K)
    //Serial.begin(115200);
    pinMode(6, OUTPUT);
    radio.begin();
    radio.setPALevel( RF24_PA_HIGH ); //MIN=-18dBm, LOW=-12dBm, MED=-6dBm, and
HIGH=0dBm
    radio.setDataRate(RF24_2MBPS);//RF24_250KBPS for 250kbs, RF24_1MBPS for 1Mbps,
or RF24_2MBPS for 2Mbps
    radio.setCRCLength(RF24_CRC_8);//RF24_CRC_8 for 8-bit or RF24_CRC_16 for 16-bit
    radio.setAutoAck(false);
    radio.setRetries(1,1);
}
```

```
radio.setChannel(110);
if(EEPROM.read(0) == 255 || EEPROM.read(1) == 255){
  EEPROM.write(0, random(0, 254));
  EEPROM.write(1, random(0, 254));
}
ID_L = EEPROM.read(0);
ID_H = EEPROM.read(1);
if(EEPROM.read(10) == 1){
  Endereco_Maquina =0x0000;
  Endereco_Mesa =0x0000;
  Endereco_Maquina = Endereco_Maquina | (EEPROM.read(12) << 8); //Byte Mais
significativo
  Endereco_Maquina = Endereco_Maquina | EEPROM.read(11); //Byte Menos significativo
  Endereco_Mesa = Endereco_Mesa | (EEPROM.read(14) << 8); //Byte Mais significativo
  Endereco_Mesa = Endereco_Mesa | EEPROM.read(13); //Byte Menos significativo
}
radio.openReadingPipe(1,pipe[0]);
radio.startListening();
//radio.openWritingPipe(pipe[2]);
}/--(end setup )---
void loop() /****** LOOP: RUNS CONSTANTLY *****/
{
  if ( radio.available() ) //#####Receção de Dados
  {
    // Read the data payload until we've received everything
    while (radio.available())
    {
      // Fetch the data payload
      radio.read( AuxRxBuffer, 18 );
    }
    if(AuxRxBuffer[0] < 200){ //*****Receção de Tramas DMX
      for(int j=0; j < 16; j++){
        RxBuffer[(AuxRxBuffer[1] * 16)+j]=AuxRxBuffer[j+2];
```

```

    }
    ID_Pacote_Act = AuxRxBuffer[1];
    RxBufferReady = 1;
    }
    if(AuxRxBuffer[0] == 201 && AuxRxBuffer[1] == 100){
//*****
*****Inicio Mapeamento
        if(Mapping_step_0 == 0){
            Init_Mapping_step_0 = millis();
            Interval_Mapping_step_0 = random(0,200);
            Mapping_step_0 = 1;
        }
    }
    if(AuxRxBuffer[0] == 202 && AuxRxBuffer[1] == 100 && AuxRxBuffer[2] == ID_L
&& AuxRxBuffer[3] == ID_H){ //*****Confirmacao
Mapeamento
        Mapping_step_0 = 0;
    }
    if(AuxRxBuffer[0] == 203 && AuxRxBuffer[1] == 101 && AuxRxBuffer[2] == ID_L
&& AuxRxBuffer[3] == ID_H){ //*****Inicio
Endereçamento
        Mapping_step_1 = 1;
    }
    if(AuxRxBuffer[0] == 204 && AuxRxBuffer[1] == 101 && AuxRxBuffer[2] == ID_L
&& AuxRxBuffer[3] == ID_H){ //*****Confirmacao
Endereçamento
        Modo_Dinamico_ON = 1; //Ativação do Modo Dinâmico
        EEPROM.write(10, 1); //Ativação do Modo Dinâmico
        Mapping_step_1 = 0;
        Endereco_Maquina = 0;
        Endereco_Mesa = 0;
        Endereco_Maquina = Endereco_Maquina | (AuxRxBuffer[5] << 8); //Byte
        Endereco_Maquina = Endereco_Maquina | AuxRxBuffer[4]; //Byte Menos significativo
        Endereco_Mesa = Endereco_Mesa | (AuxRxBuffer[7] << 8); //Byte Mais significativo
        Endereco_Mesa = Endereco_Mesa | AuxRxBuffer[6]; //Byte Menos significativo

```

```

EEPROM.write(11,AuxRxBuffer[4]);
EEPROM.write(12,AuxRxBuffer[5]);
EEPROM.write(13,AuxRxBuffer[6]);
EEPROM.write(14,AuxRxBuffer[7]);
}
if(AuxRxBuffer[0] == 205 && AuxRxBuffer[1] == 101 && ((AuxRxBuffer[2] == ID_L
&& AuxRxBuffer[3] == ID_H) || (AuxRxBuffer[2] == 255 && AuxRxBuffer[3] == 255)){
//*****Desabilitar Modo Endereçamento

Modo_Dinamico_ON = 0; //Desativação do Modo Dinâmico
EEPROM.write(10, 0); //Desativação do Modo Dinâmico
Endereco_Maquina = 0;
Endereco_Mesa = 0;
}
} //##### Fim Recepção de Dados

if(Mapping_step_0 && (millis() - Init_Mapping_step_0 > Interval_Mapping_step_0)){
//Envio do ID para o Modulo Regie

radio.stopListening();
TXMapping[0] = 211;
TXMapping[1] = 100;
TXMapping[2] = ID_L;
TXMapping[3] = ID_H;
radio.openWritingPipe(pipe[3]);
radio.write( TXMapping, 18 );
radio.startListening();;
Init_Mapping_step_0 = millis();
Interval_Mapping_step_0 = random(0,200);
}

if(Mapping_step_1 && (millis() - Init_Mapping_step_1 > Interval_Mapping_step_1)){
//*****Locate Para Endereçamento

Init_Mapping_step_0 = millis();
int Locate_buf = random(10,250);
for(int k=0; k<512; k++){
ArduinoDmx0.TxBuffer[k]=Locate_buf;
}
}

```

```
    Mapping_step_1 = 0;
}
if(RxBufferReady &&!Mapping_step_0 && !Mapping_step_1 &&
!Modo_Dinamico_ON){/**Atualização Trama DMX modo Normal
for(int k=0; k<512; k++){
    ArduinoDmx0.TxBuffer[k]=RxBuffer[k];
}
//Serial.println("");
RxBufferReady = false;
}
if(RxBufferReady && Modo_Dinamico_ON){//Atualização Trama DMX Dinamico
    Endereco_Maquina = 0;
    Endereco_Mesa = 0;
    if(Endereco_Maquina >= Endereco_Mesa){
        for(int k = 0; (k + Endereco_Maquina)<512; k++){
            ArduinoDmx0.TxBuffer[Endereco_Maquina + k]=RxBuffer[Endereco_Mesa + k];
        }
    }
    else{
        for(int k = 0; (k + Endereco_Mesa)<512; k++){
            ArduinoDmx0.TxBuffer[Endereco_Maquina + k]=RxBuffer[Endereco_Mesa + k];
        }
    }
    RxBufferReady = false;
}
}/**--(end main loop )---
```

Anexo VIII - Software de Simulação e visualização

Este capítulo complementar, pretende demonstrar as funcionalidades dos *softwares* de simulação gráfica dedicados à iluminação profissional, de forma a realçar a sua importância no desenvolvimento de projetos e eventos. A criação deste anexo, pretende demonstrar o estudo e aprendizado realizado na exploração, por curiosidade, deste tipo de *interfaces* e na ampla aplicabilidade que podem possuir, no desenvolvimento de projetos como o realizado. Este tipo de *softwares*, requerem um vasto estágio de aprendizagem e dedicação.

Atualmente, os *softwares* de desenho técnico e simulação, permitem, pré-conceber e pré-visualizar o cenário físico do evento e o comportamento da iluminação profissional. Estes, consistem em aplicações robustas de desenho técnico virtual que requerem de um conjunto abundante de ferramentas para atender às necessidades dos *designers*. Contudo, estas ferramentas consistem em *interfaces* muito complexas que exigem grandes esforços de aprendizagem e de tempo dedicado na execução de determinadas tarefas.

O estágio de visualização e simulação desempenha um papel crucial na venda, projeção e interligação de dispositivos em eventos a realizar. Várias aplicações já estão a cumprir o pressuposto de pré-visualização e pré-controlo de capacidades, exercendo uso de recursos de visualização virtual 2D ou 3D.

O interesse em visualizar a programação de iluminação de cenários virtuais é cada vez mais uma necessidade e uma exigência no setor audiovisual. A iluminação complexa agora vista em grandes eventos como concertos, teatros, festivais e outros eventos é cada vez mais difícil e demorada para ser desenvolvida no local, devido ao crescimento exponencial do número de máquinas e à complexidade das suas características de controlo. Muitas destas tarefas complexas são atualmente realizadas em controladores dedicados de iluminação, que cada vez mais perseguem a integração destes sistemas. No entanto, a possibilidade de pré-desenvolvimento da programação de um determinado evento com recurso à simulação é cada vez mais uma opção impossível de contornar. Também a visualização e correção de movimentos e efeitos da iluminação num ambiente controlado virtual, com recursos em tempo real, realçam as amplas capacidades destes *softwares* especializados.

Existe inúmeras soluções comerciais de desenvolvimento e simulação gráfica dedicados à iluminação profissional. No entanto, apenas algumas conseguem integrar conjuntamente, ferramentas de *design*, programação, controlo de iluminação e visualização em tempo real.

As ferramentas especiais de *design* de iluminação *CAD*, como o *WYSIWYG10*, já possuem capacidades de representação gráfica final para pré-visualização em tempo real, que fornece representação virtual da exibição de luz. No entanto, alguns *designers* ainda preferem ter o *design* de palco feito com uma ferramenta *CAD* padrão, como *AutoCAD* ou *3D Studio Max* e depois importar seus desenhos para um *software* de visualização. Além disso, novas possibilidades de *interface*, como ambientes 3D imersivos e interações baseadas em localização e movimento, ainda não muito expressivas e em estudo, estimulam o avanço do estado da arte deste setor [71].

No desenvolvimento de ambientes gráficos com recursos à simulação, existem diversas etapas principais que devem ser seguidas no intuito de obter um bom resultado. O diagrama seguinte, pretende demonstrar o percurso a ser efetuado entre estas diversas etapas:

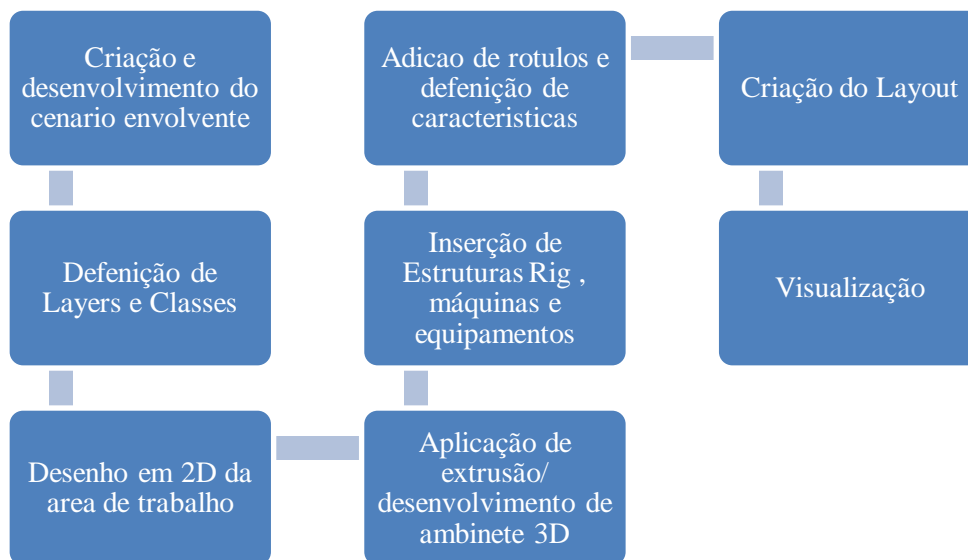


Figura 7.1 - Diagrama de criação de *Layout* do evento e respetivo cenário [72].

Uma vez criado, de forma integral, o cenário virtual, o luminotécnico ou utilizador já pode exercer uso deste, no apoio ao desenvolvimento e criação da programação a realizar. Para tal deve também, aqui, seguir alguns passos de forma a não transpor algumas etapas essenciais ao bom funcionamento da simulação aquando a realização de testes da programação desenvolvida.

No estagio de visualização devem ser seguidos os seguintes passos:

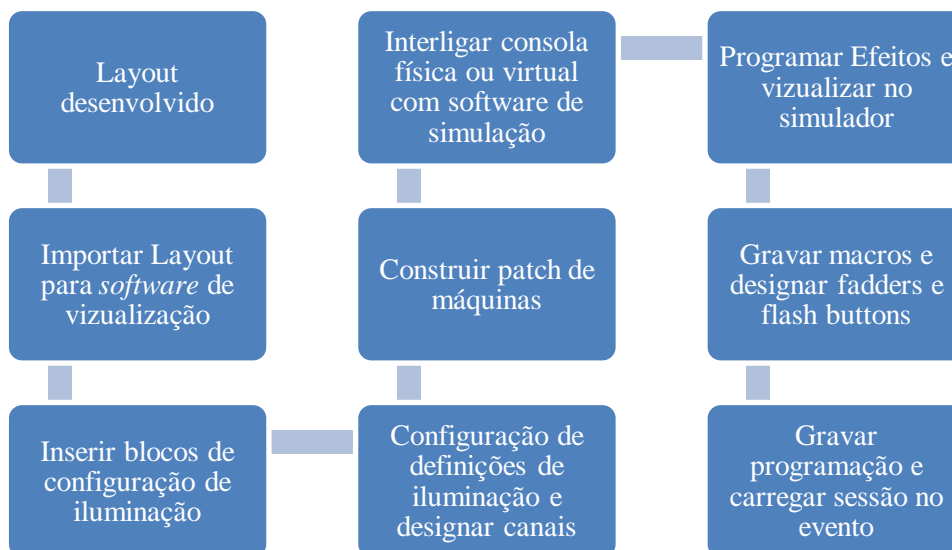


Figura 7.2 - Diagrama de Simulação e Visualização [72].

Visto isto, é efetuada uma breve introdução ao *software* WyziWyg Lighting Design, o qual é abordado e demonstradas algumas das suas características no intuito de possibilitar a intercomunicação deste com um *software* de controlo de iluminação profissional.

WyziWyg Lighting Design Software



Figura 7.3 - Logotipo do *software* Wyziwyg Lighting Design [73].

O Wyziwyg é um *software* de design abrangente, que incorpora pré-visualização virtual em tempo real, movimento e conteúdo de iluminação e *mídia*, usado principalmente para o meio audiovisual. Este permite o desenvolvimento gráfico do ambiente envolvente real da aplicação, a implementação de objetos, estruturas e equipamento de iluminação profissional. Bem como, a capacidade de comunicação de dados de movimento e controlo, através do recurso a funções e ferramentas do sistema, para criar ambientes dinâmicos ou estáticos na área de entretenimento. O *software* permite a configuração da plataforma de iluminação virtual com a possibilidade de intercomunicação com sistemas de controlo externos, como controladores ou mesas de iluminação compatíveis. Desta forma o *designer* ou luminotécnico pode desenvolver todas as configurações e cenários que pretende, podendo até, efetuar a sua programação integral, deixando apenas alguns ajustes para o início do evento. Esta facilidade é mais útil em grandes aplicações de luz em movimento e ajuda o utilizador a construir alguns dos elementos de um espetáculo antes da sua montagem. Características como representação de projeção de luz, capacidade de controlo de movimentos de máquinas, objetos e estruturas, interpretação minuciosa de sombras de objetos e superfícies, *hotspots* e outros aprimoramentos tornam este visualizador extremamente preponderante. Na Figura 7.4 é demonstrada a capacidade de realismo virtual possível do *software* descrito.

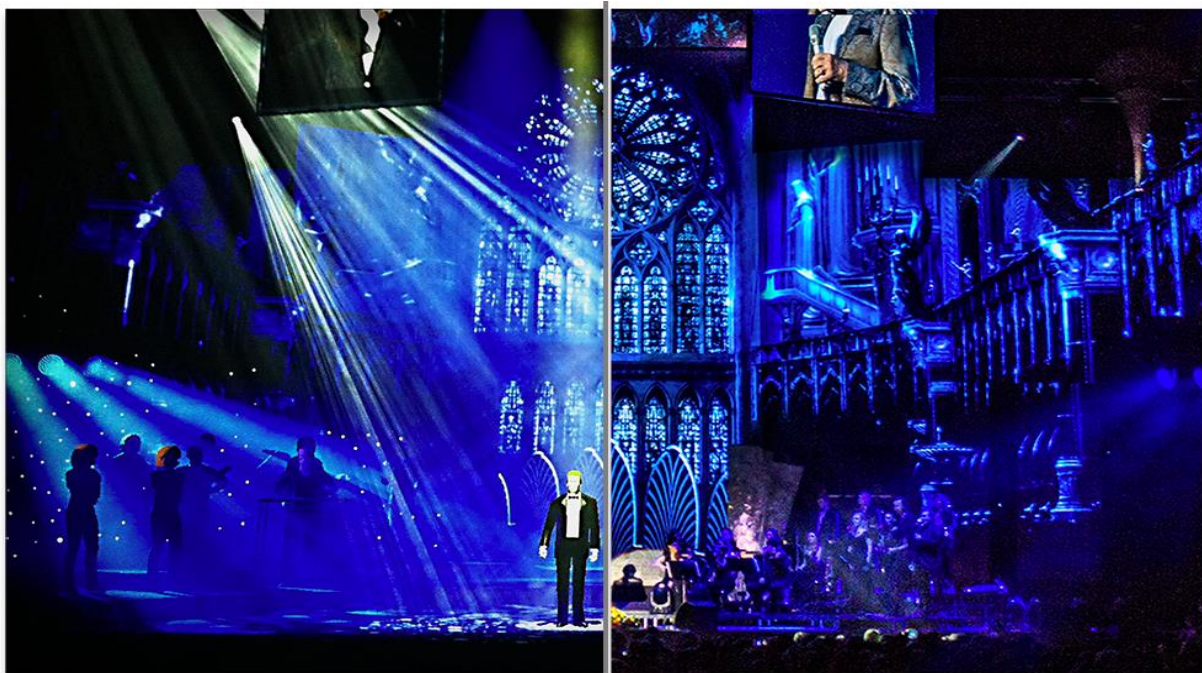


Figura 7.4 - Simulação Wyziwyg (esquerda) e evento real (direita) de Karel Gott tour 2014 [74].

A pré-programação, a configuração de posições de luz em movimento, parâmetros e de efeitos e até programação de macros são efetivamente uma mais-valia que estabelece poupanças de

recursos a todos os níveis. Estas poupanças são reais no mundo do desempenho, logístico e económico. A possível comunicação deste *software* com controladores externos reais, bem como muitos controladores *off-line*, possibilita a pré programação de toda a rede de iluminação profissional, o seu controlo e visualização em tempo real, permitindo, posteriormente, a sua análise, reedição, e guarda do programa desenvolvido. Como pressuposto os circuitos *dimmer* de iluminação convencional também podem ser programados, embora os níveis de intensidade da luz não possam realmente ser representados com precisão, devido às características e parâmetros de fabrico, desgaste das lâmpadas utilizadas, ou do próprio *hardware*.

Este *software* destaca-se pela alta velocidade de funcionamento e desenvolvimento, qualidade de imagem e a representação fiável dos ambientes construídos, dependendo integralmente da rigorosidade do utilizador. Como visto anteriormente, o WYSISYG, utiliza ferramentas de desenho 2D e 3D e avançadas ferramentas de *design* e configurações de realismo, bem como o recurso a bibliotecas de elementos gráficos básicos e projetos pré-construídos. Estas podem ser importadas de outra ferramenta de desenho CAD e vice-versa [71].

A comunicação otimizada deste *software* com o *software* de programação Avolites Titan utilizado no desenvolvimento deste projeto, revelou ser no mínimo complexa e difícil, pelo qual foi utilizado para aprendizagem básica e realização de pequenos testes experimentais, o *software* GrandMA onPC. No entanto, as exigências necessárias para trabalhar com estas interfaces, bem como o tempo exigido de aprendizagem, obrigam a uma enorme dedicação para desenvolver projetos de grandes dimensões como o demonstrado na Figura 7.5.



Figura 7.5 - Imagem da simulação *Wysiwyg* do evento *Swedish Melodifestivalen* [75].